



**HAL**  
open science

## Food Process Ontology Requirements

Damion Dooley, Magalie Weber, Liliana Ibanescu, Matthew Lange, Lauren Chan, Larisa Soldatova, Chen Yang, Robert Warren, Cogan Shimizu, Hande K Mcginty, et al.

► **To cite this version:**

Damion Dooley, Magalie Weber, Liliana Ibanescu, Matthew Lange, Lauren Chan, et al.. Food Process Ontology Requirements. Semantic Web journal, inPress, pp.1-32. 10.3233/SW-223096 . hal-03896435

**HAL Id: hal-03896435**

**<https://hal.inrae.fr/hal-03896435v1>**

Submitted on 13 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Food Process Ontology Requirements

Damion Dooley<sup>1\*</sup>, Magalie Weber<sup>2</sup>, Liliana Ibanescu<sup>3</sup>, Matthew Lange<sup>4</sup>, Lauren Chan<sup>5</sup>, Larisa Soldatova<sup>6</sup>, Chen Yang<sup>7</sup>, Robert Warren<sup>8</sup>, Cogan Shimizu<sup>9</sup>, Hande K McGinty<sup>10</sup>, William Hsiao<sup>1,11</sup>

<sup>1</sup> Centre for Infectious Disease Genomics and One Health, Simon Fraser University, Burnaby, BC, Canada

<sup>2</sup> INRAE, UR BIA, Biopolymères Interactions Assemblages, Nantes, France

<sup>3</sup> UMR MIA-Paris, AgroParisTech, INRAE, Université Paris-Saclay, Paris, France

<sup>4</sup> International Center for Food Ontology Operability Data and Semantics, Davis, California, USA

<sup>5</sup> College of Public Health and Human Sciences, Oregon State University, Corvallis, OR, USA

<sup>6</sup> Goldsmiths College, the University of London, UK

<sup>7</sup> Department of Food Technology, Safety and Health, Ghent University, Ghent, Belgium

<sup>8</sup> Glengarry Agriculture and Forestry, Guelph, On, Canada

<sup>9</sup> Kansas State University, Manhattan, KS, USA

<sup>10</sup> Department of Chemistry and Biochemistry, Ohio University, Athens, Ohio, USA

<sup>11</sup> Faculty of Health Sciences, Simon Fraser University, Burnaby, BC, Canada

\* *corresponding author*

**Abstract.** People often value the sensual, celebratory, and health aspects of food, but behind this experience exists many other value-laden agricultural production, distribution, manufacturing, and physiological processes that support or undermine a healthy population and a sustainable future. The complexity of such processes is evident in both every-day food preparation of recipes and in industrial food manufacturing, packaging and storage, each of which depends critically on human or machine agents, chemical or organismal ingredient references, and the explicit instructions and implicit procedures held in formulations or recipes. An integrated ontology landscape does not yet exist to cover all the entities at work in this farm to fork journey. It seems necessary to construct such a vision by reusing expert-curated fit-to-purpose ontology subdomains and their relationship, material, and more abstract organization and role entities. The challenge is to make this merger be, by analogy, one language, rather than nouns and verbs from a dozen or more dialects which cannot be used directly in statements about some aspect of the farm to fork journey without expensive translation or substantial dialect education in order to understand a particular text or domain of knowledge. This work focuses on the ontology components - object and data properties and annotations - needed to model food processes or more general process modelling within the context of the Open Biological and Biomedical Ontology Foundry and congruent ontologies. Ideally these components can be brought together in a general process ontology that can be specialized not only for the food domain but for carrying out other protocols as well. Many operations involved in food identification, preparation, transportation and storage - shaking, boiling, mixing, freezing, labeling, shipping - are actually common to activities from manufacturing and laboratory work to local or home food preparation.

## 1. Introduction

Food processing has developed a Janus-faced reputation, offering on the one hand historically-significant transformational changes that make food inexpensive, spoilage-resistant, more nutritious, flavorful, shelf-stable, and convenient[1]. On the other

hand, many of the same features have encouraged overconsumption, unhealthy diets, and food products that upset human gut microbiome dynamics and health in general[2]. Academic research into food engineering emerged in the 1950s to understand economic and human health implications of an increasingly industrialized and globalized food manufacturing system,

including the processing, production, handling, storage, conservation, control, packaging and distribution of food products. Food engineering is a multidisciplinary field that combines microbiology, applied physical sciences and chemical engineering to understand and design products and operations in the food and beverage industries.

People are increasingly concerned about health issues and environmental impact, and they are also looking for tasty foods, naturalness and practicality. Modern objectives for data-driven food processing include the desire for information systems to deliver health through increasingly scientifically-based precision nutrition capable of addressing the nutritional needs of individuals according to such factors as their age, health/disease state, activity/lifestyle factors, and allergic and avoidance responses to specific foods/ingredients to name a few. Food manufacturers are also finding value in personalizing foods and diets to enable consumers to make decisions about their food choices based on their ethics (e.g., animal welfare), that extend beyond the common choices of Kosher, Halal, and vegetarianism. Data insights derived from alternative processing streams also enable consumers to make choices about the environmental impact of these processes. At the same time, advances in modularized food technologies are enabling food manufacturers to begin to deliver foods according to individual flavor, texture, and other hedonic profiles.

The complex and interdependent decision making within a landscape of production, distribution, consumer and regulatory stakeholders calls for a standardized ledger of information collected along the processing history of food product lifecycles that stakeholders can contribute to. Digital twin systems built from product identification

standard like the GS1 Digital Link combined with knowledge graph infrastructure like Origin Trail[3] are providing stakeholders with a mechanism for capturing product contextual data from farm to fork but this will be unwieldy if simply a potpourri of contributor's siloed language. Participants in this information chain will benefit from ontologized food processing language that covers organism taxonomy and anatomy, farming and transportation context, and food processing and preparation methods.

Food industry supply chain strategists need their suppliers to adopt the same component specification language, or must endure the continuous cost of mapping potential supplier offerings in order to achieve flexible sourcing of components. Meanwhile, consumers, auditing agencies and self-regulating corporations desire environmental, health, safety and human rights information on farming and manufacturing practices from all the same food supply chain actors.

From a public health perspective, infectious disease epidemiologists investigating foodborne outbreaks need access to forward and backward traceability, as well as food processing and preparation details - involving raw, dried, cooked, preserved, and shelf-stable distinctions for example - of multi-component foods from multiple sources. Nutritional, rare-disease factor and allergen analysis requires similar details of both harvested and processed food. Individual and population level research on food nutrient density and energy bioavailability requires ontology vocabulary for modelling modest or intensive "industrial" food processing (for example cooking, preservation, fortification, hydrolyzation) on food group, specific foods, or other processing outputs such as fiber quality, structure (e.g. nutrient interaction,

starch structures, etc.), toxicity and feelings of satiety. This data can be applied to diet - health / disease relationship analysis, and lead to more accurate nutrition policies and individualized or public health recommendations.

Food manufacturing, preparation and distribution perspectives involve information at various levels of food processing detail including modelling of assembly line machinery, quality control sensors, and robotics in dynamic environments[4]. Even storage of food becomes more complex with “Cold chain” transportation and storage of temperature sensitive food that requires dynamic environmental control to counter aging and decaying processes. Finally, process modelling is required by semi-automated kitchen appliances that have some situational awareness (for example, IoT refrigerators that have a database of their contents to minimize spoilage; ovens that have cooking program profiles for certain foods).

### *1.1. Scope of process ontology survey*

A main objective of this gap analysis work is to provide recommendations for a generic process modelling framework that works seamlessly within the open source Open Biological and Biomedical Ontology Foundry (OBO)[5] community of ontologies. We aim primarily to review current OWL ontologies that have object properties and classes needed for an OBO process model. We focus on comparing entities involved in temporal, part-hood, input and output, and participant / actor, and process dependencies which are required to model food processing and more general lab or manufacturing protocols sufficiently. These entities can then be added-to or echoed in the FoodOn[6] food ontology, a hub within OBO for agricultural,

health and nutrition content related to food products and food processing which is essential for food traceability and other farm-to-fork applications. FoodOn was initially created as an OWL ontology transform of LanguaL[7], a popular longstanding food composition thesaurus.

Crafting a generalized process ontology requires a wide community of domain-specific users to critique it, as well as due diligence - an examination of the objectives, capabilities, and lessons learned from existing process related ontology projects. We review the EXperimental ACTions ontology (Exact2) [8], Process and Observation Ontology (PO2)[9] and the Ontology for Biomedical Investigations (OBI)[10] planned process model, which have various degrees of OBO compatibility, as well as the PROV ontology (PROV-O) [11], the Time Ontology (OWL-Time)[12] the Sensor, Observation, Sample, and Actuator ontology (SOSA)[13] which have significant adoption momentum; and finally touch on Schema.org’s recipe related model as it has a number of process related elements. Some object properties can be reused as is (such as OWL-Time “hasTime”), while others would need equivalent OBO entities created due to OBO principles as discussed in section 1.2 .

Following the review, we synthesise a recipe food processing model from parts of the compared ontologies, and conclude by presenting a simple “boiled carrots” recipe use-case to illustrate how a generic processing model could satisfy a variety of stakeholders: consumers who simply need well-organized recipe instructions; research chefs that formally iterate end product characteristics to achieve a formulation; and food scientists who translate a batch formulation into industrialized continuous processing protocols. Our aim is to achieve a

generic process model that applies to food processing, as well as laboratory, assembly line and robotics applications without having to change the language of its entities and relations. We break down the recipe into discrete steps which have a technical model able to support the needs of all three aforementioned stakeholders.

Our proposed recipe model is already being promoted in a recent paper, “Food Recipe Ingredient Substitution Ontology Design Pattern”[14], which also commits to both OBO food ontology network and OBO relations compatibility, and advances the semantic framework of ingredient substitution beyond FoodOn’s basic “has food substance analog” object property. Their work is contributed as Ontology Design Patterns (ODPs)[15] to the [ontologydesignpattern.org](http://ontologydesignpattern.org) web portal[16]. They use what appear to be provisional recipe model term identifiers that are ideally replaced by our recommended term’s identifiers as they are finalized.

Our gap analysis is not focused on identifying potential taxonomies or ontologies of types of material processing (for example, “heating” or “frying”) or food products or their categorization – these are mentioned only by way of example towards the end of the paper, and are left to specific agricultural, culinary, regulatory, and food science domain ontologies such as FoodOn to support. For example, the “FOOD in Open Data”[17] pilot suite of food related ontologies published in 2016 provides policy-related information about Italian “Protected Designation of Origin” food items, but has no object properties pertaining to food processing and so was not reviewed. Similarly, while FoodOn has a hierarchy of over 250 food transformation processes (FOODON:00002451) which have some modelling with respect to specified inputs

and outputs, we do not reference this hierarchy here except by way of a few examples. A separate project is under way to recast this hierarchical structure within the larger scope of planned (intentional) and unplanned (natural) processes organized by biological, chemical, or mechanical transformation or phenotypic objective.

We did not review Simple Knowledge Organization System vocabularies although they are a close relative of OWL ontology in the semantic vocabulary pantheon, offering more of a broader-narrower concept hierarchy. For example, Agrovoc[18], an FAO supported SKOS vocabulary which excels at a multi-lingual library science style organization of over 40,000 concepts, including process, partitive and quantitative relationship object property branches which have similarities to OBO foundry ones, has been a longstanding target for OBO Foundry term mapping[19]. However, SKOS provides a looser logical framework than what we seek to utilize in OWL (there is no way to express compound term axioms for example)[20], so it was not reviewed.

We do not provide a comprehensive review of process model vocabularies outside of the OWL sphere, but we do reference some of their features in section 1.3 so that the reader can generally see where our reviewed OWL ontologies excel or fall short of a generic process model. A number of these vocabularies, like the flow-chart-like Business Process Model and Notation (BPMN)[21] an ISO standard notation for modeling and depicting sequential, branching and loop processing, have been adopted by projects that evolved towards OWL ontology expression. For example, to support speedy foodborne pathogen investigation, a food supply chain traceability paper [22] focused on BPMN-based modeling of food supply chain actors

and processes; this work was subsequently built upon in the Food Ontology for Traceability Purpose” (FTTO) [23] which echoes the BPMN food supply chain paper’s process branch. We were unable to source and review the FTTO ontology directly; but its published process model did not appear to offer features beyond what OBO already has.

### *1.2. Targeting an OBO-based process model*

We target OBO because we favour OBO’s strategy of having a minimal set of data properties, a reduced set of object properties, and more emphasis on defining classes of entity at either end of an object property, as a general strategy for enabling semantics to be surfaced about those entities which would otherwise be hidden in computationally opaque property names. Briefly, in this approach, rather than using a data property ‘has age’ to say “Cedar ‘has age’ 4” we create an instance of an ‘age’ data item, state that it is about an instance of an organism named Cedar. This provides a convenient way to type what kind of age it is (since conception, birth, planting, germination) and how it was measured (in years, trimesters, months, life stages, tree rings) without loading relationships with such distinctions. This structure can still be translated to other systems’ more nuanced object and data property representations.

A second motivation for OBO – though not unique to its framework - is to encourage standardization by reducing the number of semantically duplicate terms across member ontologies, thus promoting an encyclopedic comprehensiveness. OBO member ontologies are encouraged to reuse a basic grammar of relationships, supplied mainly from the Relation Ontology (RO)[24], which act like a smaller set of verbs one is allowed

to construct sentences (data structures) with across OBO Foundry domains, making for a lighter learning curve. This helps achieve a second OBO commitment - that member ontologies are logically compatible with each other. The current OBO set of process-related relations is introduced in section 3.1. The opposite paradigm is exemplified by WikiData, a federated RDF graph database which merges databases that make use of over 9,000 properties[25]. This landscape is permeated by diverse and not necessarily complete data models and datasets requiring a larger learning curve to navigate.

Regarding logical consistency, OBO has tacitly promoted reasoning over a given ontology’s logical structure merged with the Basic Formal Ontology (BFO) [26] to expose internal inconsistencies. (Currently, it is up to individual projects to test other pertinent combinations of OBO ontologies however.) By analogy, BFO acts as a grammarian enforcing the proper use of noun and verb types. More recently an alternate compatibility starting point is with the OBO Core Ontology for Biology and Biomedicine (COB)[27], an upcoming ontology that has a handful of commonly used upper-level classes and relationships - material entity, process, characteristic, and information - combined into a single resource with a principle aim of supporting RO relations’ domain and range constraints. COB is a more agnostic semantic territory that other ontologies might establish equivalencies with. A future harmonisation question would be to see, for our proposed OBO process model, how many properties it shares with other upper-level-ontologies such as Dolce-Ultra light[28].

To fit in OBO, a process ontology and the components it imports must generally adhere to key OBO membership criteria. If useful components are not reusable as is in OBO,

then they must be replaced with comparable terms that meet OBO criteria[29], examples of which are listed here:

- **Permanent URL (PURL) management:** Each ontology term is given a URL and attached to a service which returns human and computer readable information about the term. The term URL is expected to exist in perpetuity; a deprecation and replacement term reference system exists which facilitates database updates in the face of evolving ontologies.
- **Curation standard:** Terms are explained in the singular, are given in English, and are lowercase except for proper noun parts. Each term is given an Aristotelian definition that references the parent class and differentiates it from its siblings. Credit is provided for term curators and definition sources.
- **Axiomatization:** Terms are to some extent logically connected by relations to other entities.
- **Collaboration:** An ontology imports another ontology's term rather than replicating the same semantics in one of its own terms - the Minimum Information to Reference an External Ontology Term (MIREOT) principle[30]. Within OBO Foundry, "reference" ontologies stake out domains that are the go-to resources for other ontologies that need them, so ideally one ontology for taxonomy, one for chemistry, one for anatomy, et cetera, but admittedly this level of quality is still a way off.

If a term resource is not completely OBO compatible, data translation could still be

accomplished by term mapping using the "has database cross reference" (oboInOwl:hasDbXref) data property, or via SSSOM mapping[31] for finer semantic tuning that can differentiate between semantic broad/narrow/exact synonymy.

Data interoperability is promoted when an ontology is developed as a set of micromodels each focused on an entity in a given context and level of abstraction[32] as illustrated in Fig. 1 which highlights Phenotypic Ontology (PATO)[33] term application; the expectation is that micromodels can plug into each other and be used for various purposes by other users outside a given curation community. This entails potentially splitting entity management across to other ontologies which is a more complex process requiring mutual understanding, a resource price tag, and stamina. On a positive note, this upfront work avoids dependence on a patchwork of reference materials written in other dialects, and the problems that such vocabulary brings

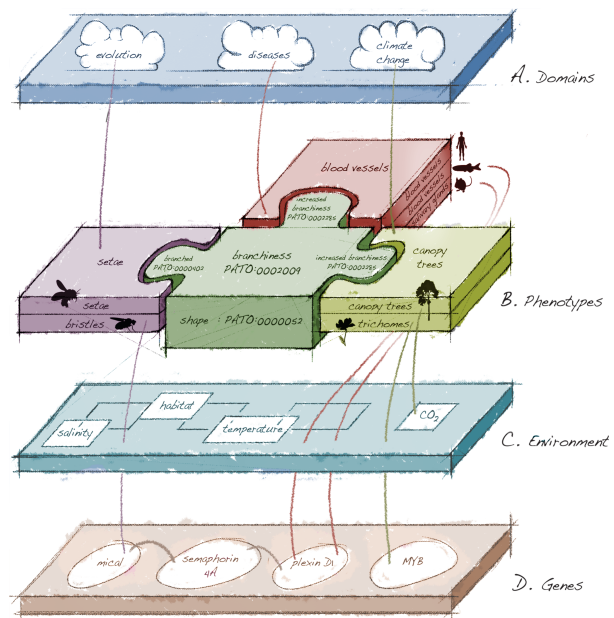


Fig. 1: Micromodel linkage. Used with permission from: "Finding Our Way through Phenotypes"[32].

in terms of comprehension and training time. Also, the more partners involved, the more easily an ontology goes from essentially a database solution for one project, to an inter-agency standard. However, the capabilities of the OBO curation model are being challenged as it evolves from its initial life science comfort zone into connected domains such as manufactured products and the build environment.

Some other ontologies or model frameworks were considered but not given in-depth review: AMALQEIA [34] captures dish roles with respect to a meal (main course etcetera), style of cooking, ingredients, and their nutritional value, and mentions food process modelling but leaves it to a future task to “express precisely which ingredients undergo which process.” Pertinent to food science recipe formulation modeling, Shimizu et al.’s study[35] defines an ecosystem of ontology design patterns that capture chemistry lab procedures and chemical interactions which parallel cooking procedures and food transformations. An abstract “semantic-trajectory pattern” specializes to a “state transition pattern” that then provides a template on which the “chemical process pattern” is defined. The research utilizes patterns such as “action”, “chemical activity”, “chemical process”, and “chemical system”. This state-change-based process model is not pursued here but can be translated to states as an expression of process inputs and outputs, and their characteristics, which are the focus of this paper’s approach.

### *1.3. Process model capabilities*

There are various kinds of physical, chemical, biological or data transformation processes, and intentional processes that harness them. We can model unplanned

processes found in the physical world, which can then be harnessed by planned processes that involve one or more objectives. For example, the ripening of fruit, a biological process, can be manipulated by artificial ripening, and/or timed harvesting and transportation that altogether meet an objective of delivering ripe fruit to customers. Planned processes have a plan or detailed protocol to achieve, and operate on inputs and outputs which are material entities or data. A planned process may involve intrinsic or extrinsic transformations:

- Transform an entity’s composition through mechanical, chemical, biological or other physical production processes.
- Characterize (generate information about) an entity using observational processes which may be invasive or non-invasive.
- Change the relative context or extrinsic relations of an entity, like transportation of things from one location to another.
- Passively affect an entity, for example an object being stored with an objective of preservation.

Processes can be organized linearly or combined in more complex parallel or networked formations that collectively behave as batch or continuous operations. They can be constrained by dependencies and input availability, requiring various resource inputs and performers such as devices and people. They may have minimally necessary durations, rates of change, sub-processes, and side-effects.

An objective of a general process ontology is that it covers more specific process ontology niches - from food traceability, which calls for rough granularity



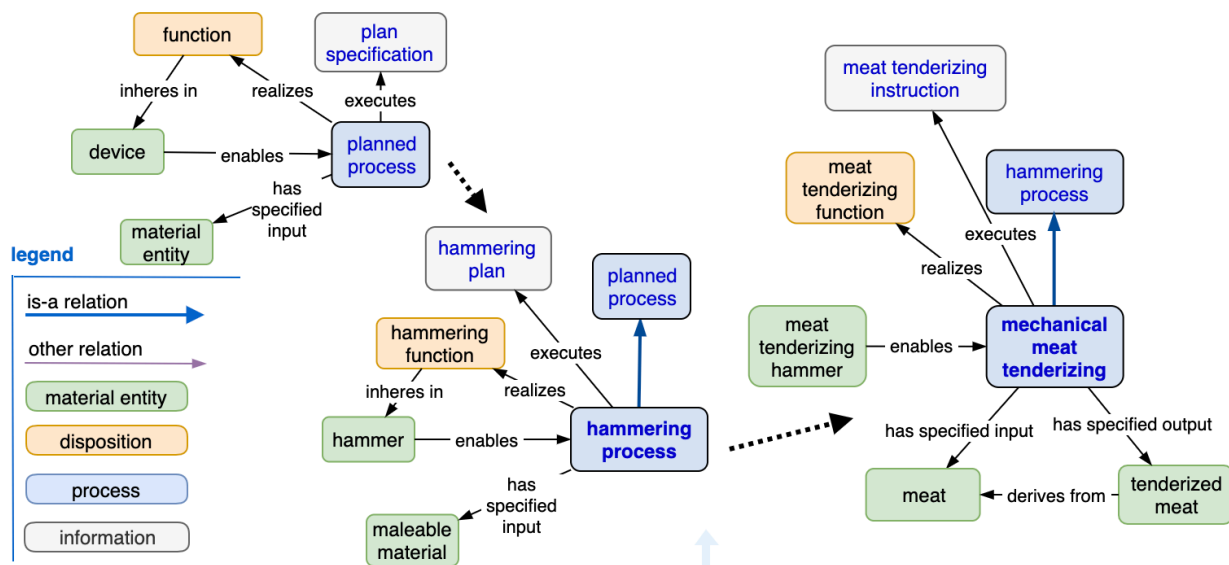


Fig. 2: Specialization of planned process and plan specification.

of process - harvest, storage, transport, division, combination - to more specific modelling, such as how to put a recipe together. Can the same generic process model relations cover specific niche modelling, so that relations in those niches are revealed to be equivalent to generic ones?

## 2. Ontology-based Process Modelling

### 2.1. Process objectives

A process objective can be expressed simply by referring to an output entity, for example, “tenderized meat” as a primary objective - a desired product. A variety of processes may be known to achieve such an output, but choosing one may be constrained by available devices or inputs (for example, tenderizing hammer, vinegar), or time constraints (for example, mechanical tenderizing is the fastest), or skillset issues (for example if a related plan specification / instruction is missing or an agent is unable to comprehend it). More generalized modelling is achieved by describing what the output

entity is such that the process can be recognized as a means to that end. Having defined “tenderised meat” as “meat which has its muscle fibre and connective tissue severed into shorter segments”, then any process which has a similar output can be inferred to be a potential substitution.

Process objectives can also be expressed by referencing functional capabilities of devices. In BFO a function inheres in a device or material, and a process “realizes” or carries out the transformation that a function characterizes (Fig. 2). A “meat tenderizing function” inheres in a meat tenderizing hammer. A mechanical meat tenderizing process “realizes” the tenderizing function. A plan including step by step instructions may be involved - it is “executed” by the process.

There is a philosophical distinction about functionality that exists as a result of evolutionary or intentional design, and a looser semantic of potentially unintended device capabilities that can be evaluated with respect to achieving some goal. A creative approach to process modelling sees overall process objectives satisfied by a selection of

devices based on capability regardless of what they were intentionally designed for, but the challenge then becomes how to recognize (enumerate) such capabilities. Other research[4] has been done on the challenge of having plan specifications include “objective specifications” (IAO:0000005) in a language that both humans and robots can interpret, such as “... the goal of having the pancake mix in the hand”; this is not addressed here.

### 2.1.1. Process modularity variations

There are a few kinds of process model approaches and capabilities which ideally an ontology-driven model can satisfy. A holistic model would capture both autonomous behavior of agents free to roam an environment, actively seeking or waiting for contextual information which they are designed to react to; as well as models limited to describing steps in a workflow, implicitly controlled by a more abstract layer.

**Autonomous-oriented process models** can express the conditions under which a process is activated - a set of criteria about the required environmental context, input material, energy, time constraints, device(s) and operators. **Control-oriented process models** can have a parallel layer of controlling processes that supply input signals such as “start”, “pause”, and “stop”, alongside the agent-oriented conditions for activation. For example, the Data Documentation Initiative (DDI)[36] and recent variant DDI Cross-Domain Integration (DDI CDI)[37] provide a process model, partially illustrated in Fig. 3, that exhibits this approach based on W3C PROV provenance specification of which PROV-O is a part.[38]

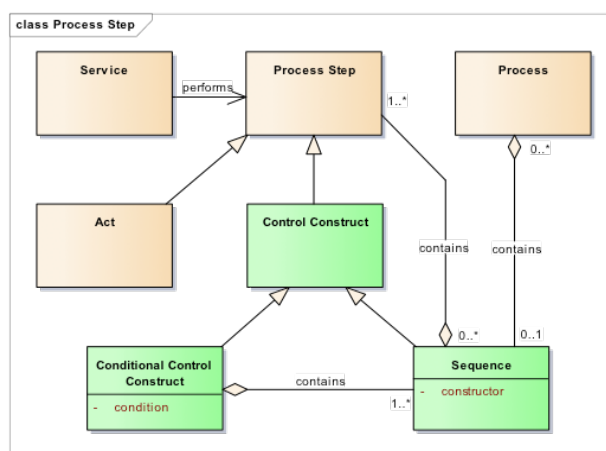


Fig. 3: A DDI process control schema. (From: CC-BY 4.0 license: “Using the Process pattern” — DDI 4.0 dev documentation,(n.d.). <https://ddi4.readthedocs.io/en/latest/userguides/processpattern.html> , accessed November 12, 2021).

Both kinds of model can be connected in a plug and play fashion to create process dependencies and overall process transformation of material / phenotype from an initial to a final state. Rule engines enact the autonomous approach by monitoring an environment’s state. An autonomous process begins to resemble a control-oriented process the more that its environment is reduced to a narrow set of stimuli / inputs (inputs which cause reactions become more apparent.)

Workflow specifications and their computational frameworks that enable a network of processes are the backbone for data processing and experimental reproducibility. Workflow configuration is often a process of manual configuration, but many systems such as CyVerse[39], and Galaxy[40], are providing user interfaces for workflow development that show required inputs and context-sensitive options. Ontology-driven process modelling should be able to fulfill this more control-oriented approach by way of “process-control” classes dedicated to adding informatic control of processes.

### Process and object model

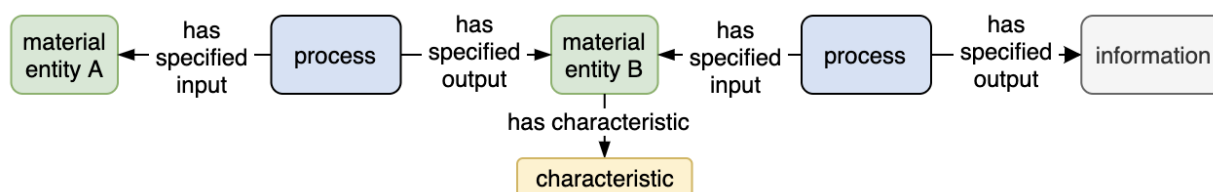


Fig. 4: A combined process-and-object model.

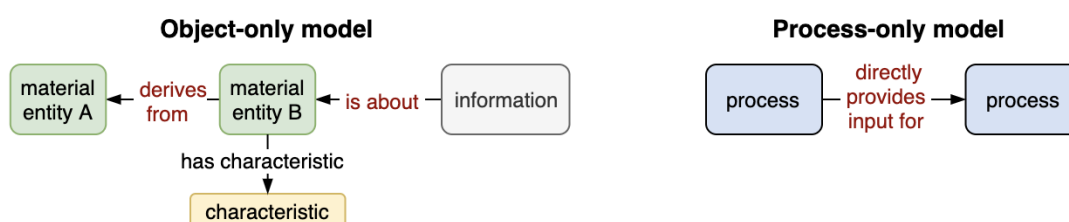


Fig. 5: A combined process-and-object model can be transformed into an object-only model or a process-only model.

#### 2.1.2. Process and object centric models

A representational distinction can be made about whether data is being modeled from a process or an object perspective, or both. A **process-and-object perspective** directly supports a provenance trail - the information about how an entity or its context has been changed in the environment, or some supply chain (Fig. 4). Transformation processes - whether they be assays, data processing methods, or agricultural, storage, shipping or manufacturing methods, link past, present and future states of an entity, which can all be captured in a knowledge graph. This aligns well with product traceability ledgers that document how products are created or transformed in their life cycle by various agents.

An **object-only perspective** of entities and their characteristics at some moment in time (Fig. 5) may cover the contextual who / what / when / why / where of a situation but lacks a framework for describing the “how” - what processes led to the synthesis of an entity from its constituent parts in a supply chain. The story of an entity’s past may be

abbreviated by way of a shortcut relation like “derives from”, an OBO Relations Ontology (RO) term linking a material entity to its precursor(s). An object perspective is derived from a process-and-object model by replacing any given process with a generalized and therefore less informative “derives from” or “is about” (between information and an entity) linkage between inputs and outputs as shown in (Fig. 5). Similarly, a process model that details input and output entities can be reduced to a “**process-only perspective**” dependency linkage between processes by replacing the i/o entities with an RO “directly provides input for” relation (Fig. 5). In this view one loses all the details about any characteristics that might have been associated with the objects that processes were subjected to.

#### 2.1.3. Process steps, parts, dependencies and abstraction

Various process models have a “step” entity, which from a process-control perspective is a convention for naming an abstracted process view and ordering it within a workflow. A step conveys semantics

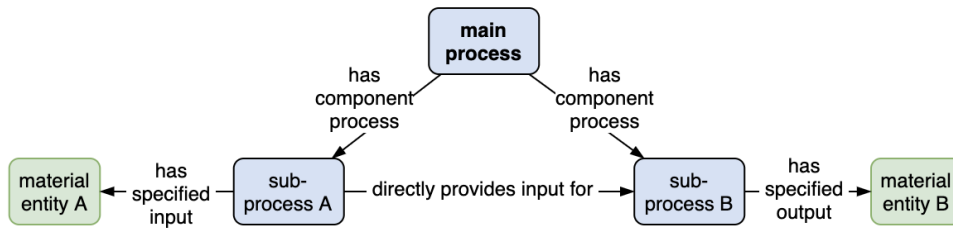


Fig. 6: The steps or sub-processes of a main process.

of a process which can be isolated as a controllable event and that either discretely fails or succeeds, and may have states like pause, or restart. A step captures some level of process modularity and control granularity. It must be completed before proceeding to a subsequent dependent step. On closer inspection a process step can usually be broken down into sub-processes (Fig. 6) which may be a network of interdependent components. This level of detail is usually needed for automation or reproducibility. An elegant process model would be able to offer different levels of granular view on a path or hierarchy or network of processes, such that components could have their ordering inferred (as steps) as constrained by input dependencies.

The necessary inputs of an initial subprocess can be associated with its parent process from a flowchart perspective, and likewise for final subprocess output. This generalization might enable all subprocesses to be hidden from view (Fig. 7).

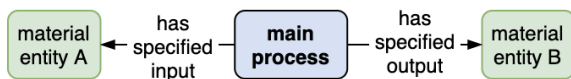


Fig. 7: A generalized process connected to ultimate inputs and outputs.

In the experimental study, drawn with dashed lines, a process is hypothesized to exist directly or indirectly (by proxy) between quality A and B. A study design can take on the delineation of independent and dependent variables.

The connections between material entities, characteristics, processes and information at some time instant are often taken together to provide the framework for describing possible “states” of a “system”. A system state is described by a snapshot (at some moment in time or across some interval) of the system’s material or information or process entities and their characteristics or roles. While some frameworks only express actions to perform on entities (such as “boiling”), the expression of input and output qualities allows expression of changes in states of a system, and conditions that must be met for processing to continue, such as “wait until the water is boiling”.

Some frameworks also allow processes themselves to have or influence characteristics (for example, BFO “process profile” or RO “regulates characteristic”) that can be observed or controlled, for example the speed of a mixer’s perturbations described by PROCO’s “stir rate profile” (PROCO:0000043).

Characteristics involving input supply rates - such as frequency or energy supply magnitude or rate of material supply have one thing in common - a time variable which is specific to one or more particular process inputs (for example, supplying a faster digital clock frequency, or more energy or catalyst per unit of time). Attaching a controlled or monitored quality to a process is a shortcut for modeling qualities that pinpoint process

input or output – a level of detail required for industrial automation.

Experimental study designs make a distinction about which material entity or process characteristics are being treated as controls (or settings), independent variables, or dependent variables. As a prerequisite for success, an experiment's actual control or independent variable levels must match the experimental design or plan, so a quality control step involves the accurate calibration (measuring) of control device setting levels. Consequently control, dependent and independent designations each depend on a simpler underlying semantic concept of measures of variables. We can remove explicit naming of which measures fall into control / independent / dependent variable categories within a study dataset directly, and instead document this distinction in the study design. Fig. 8 shows the use of OBO study design related terms in this regard. The same data collection structure can then be reused for both experimental and observational studies, and in routine production

environments where inputs undergo sampling and/or calibration.

#### 2.1.4. Process model nomenclature

An ontology term label may cater to a specific ontology domain and its expert users, which may seem limiting if repurposed outside that community, but a solution is to augment it with a synonym used by a given application user interface. A “has quality” relation could have an exact synonym “has characteristic” for use within a particular community. Such moves attempt to preserve semantic web data exchange objectives by maintaining database views that can be queried using a reduced set of relations under-the-hood.

A more difficult challenge semantically is to see how much one can rely on the more abstract “glue” or grammar of a generalist ontology - especially its relationships - in order to express specialist ontology concepts. If a generalist ontology suffices, then the chances of plug and play data harmonization across disciplines is increased. The more that

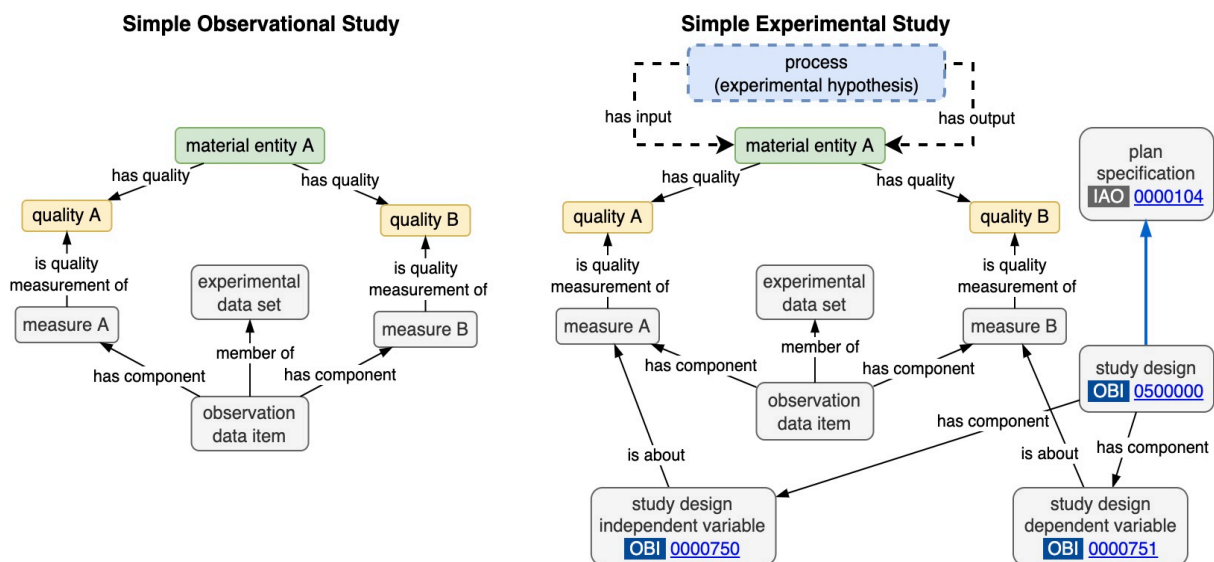


Fig. 8: Schematic comparing simple 2 variable observational and experimental study designs (for example dietary pattern and weight).

specialist grammar is required, the more expensive, sizable, error prone, and ephemeral the learning curve, and the more likely that a semantically fractured data landscape is encountered. However, the core of a generalist vocabulary has a smaller set of relations (like RISC instruction sets in computer science) and so potentially greater sentence complexity (data structure). The focus or granularity of process modelling aims will shape the structures that need to be built out, but ideally one does not have to shift languages or paradigms to construct and maintain a cross-disciplinary knowledge graph.

Terms that may seem semantically interchangeable within a speech community usually require more precise definitions in an ontology, so here we define our use of terms within the context of a generic process model developed within this paper, and avoid the subject of homonym semantics.

- **A plan:** in common language “procedure”, “protocol”, “plan specification” and “workflow” are used to reference informatic (document) constructs that guide process execution. An abstract sense of plan allows steps to be expressed as objectives. In this paper we avoid that challenge, instead focusing on using **plan specification** for organizing concrete process steps.
- **A planned process:** “operation”, “action”, “activity”, or “procedure” often appear to be equivalent to a generic planned process. If a planned process cannot be broken down into finer-grained “steps”, then it is called a unit operation.
- **A characteristic:** a “characteristic”, “feature”, “quality”, “attribute” or “phenotype” are often used to describe an observable property of an object.

Within OBO, BFO uses “**quality**” rather than characteristic for an observable object property, and so we also frequently use that term and sense below. (In food science “quality” may be used as a value judgement of a product, like a “good quality” ripe peach.)

- **A measure** (aka measurement): a data item record of a categorical, numeric, or numeric and unit value. The term “measurement” adds a sense of a real-world object being measured. Measures may be simulated or predicted, and subject to precision and accuracy, and may be the result of faulty or miscalibrated equipment. This paper takes the position that a measure does not include time, place, or sensor “aboutness” information.
- **An observation:** As a noun, an observation (or “result” in SOSA, rather than SOSA’s “Observation” activity) is a data structure output of an invasive or non-invasive observation process effectively at an instant or duration of time, and includes at least one measure that (semantically) “is about” a characteristic of an entity (for example Jane Doe’s weight). Other aspects of an observation may be documented: the particular time and context it happened in, and the sensor (device or human or other organism organ) that was involved. This sense is similar to the Extensible Observation Ontology (OBOE)[41] observation and related terms.

Some use “observation” for qualitative description of phenomena while reserving “measure” for quantitative recording of underlying characteristics of phenomena; our use has “observation” allow both as components. An observation data item requires a data structure to encompass the contextual information about the time and

place etc. of a measure. A data set composed of these observations could adopt the characteristic values they have in common, for example, if all of a dataset’s observations were made at the same location, then that location can become a characteristic of what the dataset is about, and is inherited rather than repeated at the observation level.

Process ontology implementation can reuse the above relations and terms directly into graph database representation (OWL ontologies provide the sentence structure for writing statements directly into RDF graph databases). For other kinds of databases, ontology terms can be used at a more basic level via lookup tables that name categorical variables and their choices, and numeric or date variables. From a data import/export perspective, the ontology structures discussed so far can translate into native relational database records and vice versa, though knowledge graph querying is most flexibly done in a native graph format. Object and data properties for holding values and units are important (examples are shown in Fig. 9), and vary widely between OBO Foundry ontologies and other ontologies and RDF schemes like Schema[42] and GS1[43]; their potential harmonization is not covered here.

### 2.1.5. OWL Process model limitations

A “commonsense” vs OWL logic challenge is that in OWL technically there is no easy way to express that a process has directly changed a quality of an input material (or digital entity) since any such output material instance cannot be asserted to be one and the same as the input material (for example a bleaching process changes a person’s dark hair to blond, so neither the hair nor the person could technically be the “same” entity). One approach is to reason

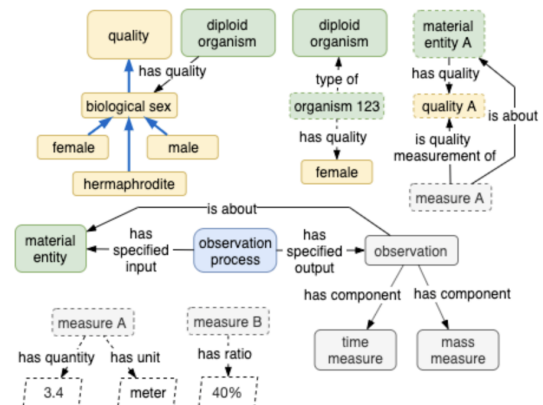


Fig. 9: Examples of abstract measures, object and data

over timestamped observations about objects, rather than over properties of an object directly. Alternatively, one can create an instance of a tracking identifier entity that is used to (manually or via software) denote a given material entity anywhere within a process input / output matrix (see identifier in Fig. 23), effectively marking a set of instance entities as roughly the same continuant while avoiding OWL equivalency logic. This won’t assist OWL classification prowess but it does provide provenance and traceability as a function of the identifiers attached to process resources.

Given the limitations of current OWL reasoner capabilities, a process ontology should mainly be considered the vehicle for providing the grammar of entity categories and relations which are used to create sentences that describe processes. OWL reasoners can only infer class membership, so process step order likely cannot be inferred from process dependencies, and algorithms outside of OWL logic are required. Some OWL reasoners can handle simple numeric data property comparisons ( $x \geq y$ ), but laboratory and assembly line automation require many non-classification operations such as optimizing materials into batches, recalibrating devices, and calculating costs of procedures. Technology

like SWRL[44] or SPARQL[45] is suited to ontology-driven knowledge graph workflow computations. By creating a separation of concern between structural validity and workflow computation, OBO Foundry (and other communities) can then focus on the interoperability mission of aligning various OWL domain ontologies to use a shared relation vocabulary that can express and validate class and instance level data structures.

### 3. Review of OBO and W3C Process Model Ontologies

#### 3.1. OBO process model

OBO offers a general-purpose process model based on more abstract object properties from a handful of ontologies, which will be reviewed here.

- Ontology for Biomedical Investigations (OBI)[10] introduced “planned process”.
- Information Artifact Ontology (IAO)[46] handles data items arising in process i/o.
- Relation ontology (RO)[24] holding most relations that surround a process.

##### 3.1.1. Planned versus unplanned processes

A basic distinction is drawn between an “unplanned” physical, chemical or biological process, and a “planned process” (OBI:0000011) that “executes” (COB:0000086) a “plan specification” (IAO:0000104), which is a kind of “directive information entity” (IAO:0000033). This can be for example a word-of-mouth or printed recipe, or in a digital data structure. A chemical reaction process has no such plan

specification, only a behavior described by chemical equation. The focus in this paper is on processes that involve plan specifications. It is expected that planned processes can in effect control the behavior of unplanned ones by controlling their environmental conditions and inputs.

RO adopts generic process ontology terms for a number of its relations, such as “has input” and “has output”; OBI extends these with “has specified input” and “has specified output” that pertain to planned process, and which have material entities (for assays) or information (for data transformations) as their targets. Fig. 11 shows these relations as well as others that optionally connect to a given planned process.

OBI’s model’s input and output relations and the “has performer” relation (all children of RO “has participant”) can be qualified by a role as shown in pink in Fig. 11, if there is a need to disambiguate participant entities further. For example, the OBI “performing a clinical assessment” process has for input an organism carrying a patient role. It could also have a doctor as a performer, that is, a person carrying a doctor role.

Process parthood is covered by the “has component process” relation connecting to a sub-process; OWL cardinality[47] can be used to mark them as optional; the upstream dependencies of component processes can be specified by way of their input and output entities; this allows for modelling down to an arbitrary level of granularity. Parallel processing constraints are offered by the suite of RO “temporally related to” (RO:0002222) Allen relations which impose restrictions on how processes align with each other - starting or ending at the same time, or in linear fashion (for example see the “starts with”, “ends with”, “preceded by”, “immediately preceded by” relations).



### OBO process model parts + OWL-Time

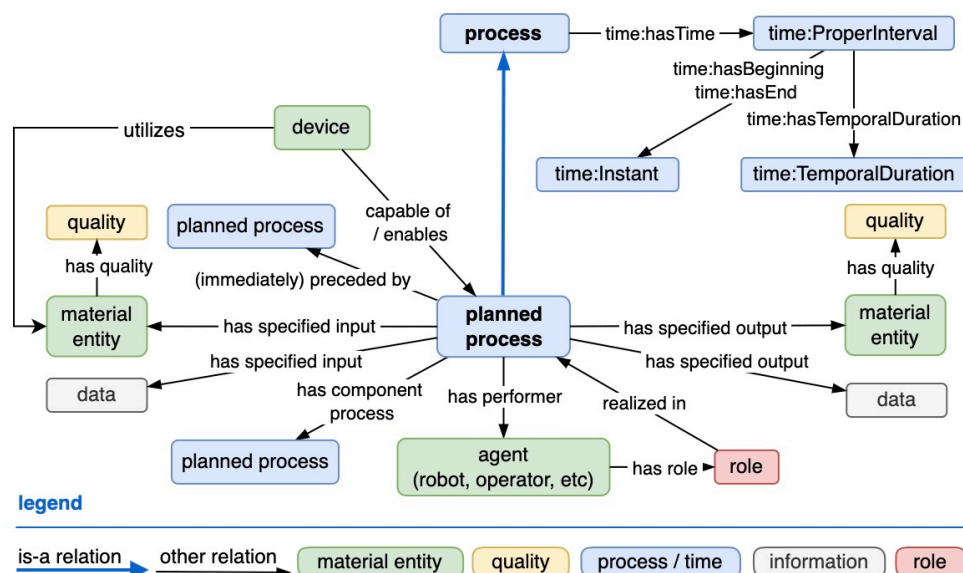


Fig. 11: OBO + OWL-time process modelling. Inputs and outputs can have qualities (or characteristics) attached to them which can change from input to output.

OBO lacks a generic “step” planned process entity, although the member candidate PROcess Chemistry Ontology (PROCO)[48] has “unit operation in chemical processing” (PROCO:0000051) which looks promising for food applications (Fig. 10), and appears equivalent to PO2’s step described below. Note however that the definition and comment in Fig. 10 goes beyond chemistry domain, so a revision of the label or definition scope is needed. The term “step” (NCIT:C48176) can be found in OBO Foundry member NCIT Thesaurus vocabulary[49] but this vocabulary has not been harmonized with the upper-level COB or BFO classes. These issues illustrate curational dilemmas that occasionally occur within OBO.

Additionally, the OBO plan specification could benefit from an ordered list construct such as the SemanticScience Integrated Ontology’s “list” (SIO:000150) within which to organize steps. As well although there are many matches for an OBO Foundry search on “observation” as a process or data item,

none provides a micromodel containing participants, measures, and time, so this needs to be developed to record process step features. The PO2 review illustrates work in this area.

### 3.1.2. Realization of planned process

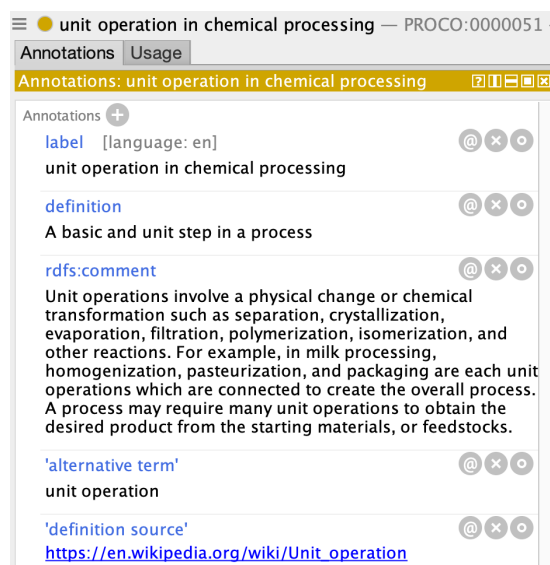


Fig. 10: Protégé view of PROCO unit operation.

OBI skirts around modelling failed processes due to the complexity in writing supporting definitions and axioms. However, COB enables a new semantic for planned process (COB:0000082) which allows for the intention to realize some plan. To indicate success or failure of a process instance, it can be typed as one of two subordinate classes: “completely executed planned process” or “failed planned process”. There is no “work in progress” state.

The instance vs. class distinction arises when looking at OBO entities for general or more specific plans. OBI defines a vague sense of “plan” (OBI:0000260) as the expression of a general intention of an agent to achieve an objective, but largely avoids using this in favor of IAO plan specification which is designed to capture a sense of a step-by-step plan; as well OBI provides “protocol” (OBI\_0000272), “A plan specification which has sufficient level of detail and quantitative information to communicate it between investigation agents, so that different investigation agents will reliably be able to independently reproduce the process.” OBO currently does not specify the data structure details for this. Regardless, an instance of a plan specification or protocol - which details the needed device(s), settings and material or digital inputs - is executed by a process, and that instance presumably can reflect refinements regarding the possible tools at hand and their operation.

BFO 1.0, 2.0 and 2020 have a “one dimensional temporal region” which exemplifies “the temporal region during which a process occurs”. Its BFO 2020 subclass, a continuous “temporal interval”, can have a processes attached to it via the “occupies temporal region” object property, and can have “has first instant” and “has last instant” properties that enable calculation of

durations. Meanwhile, from past work, IAO has “is duration of” to connect a time measurement datum number and unit to a process. These two patterns are not synchronized, so we have been drawn to reuse of the OWL-Time ontology instead.

### 3.2. OWL-Time

A few basic time components are needed in a process ontology model:

- An ability to express its starting and ending instants.
- An ability to express the duration of the time interval during which it occurs. This can be expressed declaratively without necessarily having starting and ending times stated. For example, “boiling an egg requires a duration of 3 minutes”.
- Flexibility in the granularity of how durations are expressed (in days, seconds, weeks, perhaps even moons, tides, and with minimum and maximum values).
- Ideally an ability to deal with time adjacency as a function of scale precision. Two day-long concerts a week apart become adjacent when positioned in a 52 week year, and may map to a single month if viewed in months.

Most of this functionality can be supplied by OWL-Time[50], which is in widespread use. Illustrated in Fig. 12, its overarching `time:TemporalEntity` could be connected to any entity using the “`time:hasTime`” object property. `TemporalEntity` is either a `time:Interval` with a different start and end time, or a `time:Instant` (where start and end time are the same). `Interval` specializes into `time:ProperInterval`, backed up by Allen

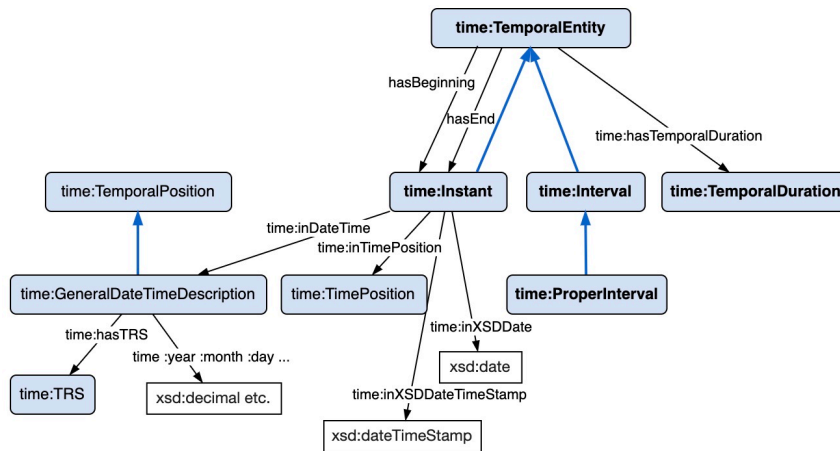


Fig. 12: OWL-Time Temporal Entity subclasses.

“algebra of binary relations on intervals”[51] to establish temporal proximity of intervals.

A BFO process could naturally connect with `time:hasTime` to a `ProperInterval`, thus gaining access to `time:hasBeginning` and `time:hasEnd` object properties for specifying corresponding `time:Instant` objects (which in turn are specified with `xsd:date` or `xsd:dateTimeStamp` datatypes). A

based on instance data that includes process start or stop times.

One modelling choice is whether to utilize the `time:ProperInterval` class to structure dependencies between processes. The alternative is to attach statements about process dependency directly to the processes themselves. In that scenario, RO is used to state that process B “ends with” process C, or process A “immediately precedes” B, as in the boiling carrots recipe model below. It would seem like a more intentional design to express dependencies on the processes themselves, and leave reasoning to infer impacts on associated OWL-Time namespace objects.

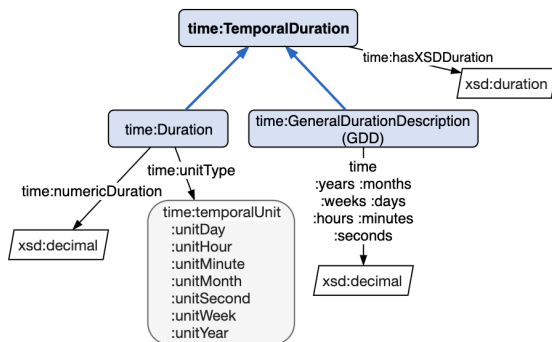


Fig. 13: Using OWL-Time to express duration.

`ProperInterval` also gets a `time:TemporalDuration` (Fig. 13) which has a variety of value formats. Note that unless a process is explicitly triggered by specific calendar dates (for example a Remembrance Day ceremonial process), modelling can occur entirely with reference to durations, and temporal dependencies between processes can be later calculated or reasoned

This raises the question of whether or not any completed process can be reasoned to be a type of `time:ProperInterval`, since such a process has exactly one beginning and end time, and a duration. If so, that would mean the arm’s length “process hasTime some `ProperInterval`” relation would no longer be needed, and RO’s Allen relations and OWL-Time’s Allen relations would have equivalent domains and ranges – and therefore could be unified. However, this simplification assumes that a process only inhabits a proper interval, rather than a `time:Interval` (or BFO one-dimensional-

temporal region) that may have gaps or breaks in a timeline. Further work on modelling aggregated processes, for example smoking sessions that occur over time (a focus area of the Human Behaviour Change Project[52]) will be needed for clarity here. Either way, a rule engine (driven by SWRL rules for example) can use OWL-Time to calculate Gantt chart interval dependencies and time points within a process network.

### 3.3. Exact2 (EXperimental ACTions) ontology

The Exact 2 ontology[8], a prototype designed with reuse of OBO terms, describes the process steps of an experimental protocol in order to enable reproducible experiments. As OBI “protocol” refrains from defining how it might detail necessary steps / processes especially as they involve time modelling (see OBI “waiting” term curator note), Exact 2 defines an intermediate level of process model which contains general “equipment” and “biochemical entity” terms, and over 90 “experimental actions” one finds in a laboratory context. Besides OBI input and output relations (Fig. 14), Exact2 introduces a general “has proposition” object relation that connects an action to various contextual objects such as expressing a

condition for an action to proceed, a duration, or a “protocol method” to follow.

Exact 2 has a few OBO Foundry related differences:

- PATO qualities like temperature and color are attached directly to actions; OBO has only recently allowed characteristics to be associated with processes by way of a “regulates characteristic” object property.
- Exact 2 introduces “optional descriptor of experimental action” to convey that an action had optional descriptors in the belief that OWL does not support optional features; however this capability is available within OWL by using cardinality statements[53] (e.g. “person ‘has part’ ‘min 0’ hair”).
- OBO doesn’t have a model for expressing condition(s) for an action, while duration and protocol are attached differently to processes.

Exact 2 avoids investing in a detailed ontology process model because of the problem of maintaining identity of process inputs and outputs mentioned in the OWL logic discussion section. Instead, the Exact 2 project proposes incorporating Petri net modelling terminology that allows discrete state changes, and is often used in computer science and engineering modelling. What we foresee is that the same functionality can be achieved by modelling processes as the loci of state change, with input and output materials, energy, time, and participating devices and agents as factors whose characteristics can be stated to change as a result of a process. Instances of these can then be networked together via input/output relations. In either case rule engines or other

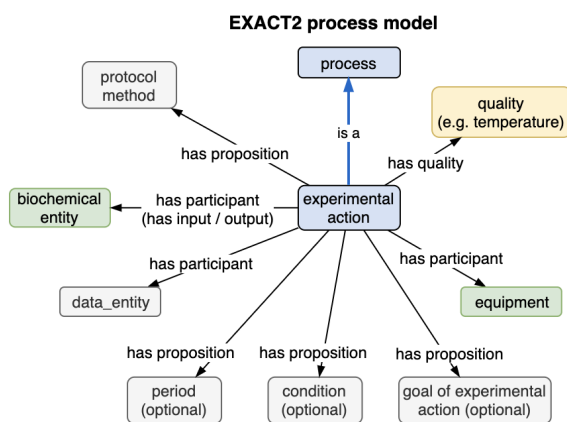


Fig. 14: The Exact2 process model.

transformative queries are introduced to document or simulate the properties of instances of a process model and its transformed materials over time, as noted in section 2.1.5.

Exact 2 uses PATO to describe typical qualities like volume, speed, temperature that are measured or manipulated in an action. Crucially, the model has been validated on real-world experimental protocols, recognizing 83-95% of protocol actions.

### 3.4. PROV-O: The provenance ontology

PROV-O, a W3C standard OWL ontology is often examined for process modelling because of its stated ambition of providing *“the core concepts of identifying an object, attributing the object to person or entity, and representing processing steps”* in order to represent physical or digital object origins. PROV-O aspires to capture provenance information contained in plans (such as workflows or instructions) *“which contain descriptions of the entities and activities involved in producing and delivering or otherwise influencing a given object.”*[54] This close resemblance to a generic process ontology has drawn many projects to use PROV-O as one despite the absence of key process model features.

- PROV-O `wasInformedBy` attaches between two processes to show a loose semantic input connection akin to the more concrete RO “directly provides input for” object property.
- Because OWL-Time was under development as PROV-O was released, PROV-O was impelled to introduce its own time relations. It does not have an Activity duration relation.
- PROV-O does not seem to have a relation for breaking a process into parts, in

contrast to RO’s “has component process”.

- PROV-O Role of Agent in an Activity is equivalent to RO “role of” participant in a process.
- While PROV-O treats software as having an agent role in an activity, IAO/OBI has a computer be a device participating in a process of executing software.
- PROV-O provides a way for an Activity to be linked to an Association class, and in turn, to an Agent, and by way of the “hasPlan” object property, to a plan. OBI in contrast has a planned process which executes a plan specification or protocol. Agents are associated with a process by way of the ‘has performer’ property.
- PROV-O and SOSA (see sec 3.5) have been aligned [55] such that a `sosa:Procedure` can be a subclass of `prov:Plan` and `sosa:Observation` can be a subclass of `prov:Activity`.
- In line with its provenance mission, PROV-O is backward looking, as illustrated by the many past tense verbs in its relations. This makes it sound awkward with respect to process modelling where the focus is on possible outcomes resulting from variable state manipulation.

An example book publishing model[56] (Fig. 16) describes key PROV-O relations. An overall entity reference is created for the document that links to its various “specialization” states. Anonymous (blank) nodes encapsulate short statements that connect the document in its state before and after being published. PROV-O advocates point out that such nodes could be typed if desired, so we could have instances of classes like “draft book” or “printed book”. Fig. 17 offers a brief OBO process-oriented

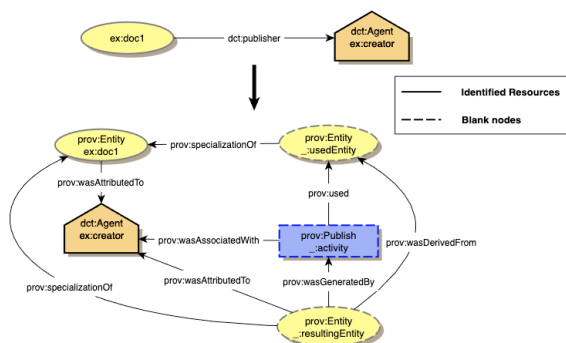


Fig. 16: A PROV-O book publishing model.

Credit: Figure 1 of <https://www.w3.org/TR/2013/NOTE-prov-dc-20130430> Copyright © 2013 W3C® (MIT, ERCIM, Keio, Beihang), All Rights Reserved. W3C liability, trademark and document use rules apply

comparison, where the book publishing process is performed by a publisher (Fig. 17). A particular draft book is input to a publishing process with both input and output book entities tracked informatically by book identifier. Note however an RO issue: if the draft book is in digital form, the shortcut “derives from” relation does not apply - it only attaches between material entities. Technically, using BFO “inheres in” might work but is problematic if editing has substantially changed the end product.

### 3.5. SOSA/SSN model

Introduced in 2018, the Sensor, Observation, Sample, and Actuator (SOSA) ontology [57] is “*designed to provide a flexible but coherent perspective for representing the entities, relations, and activities involved in sensing, sampling, and actuation*”. SOSA approaches modelling the process of observations with a lower-level set of relations detailing “observations” or “sampling” of features of interest, and the sensors that generate them (Fig. 15).

SOSA is the core of an updated Semantic Sensor Network (SSN) ontology which covers the organization of sensors and actuators into systems, and the detailing of their operating characteristics. In the revision

of SOSA/SSN observations are now conceived as acts or events. The core,

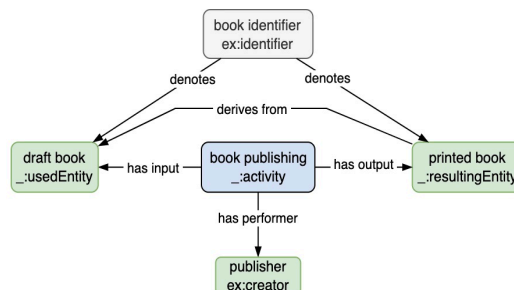


Fig. 17: An OBO book publishing model.

consistent with observation models in other ontologies, provides a common pattern for observation, actuation and sampling which is aligned with PROV-O[13]. However, it represents a break with the original SSN ontology that built upon the SSO pattern in which an observation was effectively a record or description of an observation context (a kind of `dul:Situation`), rather than an activity in the world (a `dul:Event`). For further reading on using events versus records for observations, refer to [13].

By design, `sosa:FeatureOfInterest` sidesteps the direct naming of the thing whose feature is being sensed (a domain-specific entity and its qualities), leaving it up to 3rd party ontologies to do so. A “`sosa:Observation sosa:hasFeatureOfInterest some sosa:FeatureOfInterest`” translates to

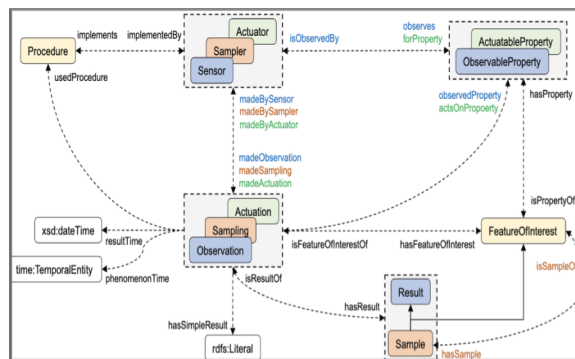


Fig. 15: The SOSA model. CREDIT: (sent request to <https://www.researchgate.net/publication/325644558>).

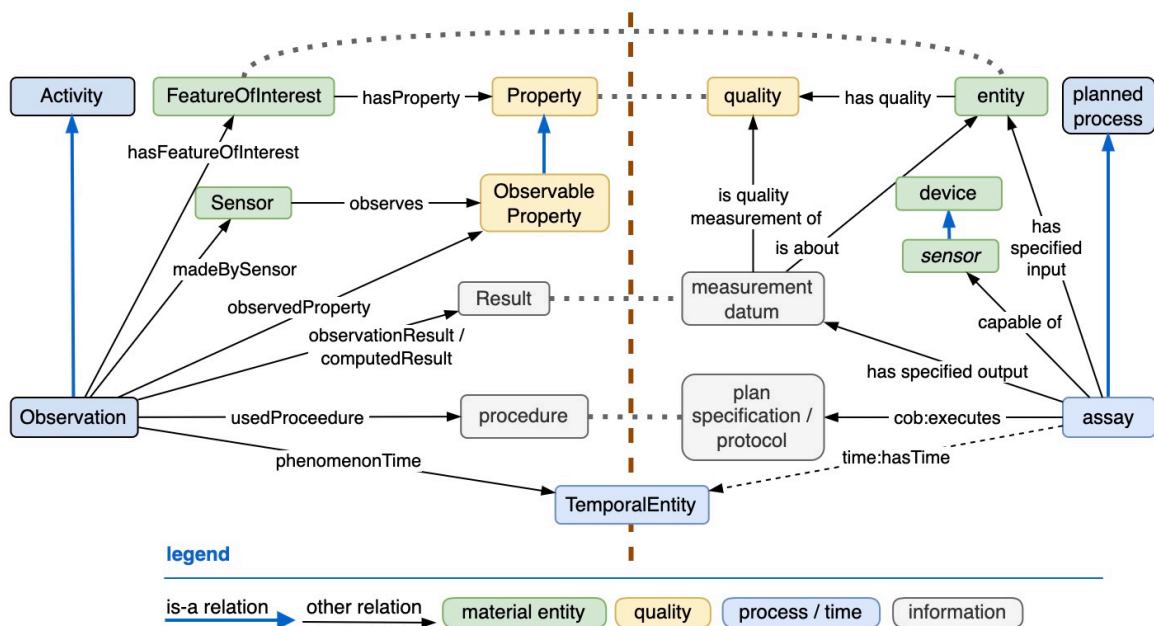


Fig. 18: Comparison of SOSA Observation and OBO time-enhanced process.

OBI “assay and has\_specified\_input some (material entity and has role some evaluant role)” where the material entity is mereologically related to some entity of interest, or has qualities of interest. OBO process model keeps general relation names in play rather than specializing them with “observation/sampling/actuation”.

OBO measures an entity’s quality, but how detailed the location of the quality (e.g. hair color, top of head=grey, sideburns = brown) is similarly left to implementers to structure. If in the life sciences, OBO likely has more capability amongst its member ontologies for describing entities and their qualities (mainly via PATO). Loose equivalencies between SOSA and OBO models are shown to left and right of the dotted grey line in Fig. 18. Time relations on the OBO side are tentative as OBO has no recommendation about such usage.

SOSA “sosa:Result” and IAO “measurement datum” are comparable as they both allow the output of an observation process to be a measure or computation.

“sosa:phenomenonTime” pertains to the (OWL-Time) instant or duration that the characteristic measure was true of the FeatureOfInterest, rather than the observational process. OBO may benefit from a similar distinction where the end of an observation process doesn’t coincide with the time the measurement is accurate for. As a kind of assay, OBO would also benefit from an explicit “observation” micromodel pattern of observation time and other contextual information.

### 3.6. Process and Observation Ontology

Based partly on the Sensor, Observation, Sample, and Actuator (SOSA) ontology [13], but also situated within the BFO hierarchy, PO2 is designed to monitor industrial food processing, and describe food formulation.

PO2 can represent a food transformation process described by a set of experimental observations available at different scales and evolving in time through the different unit operations of a production process. The

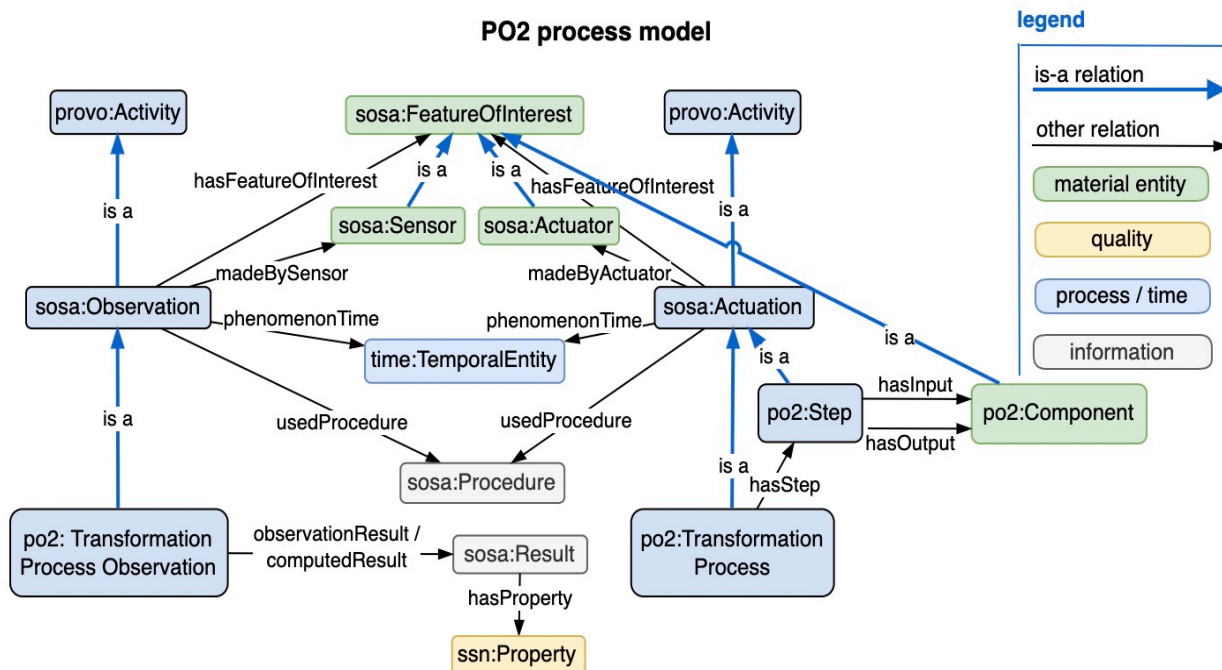


Fig. 19: Main classes of PO2 and the relationships between them.

initial model called MS2O (Multi-scale Multi-step Ontology) was built to represent a bioprocess of yeast production and stabilization [58] and further developed to fit the need of dairy gel manufacturing and the integration of heterogeneous data [59,60]. Finally, the core model of PO2 was reengineered with SOSA/SSN to be able to adapt to any transformation process.

The design of PO2 ontology meets three essential needs:

- Representing foods and their characteristics such as nutritional and texture data, and sample information.
- Representing processes that "link" food characteristics with their recipe to help product formulation.
- Representing and integrating common patterns in different bioresource transformation domains.

Therefore, the ontology is designed in two layers: a core layer and a domain layer. The core ontology PO2 is dedicated to the generic modeling of both transformation processes and characterization processes (called transformation process observations). PO2 core model reuses various existing ontologies: BFO, SOSA, IAO, Time Ontology and QUDT. PO2 Dairy gel[61] and biorefinery[62] domains are publicly available, and other private and public domains representing the characteristics of foods during manufacturing are complete or in development.

Fig. 19 shows the main classes of PO2 and the relationships between them, including:

- The *po2:Step* class represents a unit operation of a transformation process; it is a subclass of the *sosa:Actuation* class, defined by the



W3C/OGC standard as an act that executes an (actuation) Procedure to change the state of the world using a `sosa:Actuator`.

- The `po2:Transformation Process` class represents the chain of unit operations (called “steps”) of a transformation process; it is a subclass of both `sosa:Actuation` and `bfo:process`.
- The `po2:Component` class represents objects (or material entities) which are sampled or transformed in a process; it is a subclass of both `sosa:Feature of Interest` and `bfo:material entity`.
- The class `sosa:Result` represents the result (the value) of an observation and link it to `ssn:Property`; in the PO2 model, `sosa:Result` is a subclass of `iao:measurement datum`
- The `po2:Transformation Process Observation` class links the unit operations (elementary steps), the participants (components) and the results (values); it is a subclass of the `sosa:Observation` class, which executes the observation procedure, and a subclass of `bfo:process`.
- The `sosa:Procedure` class represents a protocol, an algorithm or a calculation method specifying how to make an observation, create a sample or modify a state. In the PO2 model, it is a subclass of `iao:information content entity`
- The `sosa:Actuator` class represents a device that is used by, or implements, a (actuation) procedure that changes the state of the world, namely realizing a transformation; it is a subclass of both `sosa:Feature of Interest` and `bfo:material entity`.

- The `sosa:Sensor` class represents a device, agent (including humans), or software (simulation) involved in, or implementing, a `sosa:Procedure`. Sensors respond to some input and generate a `sosa:Result`.
- The `time:TemporalEntity` class is used to represent the sequence of steps which compose the process.

## 4. DISCUSSION

### 4.1. Process Relation Comparison

Harmonizing the language of semantic web content such that queries can traverse it without needing translation is a future ideal to incrementally progress towards. Having more participants agree to use stand-alone components like OWL-Time helps this effort. Ideally a single set of general process ontology design patterns that work across domains could be adopted, but so far none of the design choices have universal appeal so they must be hashed out to achieve greater consensus. Table 1 surfaces various classes and relations that our compared ontologies have for process modelling. Discussion and our recommendations for consideration by the OBO community are provided below.

#### 4.1.1. OBO – OWL-Time

Rather than use OBO’s interval and duration expressions, we suggest OWL-Time `hasTime`, `hasDuration`, `hasBeginning`, and `hasEnd` object relations, as well as `Instant`, `ProperInterval` and `Duration` classes. Upcoming COB work will provide guidelines for using OWL-Time data properties in a limited way, such as `inXSDDate` and perhaps `numericDuration` and `unitType`. Additionally, if OBI ‘planned process’ can be logically typed as a proper

interval then OWL-Time's Allen relations could be considered for direct use.

#### 4.1.2. OBO – PROV-O

We were unable to find any additional PROV-O components needed for an OBO process ontology, with some provisos. As shown in Table 1, most PROV-O object properties have an equivalent OBO property, or are handled in a different way. Regarding PROV-O's "value" data property[63] system, COB data properties are not finalized, but there will be a "has value" object property with possible parts including "has quantity", "has unit", and "has category". OWL-Time is used instead of PROV-O's time relations.

#### 4.1.3. OBO – SOSA/SSN

One design choice exposed in our comparison is a preference for minimized object properties, which is illustrated by contrasting SOSA's specific relations "madeObservation", "madeSampling", and "madeActuation" (each linking to a type of process output) with OBI "has\_specified\_output". One can query

```
“SELECT X sosa:madeObservation Y”
```

and know X,Y match Sensor and Observation respectively. This could be generalized to the following query, which needs one constraint on the subject, making it a touch more complex:

```
“SELECT (sosa:Sensor X)  
OBI:has_specified_output Y”
```

This assumes a sensor only outputs Observations - if a sensor also outputs other types of things, then the object must also be filtered. This simpler language allows the same richness of class axiomatization (for

example, "Sensor: 'has\_specified\_output' exactly 1 Observation" controls output range and cardinality), without needing a learning curve when traversing domains. The minimized object property approach is also translatable to the "liberal" object property approach which may have particular application requirements in terms of database or computational efficiency.

OBO has coverage for SOSA process output relations, but it needs sensor, actuator, and sampler device classes for industrial automation modelling, and it does not have an actuation process equivalent.

#### 4.1.4. OBO – PO2

In addition to the above process model components inherited from SOSA that PO2 reuses, PO2's "Step" process is needed for grouping processes together into a unit operation as described in the next section.

#### 4.1.5. Synthesis: Process modeling of recipes

As a synthesis of the above process model discussion, we turn to the use-case example of modelling recipes which are generally a set of instructions that describe how to prepare or make something. We focus on prepared dish or food recipes which often include a list of ingredients to obtain, in the order of their use and/or listed by quantity, the necessary equipment used for the dish,

Entity Type	OWL-Time	PROV-O	SOSA / SSN	PO2	OBO
<b>Class</b>					
occurent	Instant	InstantaneousEvent			BFO:zero dimensional temporal region
occurent	ProperInterval				BFO 2020:temporal interval
occurent	TemporalDuration				BFO:one dimensional temporal region
process		prov:activity		PO2:Transformation process	OBI:planned process
process				PO2:Step	Missing
process			sosa:Actuation	sosa:Actuation	Missing
process			sosa:Sampling		OBI:material sampling process
process			sosa:Observation	sosa:Observation	OBI:assay
characteristic			ssn:Property	ssn:Property	PATO:quality
characteristic					PATO:energy
material entity			sosa:Sensor	sosa:Sensor	Missing
material entity			sosa:Actuator	sosa:Actuator	Missing
material entity			sosa:Sampler		Missing
material entity				PO2:Component	FOODON:food material
material entity			sosa:FeatureOfInterest	sosa:FeatureOfInterest	BFO:material entity
data structure			sosa:Procedure	sosa:Procedure	IAO:plan specification / OBI:protocol
data structure			sosa:Result	sosa:Result	IAO:measurement datum
data structure				PO2:scale	
<b>Data Property</b>					
		prov:atTime			Unused
		prov:startedAtTime			Unused
		prov:endedAtTime			Unused
		prov:value			OBI:has specified value / owl:hasValue COB data properties are not finalized
<b>Object Property</b>					
	hasTime				Missing. IAO:has time stamp is limited to instants for domain, and time measurement datums for range.
	hasBeginning				Missing
	hasEnd				Missing
	hasTemporalDuration				Missing. IAO:is duration of is similar but requires a time measurement datum to express duration
		E2 wasDerivedFrom E1		E1 isComposedOf E2	E2 derives from E1; E2
		P used E		P1 has input E1	P has input E; P has specified input E P has primary input E
					P1 directly provides input for P2
		E wasAttributedTo A			E produced by A
		E wasGeneratedBy P		P1 has output E2	P has output E; P has specified output E P has primary output E
		A wasAssociatedWith P			P has participant A
		P1 wasInformedBy P2		P hasStep P1 P1 hasForSubStep P2	P1 has component process P2
		E1 specializationOf E			ID denotes E1, E2 etc.
	P2 intervalAfter P1			P2 time:intervalAfter P1	P2 preceded by P1
	P2 intervalMetBy P1				P2 immediately preceded by P1
	P1 intervalStartedBy P2				P1 starts with P2
	P1 intervalEndedBy P2				P1 ends with P2
		R sosa:hasProperty C		R sosa:hasProperty C	E has quality C
					I is about C; I is quality measurement of C
		O observedProperty C			No direct equivalent.
		O sosa:hasResult R		O sosa:hasResult R	P has specified output E (as above)
<b>Legend</b>					
A: Agent C: Characteristic / Quality D:Date/Time E, E1, E2: Entity / Component I: Information / Observation ID: Identifier O: Observation P, P1, P2: Process / Action / Step R: Result					

Table 1: A gap analysis of selected OWL ontology process related terms

and cooking advice in step-by-step instructions, and duration of preparation and cooking. This use-case demonstrates OBO process model components in such a way that application to other domains such as pesticide or paint formulation in a lab or manufacturing plant is easily envisaged too.

Specific to the food domain, a recipe may also indicate the number of servings that the dish provides (yield), serving procedures and culinary variations. Estimated or analytically tested nutritional information, including the number of calories or help for dietary restrictions may be provided for a serving portion. Sometimes, the origin or history of the dish is included, as well as images or videos. This content has often been narrative in nature and indeed constitutes an entire book genre.

Generally, an ingredient (aka component) is a substance that forms part of a mixture. In most developed countries, legislation requires that ingredients be listed according to their relative weight in a food product. If an ingredient itself consists of more than one ingredient, then that ingredient is listed by what percentage of the total product it occupies, with its own ingredients displayed next to it in brackets. Industrial standards such as the GS1 webvoc provide support for listing ingredients in this context. Country and trading-block level regulations also control reference to additives as ingredients, and to nutritional content labelling.

In the realm of household kitchens, food material ingredients provide input to a cooking process which (by way of a household chef) executes a recipe and creates food products having organoleptic properties. An industrial processing paradigm brings more formality to the language: A **formulation** provides the “criteria for selection of ingredients” together

with the process of incorporating selected ingredients into foods. Its fundamental meaning is the putting together of components in appropriate relationships or structures according to a formula. Comparable to home cooking and its recipe steps, commercial “experimental” or “pilot” food preparation involves a matrix of process step variations to an existing food product formulation in order to choose a final a procedure for industrial batch or continuous processing.

Some ingredients (including additives) impart specific properties to a formulation - qualities or functionalities required or desired in the final food (for example, sensorial or nutritional, or related to stability and shelf life, or necessary to adjust processes according to variability of raw materials). Here especially, a vocabulary of food characteristics is required so that food transformation objectives can be stated, as well as standardizing language for grading food to guarantee regular quality and yield. In terms of OBO development, this alone requires a whole new universe of micromodel vocabulary in PATO (attribute) and is not addressed in our process modelling.

Devices (equipment) are used by operators (human or robot) as part of the planned process to execute a plan specification that achieves formulation objectives. Sensors, a kind of device, are used in cooking as with other planned processes to determine process step completion, and they can be electronic, mechanical, or human (for example, sight, hearing, heat sensitivity). Sometimes energy is considered a process input or output directly via a device or ingredient; other paradigms consider it a quality of an input.

Recipes are widely available on the World Wide Web and are shared as part of social

networking applications. Schema.org, a popular resource for RDF patterns, defines a Recipe class markup, which, while not fully ontological, provides an RDF structure for its representation. Because of its adoption by several online recipe web sites, the Recipe class is one of the most highly instantiated classes on the World Wide Web. However, the underlying structure of schema.org is intended for information retrieval and as such prizes simplicity. Furthermore, schema.org is constantly evolving in its documented design, its consumption by search engines and in the actual implementation by users, which has led to several incompatible designs for modeling recipes with multiple levels of details (for example schema:CookAction and schema:HowToSection). The result is a data structure suited to informational retrieval algorithms with limited ontological logic.

The most common Recipe class implementation is as a list, or a single narrative (free-text) string, of the recipe steps with no possibility of representing concurrent processes unless expressed explicitly within the narrative. The sequence of these steps is implicit to their physical ordering within the serialization itself which is not guaranteed outside of the specific context of schema.org Search Engine Optimization documents. Executing a recipe step involves quantities of a subset of the ingredients, which implies a relationship between the set of ingredients and each step. In a completely ontological framework, each step would reference the same individual ingredient nodes as the recipe ingredient list to ensure proper entity resolution.

Schema Recipe does represent a widely-deployed base case for the requirements of consumer-grade recipes, so mapping to other standards should be pursued. To this end, we present our vision for an improvement on these limitations.

#### 4.2. Prototype FoodOn recipe model

FoodOn will incorporate a general food-related process model so that food processing methods can be detailed in a series of micromodels specific to each kind of food or process ranging from the apparently simple act of cooking or boiling, to following a

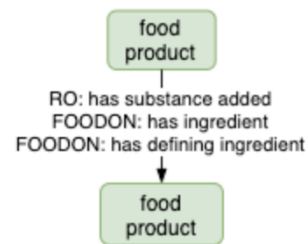


Fig. 20: Types of FoodOn ingredient.

consumer or industrial recipe process. Because of reasoning limitations, the emphasis here is more on finding value in data structures that conform to ontological distinctions about the world, rather than achieving reasoning prowess over the data itself. The design patterns below are mostly prototyped in diagram form to generate discussion about needed entities and relations, but a portion of them are currently being added to FoodOn.

Currently FoodOn has extensive vocabulary for single source ingredient composition - the taxonomy and anatomy of organism ingredients. For multi-component food composition, FoodOn allows food product classes and instances to refer to ingredients (themselves food products including additives) by way of “has ingredient” and “has defining ingredient” (Fig. 20), the latter pointing to an ingredient that a food product can’t do without, for example potato leek soup must minimally have those two ingredients. Ingredients are considered intentional additions; if a process unintentionally adds a substance to a food, the “has substance added” relation applies.

These ingredient relations alone might suffice for a shopping list but recipe ingredients are usually associated with steps, and must be ordered in product labelling, so we need an ingredient set to be transformable into ordered lists with quantities conveyed by rank, proportion or absolute value. Often a recipe also needs devices – even a simple layout table or spoon must be accounted for as a resource – which reveals the potential complexity of even small processes from a machine comprehension perspective. At the same time, vocabulary reuse at different scales of modeling - from kitchen to laboratory to robotics - hopefully falls into patterns of reuse, such that fewer relationships are needed as a generic process model develops, with the focus being more on defining new subclasses of process, material entity, device and information.

The proposed FoodOn recipe model in Fig. 21 has a number of new components (shown with dashed borders) including an “ingredient set”, a “device set” and an “instruction set” of step specifications which respectively allow reference to food materials, devices, and the planned processes they feed into. These sets will be subclasses of IAO data set (IAO:0000100) which is simply a set of things of the same kind. Some amount of instruction set recursion is needed where recipes have overall stages or steps, with each step itself potentially having an instruction set component. Each ingredient specification references a food material (product, intermediary product, or additive, or choice thereof) but also a proportional, count or scalar measure of it. Device capabilities are stated by the device specification (pot, pan or bowl size, dish oven temperature rating etc.). Free text commentary or instruction can potentially be linked to any of the class entities by using annotations such as “dc:description” or

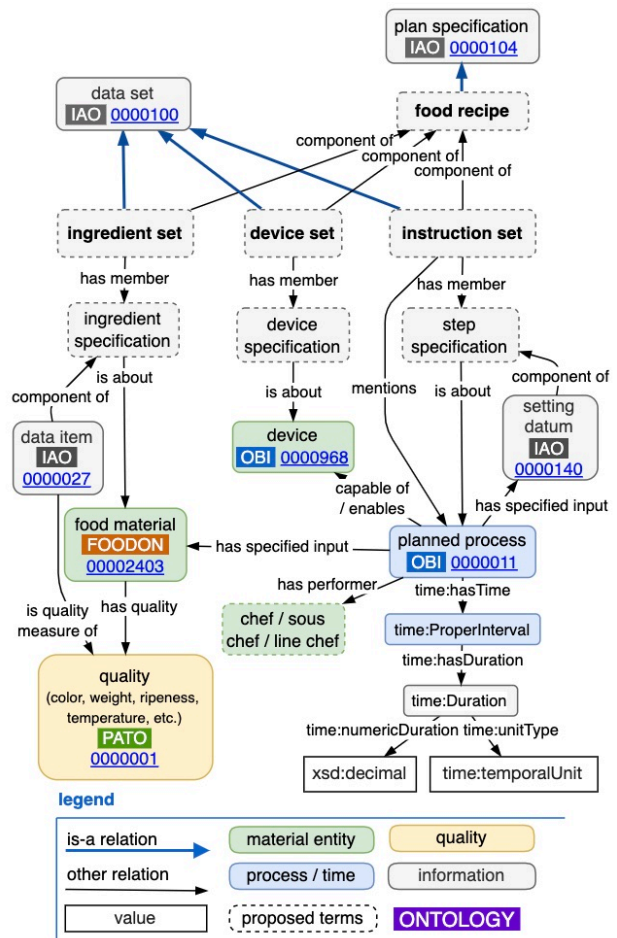


Fig. 21: A FoodOn recipe model prototype.

“rdfs:comment”. The recipe model currently does not include serving procedures or portions, or derived nutrition or cooking duration information.

One example of the IAO “mentions” object property is shown (between instruction set and planned process) as an optional shortcut, in the case where an ingredient, device or step specification is skipped. A hierarchy of “capable of part of”, “capable of”, and “enables” relations allows itemizing which devices (or tool or equipment) can partially or completely enable a planned process to produce output. The “has performer” object property references a person or other kind of

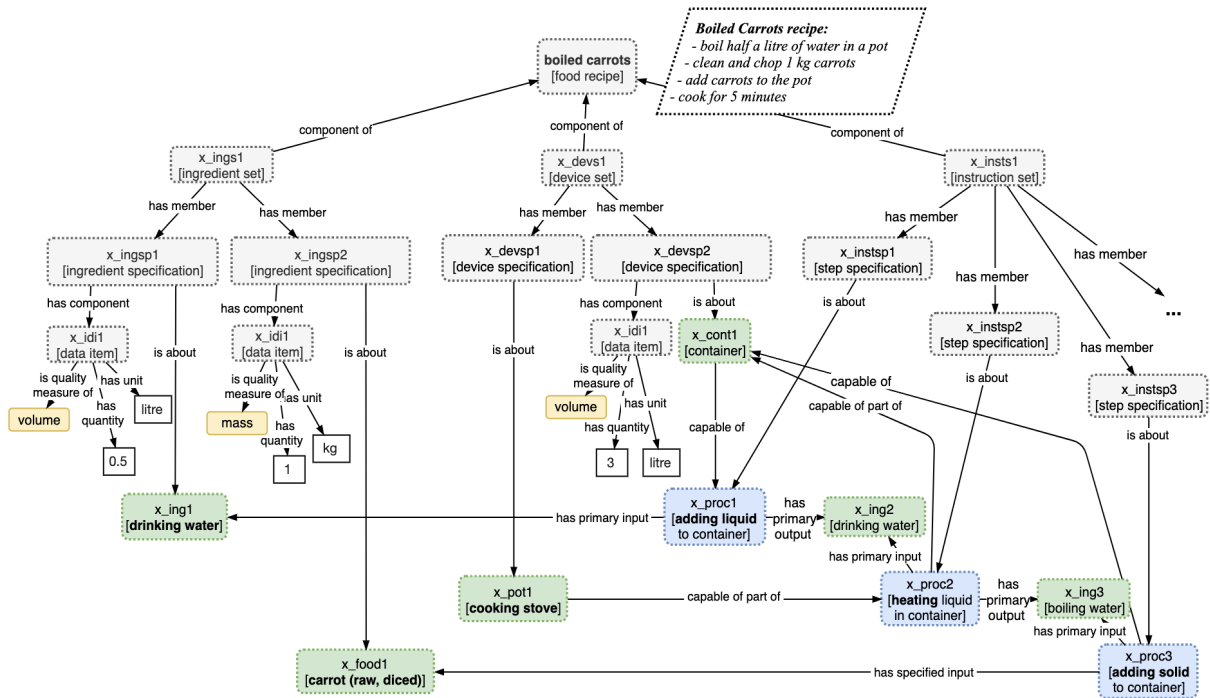


Fig. 22: Partial Boiled Carrots recipe plan specification.

agent/robot which operates devices and is accountable for the task.

We see how an OBO-centric model fares with a simple “boiled carrots” recipe example shown partially in Fig. 22, essentially “boil a pot of water, clean and chop carrots, add to the pot, cook for 5 minutes”, with a diagram shortcut insofar as it references already cleaned and chopped carrots. Volume and mass of ingredients and containers can be specified, and all inputs to processes.

Focusing in on the first step, Fig. 23 shows the general input/output of an “adding liquid to container” process, and then details about putting 0.5 litres of drinking water into a 3-litre pot, a process measured to last 1 minute. Instance-level statement entities are shown with a dotted border – these are specific parameters, settings, measures of ingredients or devices or process inputs or outputs that can be referenced directly in the specification elements of Fig. 22.

The result of the process is shown towards the right where `x_water2`, an instance of water, is located in `x_pot2` an instance of container. In this case `x_water2` appears one and the same as `x_water1`, and `x_pot2` appears to be the same as `x_pot1`. However, if the state has changed – say from a “`x_water1` not located in `x_pot1`” state, to a “`x_water1` located in `x_pot1`” state, then such states have to be isolated to avoid being unsatisfiable. For material tracking purposes, attaching tracking identifiers to these instances, for example an identifier which points to both `x_pot1` and `x_pot2`, may suffice, as described in the OWL reasoning section.

A research question is to what extent axioms and reasoning could be applied so that relations surrounding inputs such as `x_water1` could be selectively inferred to outputs such as `x_water2` as a result of attaching a common identifier to both. For example, can it be inferred that `x_water2` is

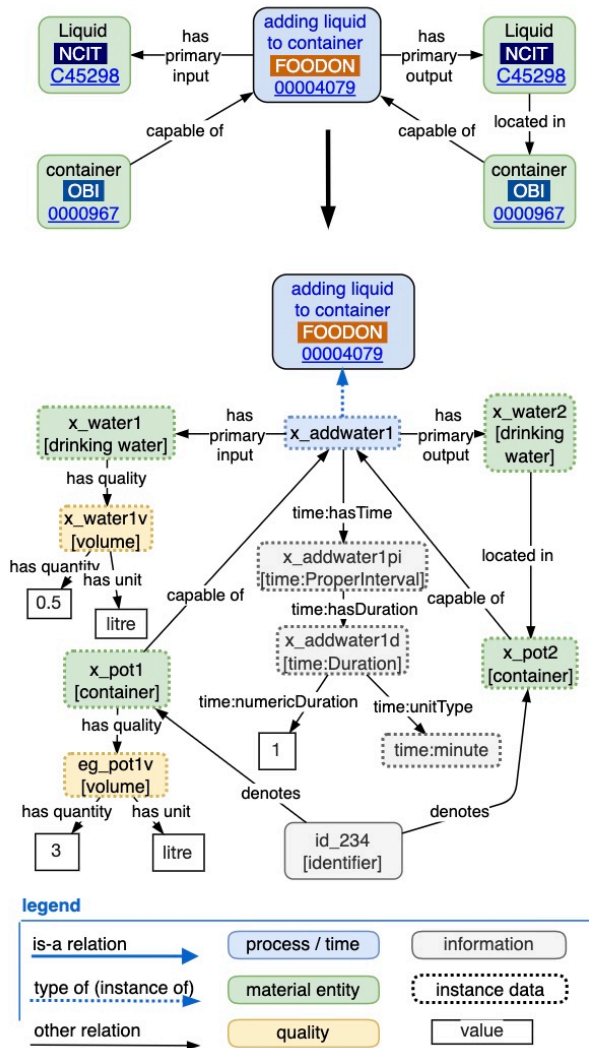


Fig. 23: Basic input/output model for putting water in a pot, and an instance-level example.

also drinking water, or that  $x\_water2$  volume is the same as the added volume? Additionally, a plan specification may dictate process durations and device specifications (like the 3-litre pot), but these measures can also be added to instance parts of a process as observed measures, thus providing an opportunity for reasoning over the satisfiability of real-world or simulated recipe experiments.

As shown in the Fig. 24 Protégé screenshot, process classes can have axioms

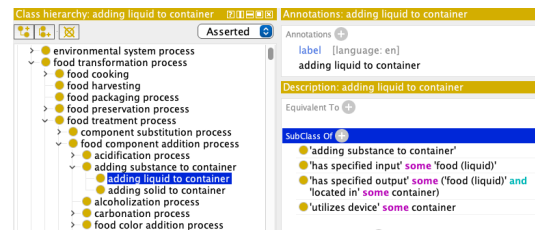


Fig. 24: Axioms indicate valid inputs and outputs and devices of the “adding liquid to container” process.

about specified inputs, outputs, and devices. Here “adding liquid to container” class mentions “food (liquid)” in its axioms.

The next step involves heating the pot until its water is boiling (Fig. 25). Here we need to acknowledge different ways of sensing the process end state - either by hearing the rumble of bubbles or whistle of a kettle, or by sight, or a thermometer. Measuring temperature is actually more complicated because of the varying boiling temperature based on atmospheric pressure, water salinity, et cetera. We can forestall that

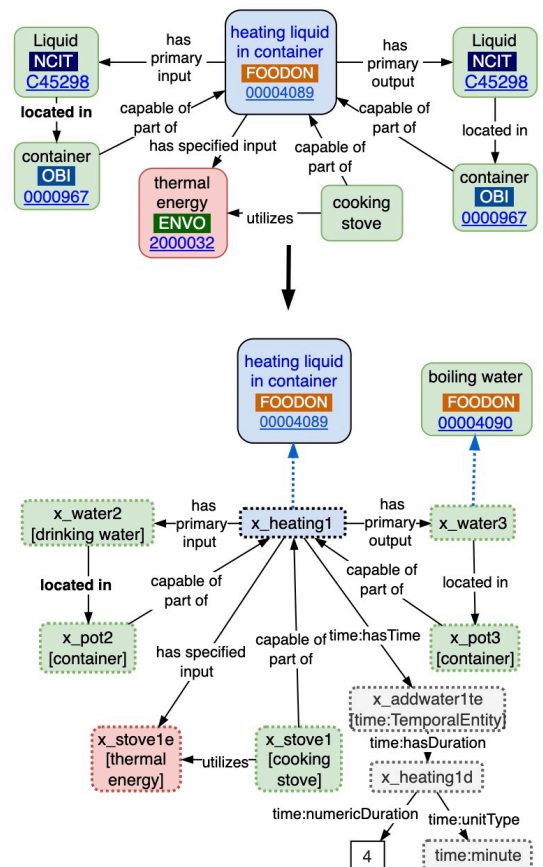


Fig. 25: Heating liquid in container.



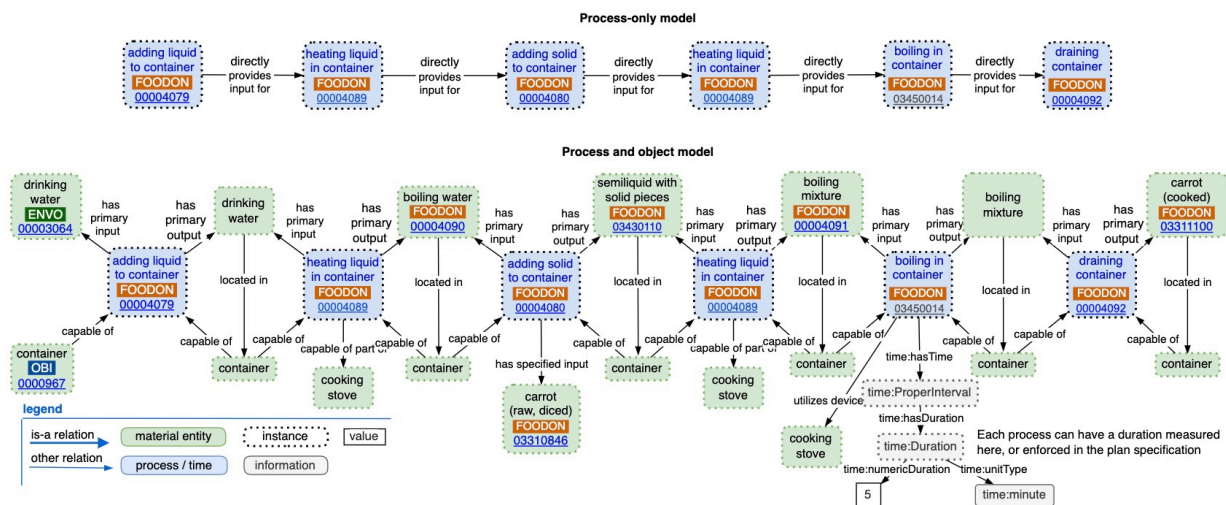


Fig. 26: Boiled carrots recipe process-oriented view, and the more detailed input / output view.

modeling challenge by just expressing that the output of the process should be a boiling water entity, and define that separately.

Similar steps follow, in adding carrots, bringing the pot back to a boiling state, keeping it “on the boil” - a process of adjusting heat input while counting down a duration - and draining. Fig. 26 illustrates how the entities from the various steps could plug together. Since each process needs to stand on its own, in a plug-and-play fashion, it must define all its input and output entities, and connect to its peer processes on those entities. Note that while OBI “has specified input” or output relation pertains only to planned processes, RO’s “has primary input” and output relations can pertain to both planned and unplanned processes. This is still a broad impression of a full-automation itinerary - the “unit operations” here would be broken down into many more steps for a robot to accomplish.

### 4.3. Food Process Ontology Requirements

The above recipe use-case yields an additional list of proposed entities and object

properties the authors of this paper recommend for FoodOn or OBO inclusion.

### Information content entity data sets:

- **Ingredient set:** A data set which contains ingredient specifications.
- **Device set:** A data set which contains one or more device specifications.
- **Instruction set:** A data set which contains step specifications.
- **Observation:** A data item which is the output of an observation process, and which has measures as components.

### Plan specification types:

- **Food recipe:** A plan specification for making a food product, which may have ingredient, device, and instruction set components.
- **Ingredient specification:** A plan specification which specifies a food material and its quantity or ratio.

- **Device specification:** A plan specification which is about some device and its settings.
- **Step specification:** A plan specification which is about some planned process and any restrictions on its participants.

**Object properties:**

- **Has ratio:** References a ratio data item as a numeric fraction or percentage. This will require numerator and denominator components to allow ingredients to be expressed as fractions (such as  $2/3$  rather than decimal 0.666).
- Ideally OBI ‘has specified output’ should be positioned under RO ‘has output’, and likewise ‘has specified input’ would fall under ‘has input’.

Kitchen utensils and equipment as well as an extensive list of food processes will also be required, but can be added incrementally to FoodOn or the Environment Ontology[64] (which houses manufactured products) or other ontology as needed.

This model can be advanced in a number of directions – by linkage to ingredient information such as nutrition, allergenic risk, ingredient substitutions, and ecological footprint assessment. Cultural practice and image elements could be included and device settings can be demonstrated. Another research task is to determine what logic or information is required to infer steps – clusters of one or more processes - from a recipe model dependency graph. Similarly, the development of more abstract recipe summaries may be possible by a transform

that yields top-level processes and all underlying ingredients. Validation of the model could be achieved by testing against the content of recipe database projects.

## 5. Conclusion

Process modelling is ultimately the narrative glue required to explain the history of material things and of derived information. Food processing - in agriculture, at point of harvest, and beyond - is a topic of immediate concern in an age of climate adaptation, global food processing and distribution, food deserts, and connections to food related health risk - obesity, malnourishment, toxicity, and foodborne pathogens. A consistent generic process ontology with specialized components will enable a clearer transition for both research and supply chain interoperability projects towards a healthier and more sustainable future. Our OBO Foundry-focused gap analysis about process modelling capability provides a roadmap for adding new globally-pertinent terms and beginning the construction of micromodels for recipe representation, and food related research and automation. Major areas of research remain in ontologizing the organoleptic and material science qualities of food (haptics, rheology, and molecular gastronomy for example) so that their transformation as a result of natural and planned processes can be described, graded formally, and predicted. We also expect our OBO process model work to be pertinent beyond the food domain. Feedback on food processing modeling requirements, broader process model semantics, and participation in this effort, is greatly welcome!

## 6. References

- [1] D. Knorr, M.A. Augustin, Food processing needs, advantages and misconceptions, *Trends Food Sci. Technol.* 108 (2021) 103–110. <https://doi.org/10.1016/j.tifs.2020.11.026>.
- [2] L. Elizabeth, P. Machado, M. Zinöcker, P. Baker, M. Lawrence, Ultra-processed foods and health outcomes: A narrative review, *Nutrients*. 12 (2020) 1955. <https://doi.org/10.3390/nu12071955>.
- [3] OriginTrail, GS1 digital link: A gateway towards trillions of digital twins —, OriginTrail. (2020). <https://medium.com/origintrail/gs1-digital-link-a-gateway-towards-trillions-of-digital-twins-8ce657122af1> (accessed March 3, 2022).
- [4] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, M. Tenorth, Robotic roommates making pancakes, in: 2011 11th IEEE-RAS International Conference on Humanoid Robots, 2011: pp. 529–536. <https://doi.org/10.1109/Humanoids.2011.6100855>.
- [5] R.C. Jackson, N. Matentzoglou, J.A. Overton, R. Vita, J.P. Balhoff, P.L. Buttigieg, S. Carbon, M. Courtot, A.D. Diehl, D. Dooley, W. Duncan, N.L. Harris, M.A. Haendel, S.E. Lewis, D.A. Natale, D. Osumi-Sutherland, A. Ruttenberg, L.M. Schriml, B. Smith, C.J. Stoeckert, N.A. Vasilevsky, R.L. Walls, J. Zheng, C.J. Mungall, B. Peters, OBO Foundry in 2021: Operationalizing Open Data Principles to Evaluate Ontologies, *BioRxiv*. (2021) 2021.06.01.446587. <https://doi.org/10.1101/2021.06.01.446587>.
- [6] D.M. Dooley, E.J. Griffiths, G.S. Gosal, P.L. Buttigieg, R. Hoehndorf, M.C. Lange, L.M. Schriml, F.S.L. Brinkman, W.W.L. Hsiao, FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration, *NPJ Sci Food*. 2 (2018) 23. <https://doi.org/10.1038/s41538-018-0032-6>.
- [7] A. Møller, J. Ireland, *LinguaL™ 2017 - Thesaurus*, 58562, 2018. <https://doi.org/10.13140/RG.2.2.23131.26404>.
- [8] L.N. Soldatova, D. Nadis, R.D. King, P.S. Basu, E. Haddi, V. Baumlé, N.J. Saunders, W. Marwan, B.B. Rudkin, EXACT2: the semantics of biomedical protocols, *BMC Bioinformatics*. 15 Suppl 14 (2014) S5. <https://doi.org/10.1186/1471-2105-15-S14-S5>.
- [9] S. Dervaux, J. Dibie, L. Ibanescu, J. Raad, PO<sup>2</sup> Process and Observation Ontology, (2021). <https://doi.org/10.15454/XSVVBW>.
- [10] A. Bandrowski, R. Brinkman, M. Brochhausen, M.H. Brush, B. Bug, M.C. Chibucos, K. Clancy, M. Courtot, D. Derom, M. Dumontier, L. Fan, J. Fostel, G. Fragoso, F. Gibson, A. Gonzalez-Beltran, M.A. Haendel, Y. He, M. Heiskanen, T. Hernandez-Boussard, M. Jensen, Y. Lin, A.L. Lister, P. Lord, J. Malone, E. Manduchi, M. McGee, N. Morrison, J.A. Overton, H. Parkinson, B. Peters, P. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, R.H. Scheuermann, D. Schober, B. Smith, L.N. Soldatova, C.J. Stoeckert Jr, C.F. Taylor, C. Torniai, J.A. Turner, R. Vita, P.L. Whetzel, J. Zheng, The Ontology for Biomedical Investigations, *PLoS One*. 11 (2016) e0154556. <https://doi.org/10.1371/journal.pone.0154556>.
- [11] P. Missier, K. Belhajjame, J. Cheney, The W3C PROV family of specifications for modelling provenance metadata, in: Proceedings of the 16th International Conference on Extending Database Technology, Association for Computing Machinery, New York, NY, USA, 2013: pp. 773–776. <https://doi.org/10.1145/2452376.2452478>.
- [12] S. Little, C. Cox, Extensions to the OWL-Time Ontology - entity relations, (2020). <https://www.w3.org/TR/vocab-owl-time-rel/>.
- [13] K. Janowicz, A. Haller, S.J.D. Cox, D. Le Phuoc, M. Lefrançois, SOSA: A lightweight ontology for sensors, observations, samples, and actuators, *Journal of Web Semantics*. 56 (2019) 1–10. <https://doi.org/10.1016/j.websem.2018.06.003>.
- [14] A. Ławrynowicz, A. Wróblewska, W.T. Adrian, B. Kulczyński, A. Gramza-Michałowska, Food recipe ingredient substitution ontology design pattern, *Sensors (Basel)*. 22 (2022) 1095. <https://doi.org/10.3390/s22031095>.
- [15] P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, V. Presutti, eds., *Ontology engineering with ontology design patterns*, 1st ed., Akademische Verlagsgesellschaft AKA, Berlin, Germany, 2016.
- [16] Submissions:Food recipe ingredient substitution ontology design pattern - odp, (n.d.). [http://ontologydesignpatterns.org/wiki/Submissions:Food\\_Recipe\\_Ingredient\\_Substitution\\_Ontology\\_Design\\_Pattern](http://ontologydesignpatterns.org/wiki/Submissions:Food_Recipe_Ingredient_Substitution_Ontology_Design_Pattern) (accessed March 8, 2022).
- [17] S. Peroni, G. Lodi, L. Asprino, A. Gangemi, V. Presutti, FOOD: FOod in Open Data, in: *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2016: pp. 168–176. [https://doi.org/10.1007/978-3-319-46547-0\\_18](https://doi.org/10.1007/978-3-319-46547-0_18).
- [18] AGROVOC thesaurus, (n.d.). <https://data.apps.fao.org/catalog/organization/agrovoc> (accessed March 5, 2022).
- [19] C. Mungall, agrovoc-obo: Agrovoc alignment to OBO Ontologies, Github, n.d. <https://github.com/cmungall/agrovoc-obo> (accessed March 5, 2022).
- [20] N. Azlinayati, A. Manaf, Exploring the relationships between OWL and SKOS, (n.d.). <http://www.cs.man.ac.uk/~stevensr/papers/iswc2009DC.pdf> (accessed March 5, 2022).
- [21] About the business process model and notation specification version 2.0.2, (n.d.). <https://www.omg.org/spec/BPMN/> (accessed March 8, 2022).
- [22] G. Mirabelli, T. Pizzuti, F.G. González, M.Á.S. Bobi, A BPMN general framework for managing traceability in a food supply chain, in: *The 11TH International Conference On Modeling And Applied Simulation*, Vienna, 2012.
- [23] T. Pizzuti, G. Mirabelli, FTTO: An example of Food Ontology for traceability purpose, in: *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, 2013: pp. 281–286. <https://doi.org/10.1109/IDAACS.2013.6662689>.
- [24] obo-relations: RO is an ontology of relations for use with biological ontologies, Github, n.d. <https://github.com/oborel/obo-relations> (accessed March 5, 2022).
- [25] Wikidata:List of properties, (n.d.). [https://www.wikidata.org/wiki/Wikidata:List\\_of\\_properties](https://www.wikidata.org/wiki/Wikidata:List_of_properties) (accessed March 6, 2022).
- [26] R. Arp, B. Smith, A.D. Spear, *Building Ontologies with Basic Formal Ontology*, MIT Press, 2015. <https://market.android.com/details?id=book-AUxQCgAAQBAJ>.
- [27] Ontology Xref Service, Core ontology for biology and biomedicine < ontology lookup service < EMBL-EBI, (n.d.). <https://www.ebi.ac.uk/ols/ontologies/cob> (accessed March 6, 2022).

- [28] Ontology:DOLCE+DnS ultralite - odp, (n.d.). [http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS\\_Ultralite](http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite) (accessed March 6, 2022).
- [29] OBO Foundry Principles: Overview, OBO Foundry. (n.d.). <https://obofoundry.org/principles/fp-000-summary.html> (accessed March 8, 2022).
- [30] Y. He, Z. Xiang, J. Zheng, Y. Lin, J.A. Overton, E. Ong, The eXtensible ontology development (XOD) principles and tool implementation to support ontology interoperability, *J. Biomed. Semantics*. 9 (2018) 3. <https://doi.org/10.1186/s13326-017-0169-2>.
- [31] N. Matentzoglou, J.P. Balhoff, S.M. Bello, C. Bizon, M. Brush, T.J. Callahan, C.G. Chute, W.D. Duncan, C.T. Evelo, D. Gabriel, J. Graybeal, A. Gray, B.M. Gyori, M. Haendel, H. Harmse, N.L. Harris, I. Harrow, H. Hegde, A.L. Hoyt, C.T. Hoyt, D. Jiao, E. Jiménez-Ruiz, S. Jupp, H. Kim, S. Koehler, T. Liener, Q. Long, J. Malone, J.A. McLaughlin, J.A. McMurry, S. Moxon, M.C. Muñoz-Torres, D. Osumi-Sutherland, J.A. Overton, B. Peters, T. Putman, N. Queralt-Rosinach, K. Shefchek, H. Solbrig, A. Thessen, T. Tudorache, N. Vasilevsky, A.H. Wagner, C.J. Mungall, A Simple Standard for Sharing Ontological Mappings (SSSOM), *ArXiv [Cs.DB]*. (2021). <http://arxiv.org/abs/2112.07051>.
- [32] A.R. Deans, S.E. Lewis, E. Huala, S.S. Anzaldo, M. Ashburner, J.P. Balhoff, D.C. Blackburn, J.A. Blake, J.G. Burleigh, B. Chanet, L.D. Cooper, M. Courtot, S. Csösz, H. Cui, W. Dahdul, S. Das, T.A. Dececchi, A. Dettai, R. Diogo, R.E. Druzinsky, M. Dumontier, N.M. Franz, F. Friedrich, G.V. Gkoutos, M. Haendel, L.J. Harmon, T.F. Hayamizu, Y. He, H.M. Hines, N. Ibrahim, L.M. Jackson, P. Jaiswal, C. James-Zorn, S. Köhler, G. Lecointre, H. Lapp, C.J. Lawrence, N. Le Novère, J.G. Lundberg, J. Macklin, A.R. Mast, P.E. Midford, I. Mikó, C.J. Mungall, A. Oellrich, D. Osumi-Sutherland, H. Parkinson, M.J. Ramirez, S. Richter, P.N. Robinson, A. Rutenberg, K.S. Schulz, E. Segerdell, K.C. Selmann, M.J. Sharkey, A.D. Smith, B. Smith, C.D. Specht, R.B. Squires, R.W. Thacker, A. Thessen, J. Fernandez-Triana, M. Vihinen, P.D. Vize, L. Vogt, C.E. Wall, R.L. Walls, M. Westerfeld, R.A. Wharton, C.S. Wirkner, J.B. Woolley, M.J. Yoder, A.M. Zorn, P. Mabee, Finding our way through phenotypes, *PLoS Biol.* 13 (2015) e1002033. <https://doi.org/10.1371/journal.pbio.1002033>.
- [33] G.V. Gkoutos, E.C.J. Green, A.-M. Mallon, J.M. Hancock, D. Davidson, Using ontologies to describe mouse phenotypes, *Genome Biol.* 6 (2005) R8. <https://doi.org/10.1186/gb-2004-6-1-r8>.
- [34] S. Markantonatou, K. Toraki, P. Minos, A. Vacalopoulou, V. Stamou, G. Pavlidis, AMAAΘEIA: A Dish-Driven Ontology in the Food Domain, *Data*. 6 (2021) 41. <https://doi.org/10.3390/data6040041>.
- [35] C. Shimizu, L. McEwen, Q. Hirt, Towards a Pattern-Based Ontology for Chemical Laboratory Procedures, in: *WOP@ ISWC, ceur-ws.org*, 2018: pp. 40–51. [http://ceur-ws.org/Vol-2195/research\\_paper\\_1.pdf](http://ceur-ws.org/Vol-2195/research_paper_1.pdf).
- [36] Welcome to the data documentation initiative, (n.d.). <https://ddialliance.org/> (accessed March 8, 2022).
- [37] DDI CDI: Cross-domain integration, (n.d.). <https://ddialliance.org/Specification/DDI-CDI/> (accessed March 9, 2022).
- [38] Using the Process pattern — DDI 4.0 dev documentation, (n.d.). <https://ddi4.readthedocs.io/en/latest/userguides/processpattern.html> (accessed November 12, 2021).
- [39] N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, P. Antin, The iPlant Collaborative: Cyberinfrastructure for enabling data to discovery for the life sciences, *PLoS Biol.* 14 (2016) e1002342. <https://doi.org/10.1371/journal.pbio.1002342>.
- [40] E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Cech, J. Chilton, D. Clements, N. Coraor, B.A. Grünig, A. Guerler, J. Hillman-Jackson, S. Hiltmann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko, D. Blankenberg, The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update, *Nucleic Acids Res.* 46 (2018) W537–W544. <https://doi.org/10.1093/nar/gky379>.
- [41] D.D. Pennington, I.N. Athanasiadis, S. Bowers, S. Krivov, J. Madin, M. Schildhauer, F. Villa, Indirectly driven knowledge modelling in ecology, *Int. J. Metadata Semant. Ontol.* 3 (2008) 210. <https://doi.org/10.1504/ijms0.2008.023569>.
- [42] Schema.Org - schema.Org, (n.d.). <https://schema.org> (accessed March 8, 2022).
- [43] GS1 Web Vocabulary, (n.d.). <https://www.gs1.org/voc/> (accessed March 8, 2022).
- [44] SWRL: A semantic web rule language combining OWL and RuleML, (n.d.). <https://www.w3.org/Submission/SWRL> (accessed March 10, 2022).
- [45] E. Prud'hommeaux, A. Seaborne, SPARQL Query Language for RDF. (2006). <https://www.w3.org/TR/rdf-sparql-query/> (accessed October 17, 2017).
- [46] O.T. Wg, Information artifact ontology, (n.d.). <https://obofoundry.org/ontology/iao.html> (accessed March 17, 2022).
- [47] OWL web ontology language reference, (n.d.). <https://www.w3.org/TR/owl-ref/> (accessed March 8, 2022).
- [48] PROCO: PROCO: A formal ontology in the domain of process chemistry, Github, n.d. <https://github.com/proco-ontology/PROCO> (accessed March 8, 2022).
- [49] thesaurus-obo-edition: OBO Library edition of NCI, Github, n.d. <https://github.com/NCI-Thesaurus/thesaurus-obo-edition> (accessed March 8, 2022).
- [50] Time ontology in OWL, (n.d.). <https://www.w3.org/TR/owl-time/> (accessed June 9, 2021).
- [51] J.F. Allen, Towards a general theory of action and time, *Artif. Intell.* 23 (1984) 123–154. [https://doi.org/10.1016/0004-3702\(84\)90008-0](https://doi.org/10.1016/0004-3702(84)90008-0).
- [52] S. Michie, J. Thomas, M. Johnston, P.M. Aonghusa, J. Shawe-Taylor, M.P. Kelly, L.A. Deleris, A.N. Finnerty, M.M. Marques, E. Norris, A. O'Mara-Eves, R. West, The Human Behaviour-Change Project: harnessing the power of artificial intelligence and machine learning for evidence synthesis and interpretation, *Implement. Sci.* 12 (2017) 121. <https://doi.org/10.1186/s13012-017-0641-5>.
- [53] OWL web ontology language overview, (n.d.). <https://www.w3.org/TR/2004/REC-owl-features-20040210/> (accessed March 9, 2022).
- [54] PROV model primer, (n.d.). <https://www.w3.org/TR/prov-primer/> (accessed March 17, 2022).
- [55] Semantic Sensor Network ontology, (n.d.). <https://www.w3.org/TR/vocab-ssn/> (accessed March 9, 2022).
- [56] Dublin core to PROV mapping, (n.d.). <https://www.w3.org/TR/2013/NOTE-prov-dc-20130430> (accessed March 9, 2022).
- [57] A. Haller, K. Janowicz, S.J.D. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, C. Stadler, The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation, *Semantic Web*. 10 (2018) 9–32. <https://doi.org/10.3233/sw-180320>.

- [58] J. Dibie, S. Dervaux, E. Doriot, L. Ibanescu, C. Pénicaud,  $\text{MS}^2$  – A Multi-scale and Multi-step Ontology for Transformation Processes: Application to Micro-Organisms, in: Graph-Based Representation and Reasoning, Springer International Publishing, 2016: pp. 163–176. [https://doi.org/10.1007/978-3-319-40985-6\\_13](https://doi.org/10.1007/978-3-319-40985-6_13).
- [59] L. Ibanescu, J. Dibie, S. Dervaux, E. Guichard, J. Raad,  $\text{PO}^2$  – A Process and Observation Ontology in Food Science. Application to Dairy Gels, in: Metadata and Semantics Research, Springer International Publishing, 2016: pp. 155–165. [https://doi.org/10.1007/978-3-319-49157-8\\_13](https://doi.org/10.1007/978-3-319-49157-8_13).
- [60] C. Pénicaud, L. Ibanescu, T. Allard, F. Fonseca, S. Dervaux, B. Perret, H. Guillemin, S. Buchin, C. Salles, J. Dibie, E. Guichard, Relating transformation process, eco-design, composition and sensory quality in cheeses using  $\text{PO}^2$  ontology, *Int. Dairy J.* 92 (2019) 1–10. <https://doi.org/10.1016/j.idairyj.2019.01.003>.
- [61]  $\text{PO}^2$  Dairy gels, (n.d.). [http://agroportal.lirmm.fr/ontologies/PO2\\_DG](http://agroportal.lirmm.fr/ontologies/PO2_DG) (accessed March 16, 2022).
- [62] Biorefinery domain ontology, (n.d.). <http://agroportal.lirmm.fr/ontologies/PO2-BIOREFINERY> (accessed March 16, 2022).
- [63] PROV-DM: The PROV data model, (n.d.). <https://www.w3.org/TR/2013/REC-prov-dm-20130430/Overview.html> (accessed March 18, 2022).
- [64] P.L. Buttigieg, E. Pafilis, S.E. Lewis, M.P. Schildhauer, R.L. Walls, C.J. Mungall, The environment ontology in 2016: bridging domains with increased scope, semantic density, and interoperability, *J. Biomed. Semantics.* 7 (2016) 57. <https://doi.org/10.1186/s13326-016-0097-6>.