An Ontological Approach to Representing the Product Life Cycle

J. Neil Otte[a], Dimitris Kiritsi[b], Munira Mohd Ali[c], Ruoyu Yang[c], Binbin Zhang[c], Ron Rudnicki[d], Rahul Rai[c], and Barry Smith[a]*

[a] *Department of Philosophy, University at Buffalo, Buffalo, NY, USA;* [b] *École Polytechnique Fédérale de Lausanne, Switzerland;* [c] *Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY, USA;* [d] *CUBRC, Inc., Buffalo, NY, USA*

Abstract. The ability to access and share data is key to optimizing and streamlining any industrial production process. Unfortunately, the manufacturing industry is stymied by a lack of interoperability among the systems by which data are produced and managed, and this is true both within and across organizations. In this paper, we describe our work to address this problem through the creation of a suite of modular ontologies representing the product life cycle and its successive phases, from design to end of life. We call this suite the Product Life Cycle (PLC) Ontologies. The suite extends proximately from the Common Core Ontologies (CCO) used widely in defense and intelligence circles, and ultimately from the Basic Formal Ontology (BFO), which serves as top level ontology for the CCO and for some 300 further ontologies. The PLC Ontologies were developed together, but they have been factored to cover particular domains such as design, manufacturing processes, and tools. We argue that these ontologies, when used together with standard public domain alignment and browsing tools created within the context of the Semantic Web, may offer a low-cost approach to solving increasingly costly problems of data management in the manufacturing industry.

1. Introduction[1]

The ability to quickly access and share data is key to optimizing and streamlining any industrial production process. This is true across all domains of industrial production. Companies know that they must continually pursue the efficient sharing of product life cycle (PLC) data both within and across organizations (for instance, with customers and suppliers). The goal is to ensure seamless flows of data among different parties, including parties involved in different stages of the product life cycle. Having the capability to find, share, compare, and integrate data promotes collaboration; it also prevents miscommunication and other potentially costly errors. One much cited report, prepared for NIST already in 2002, estimated the cost of inadequate interoperability in the US capital facilities (including public buildings and transportation infrastructure) at $15.8 billion per year (NIST, 2002).

However, data system interoperability is elusive, for reasons that are well-known. First, there is a natural evolution of technology across both hardware and software, including machine and tool types, modes of communication, and so forth. The resulting enhancements of data platforms occur at different rates in different organizations and parts of organizations and in ways that rarely take into account the need to maintain cross-platform

---

1 integration achieved at earlier stages. Companies invest in revising their platforms in ways that rarely take into
2 account customizations made for what may be quite different purposes by partnering organizations. The frequency
3 of business mergers and acquisitions adds additional challenges to an organization's ability to effectively share data
4 across the enterprise. This frequently leaves merged companies with an archipelago of data systems that make the
5 task of integration an ongoing headache.
6   Many companies respond by developing home-grown tools to address immediate problems arising from a lack
7 of interoperability. Large companies may commission entire suites of tools to manage their data in all its aspects.
8 Even where such tools are successful, however, such successes will almost always be short-lived when new tools
9 or new data are introduced into the data pipeline by collaborators or suppliers.
10   Ontologies have been used to foster data interoperability in other domains with considerable success.[2] But work
11 with ontologies in industry has been stymied by a lack of forward, strategic, collaborative, and principled thinking
12 regarding their development and governance. This has all too frequently led to a situation where local application
13 ontologies are produced that reflect only the particular vocabularies or particular data needs of one specific group
14 within one particular organizational context. Such application ontologies reveal themselves to be incompatible with
15 other ontologies in neighboring domains, and so give rise to new data silo problems of the very sort that they were
16 intended to eliminate.
17   However, new strategies are now being implemented, above all using the widely available upper- and mid-level
18 ontologies developed in recent years by the wider community to serve as starting points for a more coherent ontology
19 development strategy for industrial purposes. But for such strategies to be effective, it is necessary that multiple
20 different organizations and groups agree to use, evaluate, and refine a common set of ontology artifacts developed
21 in tandem to support enduring interoperability. Here, we describe a suite of ontologies of this sort, which we created
22 to represent activities across the product life cycle. This suite we call the Product Life Cycle (PLC) Ontologies. It
23 uses Basic Formal Ontology as an upper-level ontology and the suite of ontologies called the Common Core
24 Ontologies (CCO) for its mid-level. We begin by providing definitions of basic terms before describing the PLC
25 domain and showing how our ontologies may be re-used to discover information of interest to industrial partners.

2. The Meaning of 'Product'

27   Our starting point is a series of general economic terms representing entities such as products, commodities,
28 services, and so forth. The meanings of such terms are often peculiar to specific contexts of use, and although
29 standards that attempt to span these contexts are available, their lack of universality makes the adoption of any
30 particular standard of questionable value. In this respect, our treatment of economic entities is no different: we, too,
31 constitute one particular context of use. What is different is that this context is one that is deliberately created to
32 allow the distinctions others wish to draw to be drawn also within our ontology but in such a way that they can be
33 clearly defined and their interrelations made explicit. If we succeed in this effort then we provide, as it were, a
34 neutral benchmark, whose existence allows others to use whatever alternative terms for our entities best suit their
35 purposes but in a way that allows interoperability to be preserved. Basic Formal Ontology[3] and the biomedical
36 reference ontologies developed on its basis (Smith, 2007) serve as a neutral benchmark of this sort in the biomedical
37 domain (Smith 2007; Arp et al. 2015).
38   If the upper-level terms in the PLC Ontologies that describe economic entities appear stipulative, narrow, or to
39 cut across important distinctions made in some other vocabulary, this is because fixing on a set of terms and
40 associated definitions that shall serve as must necessarily involve a certain degree of stipulation. Our goal, however,
41 is not to create a single terminology that can be recommended for general use, but rather to create a controlled
42 vocabulary with certain valuable features. The vocabulary will be in the public domain. All terms will be provided
43 with their own URI to enable efficient reference to definitions and examples of use. The whole will provide thereby
44 a single, stable, easily accessible, and well-architected representation of the major types of entities in the PLC
45 domains, in a way that is useful to cross-community interchange of data and information, and in such a way as to
46 ensure that there is shared meaning between the content as transmitted and the content as received.

---

[2] One notable example is SIRI, Apple's virtual assistant application.

1   To find our footing, we offer first of all a definition of the term 'product', which we take as picking out *artifacts*,
2   where the latter are defined as *objects designed by an agent to fulfill a certain function*.[4] Products are physical
3   entities that have mass, exist in space and time, and have an outer surface providing for a discernible inside and
4   outside. Cars, planes, phones, and boxes of cereal are all artifacts in the sense here intended. But a fleet of airplanes
5   and a disassembled phone consisting of a disconnected aggregate of parts are not – and thus also they are not
6   products. The total amount of sugar produced in Cuba in a given year is not a product, nor is a rack of T-shirts
7   hanging in a department store. Individual cartons of orange juice are products, as may be a box of T-shirts stacked,
8   wrapped, and standing on a pallet awaiting shipment.
9   To complete the definition, we can now specify that a product is a material artifact that is a bearer of economic
10   value.[5] This economic significance may be manifested directly in the fact that products are bought and sold, or
11   indirectly – as for instance, in production by a charitable body or a government agency – where the artifacts in
12   question are the sorts of things that could be bought and sold in certain contexts (a fact which may be reflected for
13   example in the use of shadow prices within the producing organization).[6]
14   Products are distinguished from what we shall refer to as 'commodities'. Commodities, as we define them, are
15   portions of some fungible good which are, again, bearers of economic value, including portions of both raw and
16   processed material. Commodities are commonly denoted by mass nouns ('orange juice', 'oil', 'zinc', 'wheat',
17   'silver', 'coal', and so forth). They may either be aggregates of objects, or mere (arbitrarily delineated) portions of
18   some. In many cases, commodities are used in the making of products. For instance, the orange juice produced in
19   Florida over the course of a single growing season is not a product; but portions of this orange juice are poured into
20   paper cartons for sale, and each such carton-with-contents is a product.
21   To identify products as artifacts is to distinguish the class of products from broader characterizations, such as
22   'thing that is sold or available for purchase' or 'thing produced'. This excludes from the realm of products, for
23   instance, found objects, such as seashells. A seashell may be sold in a sea-side store; but it is not a product, since it
24   was not designed to realize a function.[7] The case is different where the seashell is placed by a storekeeper in a jar
25   with sand and coquina; this is because the assembled jar-with-contents is an artifact created to have the function of
26   serving as a household decoration.
27   Also ruled out from the realm of products in the proposed sense are portions of waste material. Waste materials
28   that are removed during a process of milling or sculpting and swept aside are frequently gathered and re-sold. The
29   latter may become products, but only in result of further processes of gathering, refining, packaging, and so forth.
30   Thus, although sawdust on the floor of a mill is often called a 'byproduct' of a sawing process, such byproducts are
31   not products in our sense; rather, they may become products for example if subjected to a treatment that enables
32   portions of them to be made available as materials for sale.
33   Given that we have restricted our use of 'product' to apply only to material artifacts, we need another term for
34   items of software and other digital artifacts commonly seen as products. We call such entities 'digital goods'. This
35   choice is stipulative, but since we provide explicit criteria for the employment of the term in question we believe
36   that this stipulation is unproblematic. Digital goods, too, are created to fulfil a certain purpose.
37   With our three terms 'product', 'commodity', and 'digital good', we turn now to the entity types in Basic Formal
38   Ontology (BFO) under which these fall.[8] First, every product is an instance of BFO: *object*; every commodity is an

---

[4] For a discussion of the BFO account of function, see the work by Spear et al. (2016).

[5] To be clear, a product may have very little economic value, or may have economic value only relative to a remote market, but not in one's intended market.

[6] Parts of products may in some cases be themselves products, and in other cases, not. The head of a screwdriver is an artifact, but in most contexts, it is of no economic value independent of the screwdriver of which it is a part, and hence it fails in those contexts to be a product.

[7] A reviewer helpfully suggests drawing a distinction between products that are artifacts (what we, here, call 'products'), and products that are non-artifacts that are presented for sale, such as unprocessed fruits, vegetables, and seashells. The case of such non-artifact products may be further addressed in an update, but this is a limited set of cases that lies outside the scope of industrial manufactured products. Most commercially available apples are cleaned, coated in wax, and stickered prior to sale in grocery stores; when they are, these apples qualify as artifacts and hence may be products in our sense.

[8] A fourth example, *real estate*, is also commonly considered an economic good, and it presents an interesting wrinkle here. On the one hand, real estate may include material entities, such as landscapes and buildings upon that land—such entities may be products or commodities per our present representation. However, real estate may also refer to the space in which such

1 instance of BFO: *material entity*, and every digital good is an instance of BFO: *generically dependent continuant*.[9]
2 The terms product, commodity, and digital good, however, do not refer to subtypes of these BFO types in the way
3 that, for instance, *rabbit* is a subtype of *mammal*. This is because an entity can cease to be a product, commodity,
4 or digital good while undergoing no intrinsic change. BFO deals with entities of this sort by treating them as bearers
5 of a BFO: *role*.[10] If someone ceases to be a chief technology officer upon retirement, then she does not thereby
6 cease to be a person (and thus a BFO: *object*) as a result of this change. Rather, she loses a certain role (in this case,
7 the role of chief technology officer). Changes of this sort occur through changes in some social or organizational
8 context, such as the context of the company in which one is employed. Similarly, to be a product, commodity, or
9 digital good is both to be a particular kind of entity in the non-economic sense, but also to be viewed or treated in a
10 certain way within a particular market context in virtue of which one acquires a certain economic status.
11   How entities acquire an economic status as products, commodities, and digital goods within a given market is
12 a complex matter relating, *inter alia*, to participating in an economic exchange; being the subject of an economic
13 evaluation; being the bearer of a role that is realized in, for example, an act of economic evaluation; and bearing
14 relations to agents, such as *possessed by*, *customer of*, and *supplier of*. Most products begin their lives as bearers of
15 product roles just as soon as they begin to exist as finished artifacts, since industrial enterprises create artifacts that
16 they intend to sell.

17 ## 3. The Product Life Cycle

18
19   The product life cycle (PLC) typically begins with the identification of a need or purpose that is followed by
20 the formulation of an intention to create a corresponding product, perhaps consequent on the specification of
21 requirements. From there, the PLC passes through several phases – design, production, one or more usage phases,
22 perhaps one or more maintenance phases, and concludes with a disposal phase that may involve some sort of
23 recycling – thereby passing *from cradle to grave* (Booth et al. 2005). Our goal in what follows is to capture what is
24 true in general of instances of the type product life cycle, focusing on the manufacturing of products in the sense we
25 have defined, but proceeding in a way that is as far as possible neutral among the different perspectives, goals, and
26 intentions of those involved.
27   Product Life Cycle Management (PLM) is the field concerned with the management of product life cycle data
28 and information, in such a way as to capture the different perspectives of the parties involved at different phases of
29 the cycle. Life cycle data is generated by processes involving both artifacts and human agents, and such processes
30 may be broken out into coarser and finer grained process parts. At the coarsest level of description, the product life
31 cycle will include for example phases of design and manufacturing; usage and maintenance; and recycling and
32 disposal. Viewed at a much finer level of granularity we have product life cycle data generated by product-embedded
33 information devices (PEIDs) such as sensors and microprocessors.
34   The three main phases of the product life cycle are: BOL, MOL, and EOL (Kiritsis, Bufardi, & Xirouchakis,
35 2003). BOL includes planning, design, and manufacturing. Design includes product design, production planning
36 design, and so forth. Generally, a design action implies a recursive application of multiple sub-actions: identifying
37 requirements, defining reference concepts, doing a more and more detailed design, developing prototypes and
38 performing tests. Manufacturing includes the production of artifacts and relevant internal plant logistics. During the
39 BOL phase, the product design is generated and subsequently physically realized; the product remains within the
40 boundaries of the enterprise.

---

material entities are located. Such spaces are instances of the class Site in BFO, and thus present a fourth ontological category
undergirding economic entities of interest (Smith & Zaibert, 2001).

   [9] We are grateful for valuable comments from one of the reviewers on the BFO treatment of Generically Dependent
Continuant (GDC). As this reviewer noted, there are problems in the representation of GDCs in earlier versions of BFO, both in
its formal representation and in the natural language elucidations aimed at its clarification. These issues are currently being
analyzed by the BFO's curators.

   [10] We present a slightly different account for digital goods in what follows, using the CCO class 'stasis', which allows digital
goods to form a defined class of information content entities.

MOL includes distribution, use, and support (for example repair and maintenance). Here the product is typically in the hands of the final customer. Products are distributed, used, and supported by customers or service providers. The product history, recording distribution routes, usage conditions, failures, and maintenance provides information on the status of the product as it evolves through time. Real-time preventive services, for example car control services, can also be organized under PLM so that services may be better tailored to the performance of the product as it is being used.

EOL occurs where products are retired – actually re-collected in the company's hands (reverse logistic) – in order to be recycled, disassembled, remanufactured, reused, or disposed. EOL starts from the time when the product no longer satisfies its users, who may be the initial purchaser or a second-hand owner. Information from EOL about 'valuable parts and materials' (e.g. what materials they contain, who manufactured them) and other knowledge that facilitates material reuse should be routed to recyclers and re-users, who can obtain accurate information about product status and product content.

Closed-loop product life cycle management extends the meaning of product life cycle management in order to close the loop of the information generated in each of the successive life cycle phases. The idea is that MOL information could be used at the EOL stage in order to support decisions as to the most appropriate EOL option (for example re-manufacturing or re-use). Both MOL and EOL information could be used as feedback at the BOL stage for improving the generation of products in the next cycle. Including the activities and interests of the different actors in closed-loop product life cycle management arises from the need for measuring and controlling the total cost of the product, as well as from the growing interest not only in providing better service for users and consumers of products but also in production that is sustainable.

At present, closed-loop PLM is little more than a vision.[11] To be realized, and thereby to enable the Circular Economy[12] paradigm, a number of requirements would have to be fulfilled, including the development of an information system that would support the continuity and retrieval of life cycle data. Such a system would amount to a system of systems – one that can maintain information integration and system interoperability not just across one single life cycle of a product but across successive iterations of this cycle as model revisions are made on the basis of data collected in earlier iterations.

A major requirement for efficient PLM is the traceability of a product, which means making information about the product in all life cycle phases dynamically available in real-time. This will be a key aspect of future systems for digital manufacturing. Currently, a large amount of available information is never transformed into usable knowledge – again, because of a lack of interoperability between and thus a lack of integration of the PLM systems and models used, for example, by different enterprises involved in different stages of the cycle.

One often cited example of a significant interoperability gap in the BOL phase is the delay in the production of the Airbus A-380 (Dörfler and Baumann, 2014). Failure of interoperability can occur both within and between organizations, and can even result when the PLM systems are made by the same vendor, for example because the systems are focusing on different phases of the cycle. It can occur also as a result of a lack of attention to the discrete levels of abstraction in which data flows.[13] For instance, data at the lowest level of abstraction (for example sensor measurements of engine temperature inside a passenger vehicle) carries no useful meaning. The data needs to be processed by imposing some criterion which will transform it into information at a higher level of abstraction, for example by imposing certain thresholds for temperature to be used in defining warning triggers. By such a process, data is interpreted and rendered meaningful relative to further strategic processes. Finally, such information itself may be gathered and further processed, becoming transformed into the highest level of abstraction, into what one may call 'knowledge', whereby it is directly related to an objective or plan (for example how to deal with the recurrent high temperature of the engine). To confront these various levels of abstraction, we require the ability to track the processes involved at their respective levels, and this requires *inter alia* an information system drawing from the sort of hierarchical organization that is provided by the class hierarchy of an ontology. This brings also the advantage that ontologies are both human and machine understandable, thereby opening up an entirely new field of possibilities for automating data analysis and modeling in PLM.

---

[11] The work Witherell et al. (2013) presents an overview of the limitations of presently existing standards.

[12] What is the Circular Economy? Accessed February 01, 2019 at https://www.ellenmacarthurfoundation.org/circular-economy

[13] For a more detailed description of the data-information-knowledge continuum referenced here, see the work presented by Masud et al. (2010).

4. Ontologies, Process Data, and Closed-Loop Life Cycle Management

The PLC has evolved over time from very simple beginnings in the single-person enterprise. Today, in virtue of globalization and digitalization, a PLC may be highly complicated and involve highly knowledge-intensive processes embracing thousands of people, the latter made even more complicated by the proliferation of corporate mergers and by the increasing quantity, complexity, and variety of manufactured products. Processes were performed by a single person are fragmented into several sub-processes requiring simpler tools and lower levels of skill and knowledge (Terzi et al. 2010). This division into sub-processes applies to all processes across the PLC. For the design and manufacturing phases, the reduction to simple unit operations gave rise to the mass production paradigm that is today producing large quantities of products in automated and semi-automated plants.

These developments have consequences for the design of each new product. Today's knowledge-intensive product development environment requires the capture and representation of the product and process knowledge derived in earlier phases of product development to be reused as input to the development of future products. In the manufacturing phase, all this product information has to be shared along the production and distribution chain and represented in such a way that it can be used in future product updates. Moreover, product data needs to be put at the disposal of the service chain during the use and support phases of the PLC. During product use, input data on product behavior may need to be collected for design improvements. Recycling and retirement activities may require and provide information on components, materials, and other resources. The goal of PLM is to enable such sharing and managing of the data generated along the entire life cycle of a product.

Now, we understand an ontology to be a representation of reality comprised of taxonomy as a proper part, whose representations are intended to designate some combination of asserted classes, defined classes, and certain relations between them. The taxonomy consists of asserted classes linked by subclass or 'is a' relations forming a hierarchy of more and less general that can be represented in turn as a knowledge graph. When a class hierarchy of this sort is associated with data pertaining to instances of the relevant classes, this yields what is called a knowledge base.

Ontologies provide a shared vocabulary in environments requiring the fusion of heterogenous data. Although they are often used to reason about the entities within one single domain, by creating suites of mutually interoperable ontologies with a common top level, they can also be used as a bridge for knowledge exchange and as a tool for automatic reasoning over multiple bodies of data.[14]

See in Figure 1 a diagram of continuants participating in processes across BOL, MOL, and EOL, and the information and products that are the outputs of these processes. This representation concludes with an act of disposal whereby an artifact loses its function, leaving behind a mere material entity that may be discarded or re-used in a further production process. Three remarks are worth noting: first each node in the Figure represents an instance of a corresponding class (e.g. the depicted 'Tool' is an instance of the class 'Tool'). This means that the instance-level relations between nodes should not imply the existence of a class axiom holding that, for example, every design process has output some design specification. Secondly, there are design processes and then there are acts of design, just as there are production processes and acts of production. The Common Core Ontologies hold that the class act is a subclass of BFO: *process*, where acts are distinguished from other processes by involving an agent who plays a causative role in carrying out the process.[15] Finally, when our artifact is disposed in the EOL, the output is a material entity. This material entity was a part of the artifact, but the artifact has now ceased to exist. In other cases, however, an artifact may be disposed of and yet continue to be an instance of the class artifact, just as a hat tossed in the trash remains a hat.

---

[14] El Kadiri & Kiritsis (2015) provide a categorization of uses of ontologies.

[15] The agent in a design process, for example a human being, will engage in mental processes of various sorts. Such mental processes include, for instance, the representing to oneself the functions of an artifact that one intends to design. These matters are treated in the BFO-conformant Mental Functioning Ontology, but they are not discussed further here.

Figure 1. Information flow in PLM (cf. Kiritsis, 2011).

5. Product Data Management

Product Data Management (PDM) – also known in recent years as Technical Data Management (TDM) or Engineering Data Management (EDM) – is the planned activity of storing, archiving, and managing product engineering data (such as drawings, design specifications, instructions) and related workflows (Booth et al. 2005). PDM is a useful vehicle for structuring and maintaining the engineering data required by other enterprise functions such as manufacturing, planning, after-sales support, and so forth. It is useful also for managing data relating to product configurations and variants. It provides an efficient tool for supporting releases, version control, and the engineering change process, coming from the factory. PDM systems are comprise:

(i) an information warehouse or vault where product data is stored in a structured way;

(ii) an information management module, responsible for system administration, data accessibility, security and integrity, concurrent use of data, archiving and recovery;

(iii) a workflow management module, to be used for defining workflows and registering workflow histories;

(iv) a user interface, supporting user activities (such as queries, reporting, testing); and

(v) a series of system interfaces for programs such as Computer Aided Design (CAD), Computer Aided Engineering (CAE), and Enterprise Resource Planning (ERP).

Numerous data types, formats, and structures are managed by these tools in every stage of a product life-cycle. An overview is shown in the following Figure 2.

Figure 2. Overview of types of product data

Starting in the BOL phase, marketing data will be generated by a Project Portfolio Management (PPM) tool. In the next phase, designers will generate CAx files using CAD tools; engineers will use Electronic Design Automation (EDA) and Computer Aided Engineering (CAE). During the manufacturing phase, we can expect data to be generated by Computer Aided Manufacturing (CAM) tools in the course of Manufacturing Process Management (MPM). In the middle-of-life (MOL) phase, the sales department will use Computer Aided Process Planning (CAPP) tools and Customer Service Computer Aided Quality Control (CAQ). Throughout all of these phases, we can expect different data structures and metadata, as well as all manner of data formats. Important for our purposes is that all of these forms and formats are well known and well defined and can be represented in the form of interoperable semantic models. We believe that tracking data across such different tool-related processes, using a variety of different systems and formats, is exactly the sort of challenge that an ontology approach can help to address. The ontologies used, however, must themselves form part of a coherent suite; otherwise the same problems of cross-system incompatibility will arise once again at the ontology level.

6. Basic Formal Ontology and the Common Core Ontologies

Basic Formal Ontology is a top-level ontology, which to date, is used in over three hundred projects ranging from biomedicine to military intelligence. It is characterized both by its small size, its substantial documentation,

and its commitment to the methodology of ontological realism, whereby ontologists seek to represent entities in reality rather than concepts in the minds of people (Almeida, et. al. 2015).

The Common Core Ontologies (CCO)[16] are a BFO-compatible suite consisting of twelve ontologies that have recently become publicly available for download and re-use. The suite employs a modular design, allowing the ontologies to be used independently or in conjunction. Each specific ontology is built in the web ontology language (OWL) and designed to cover particular mid-level domains, including: agents (organizations and persons), artifacts, events, geospatial, time, qualities, and information. Figure 3 represents the import structure of the suite.

The breadth of the CCO speeds creation of ontologies within sub-domains that extend from ontologies already developed within the CCO framework. (Some dozens of CCO extension ontologies of this sort have already been created.) Approaching projects using the CCO allows a developer to continually re-use patterned expressions and greatly reduces the time and effort required by users and developers of the ontologies. When new sources of data are found, the ontologies may be extended and re-used in the process of conservative extension and enrichment. This strategy for ontology development increases a degree of semantic control, allows work to be continually re-used, and reduces the re-creation of previous work, leading to further savings in time and effort. Such a process also has downstream effects, where the mappings developed for prior projects may be re-used, along with the knowledge of developers and those working with the ontologies, who can use already tried and tested strategies for new projects.
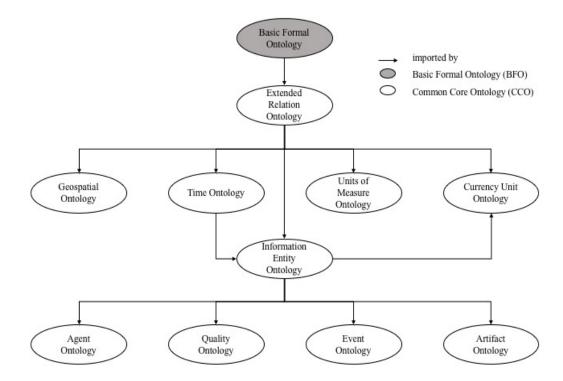

Figure 3. The Common Core Ontologies descending from Basic Formal Ontology

Like the ontologies of the Open Biomedical Ontology (OBO) Foundry, CCO uses Basic Formal Ontology as its starting point, and each component ontology is BFO compatible. The component ontologies of CCO have been developed to support a variety of defense and government-related projects, and through this continued investment and re-use, the ontologies of the CCO have been refined over time. They have also been expanded to include terms

---

[16] The Common Core Ontologies may be accessed here:
https://github.com/CommonCoreOntology/CommonCoreOntologies

of direct relevance to the manufacturing industry, and for these reasons we believe that they will be of particular interest to those developing ontologies for manufacturing and industry who are considering using BFO as top level.[17]

7. Relations in Product Life Cycle Ontologies

We describe what we can think of as the canonical life cycle process covering all things that are created, sold, and retired for profit.[18] The BOL of this canonical product life cycle includes a design process, which culminates in the production of a blueprint, model, or specification for creating a product. This specification is then refined within the particular context of a producer who generates a plan for producing the product, given the particular production context and resources available. Once this is complete, the production plan is then realized in a production process, followed by the packing, storage, shipping, and selling or leasing of the product. The MOL includes the release to a customer, whereupon the product participates in processes of use, and in many cases must be maintained, repaired, returned, or modified in order to maintain or enhance a state of readiness. At some time following use, the product enters its EOL, where it is recycled or disposed of.

Across the BOL, MOL, and EOL, many subprocesses occur. Each of these, in turn, involves various material entities that participate in the processes. These include: machines, tools, workers, portions of material (both raw and processed), and work sites, such as factory spaces, workstations, and clean rooms. To represent the relationships between these material entities and processes, we re-use a small set of two-place, instance-level relations, including:

participates in =def. a primitive relation that obtains between a process and a continuant.[19]

agent in =def. a subrelation of *participates in* that obtains when the continuant participates in the process and the continuant is causally active in the relevant process.

is input of =def a subrelation of *participates in* that obtains when the continuant participates in the process and when the presence of the continuant at the beginning of the process is a necessary condition for the start of the process.

is output of =def. a subrelation of *participates in* that obtains when the continuant participates in the process and the presence of the continuant at the end of the process is a necessary condition for the completion of the process.

In addition to material entities (including products, machinery, people), there are also information entities (documents, plans, reports, and so on) that are involved at various phases throughout the PLC, and these entities bear various relations to the other processes, whether by serving as records of the processes that have occurred, as measurements of the qualities of a resource, material, or product part at some stage, or as guiding the activities involved in the realization of some planned process. In CCO, every information content entity is an instance of the BFO class *generically dependent continuant*, and each instance stands in a primitive two-place relation of *aboutness* to some entity. CCO then distinguishes subrelations of *is about*, including *describes*, *represents*, *prescribes*, and *designates*. These relate different information content entities, such as reports, photographs, design specifications, names, and so forth, to the things they are about.

---

[17] One early example of the use of BFO in the industrial domain is provided by the OBO Ontology for Biomedical Investigations, the idea for which was first sketched by Whetzel, (2006), and more specifically in the OBI Materials Transformation branch, which deals with inputs and outputs of processes such as blending, mixing, staining, and so forth (Bandrowski et al. 2016). This part of OBI was used in the development of an ontology for functionally graded materials (Furini et al. 2016).

[18] On this use of 'canonical', see the discussion by Rosse et al. (2005).

[19] The relation of 'participates in' is taken from BFO and its subrelations from CCO.

Accounting for processes, material entities, and information entities across the successive phases of the PLC reveals a dynamic interplay of control and response that occurs across the entire cycle. Figure 4 illustrates the temporal relations between these processes, as well as the ways information garnered from the life of one single product are used in later feedback loops controlling the management of subsequent products.
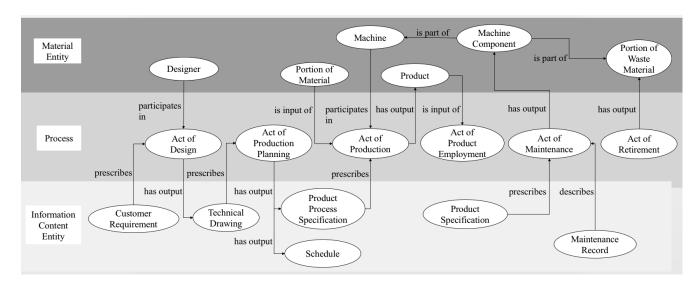

Figure 4. A sample of instances of processes, material entities, and information (and their relations) across the product life cycle.

8. Product and Product Role

We revisit now the earlier discussion of economic entities. We re-use classes from BFO and CCO as parent classes for our four major types of economic entities. From CCO we take *artifact*, which are BFO *objects* that have been designed by humans to realize some function, and *stasis*, which is a process in which a BFO *independent continuant* endures across some temporal region and bears some BFO *dependent continuant* that does not change in kind or intensity. Examples of stases include: a process during which the temperature in a room stays the same, or a process during which a light switch remains in the off position. Examples of stasis in the manufacturing domain are: the stasis of *being in the queue to be inspected for maintenance*, *of having been inspected*, *of being scheduled for maintenance*, *of having been maintained*, and so forth.

To these classes, we add *act of appraisal*, which is a process where an agent evaluates an entity, for example, to assess its monetary value.[20] With these classes in place, we then introduce a subclass of BFO Role, 'Economic Good Role', which has two subclasses of its own, distinguished by the kind of entity in which they respectively inhere:

Economic Good Role =def A role that inheres in some material entity and is realized in some act of appraisal.

Commodity Role =def An economic good role that inheres in some material entity that is not an object and is realized in some act of appraisal.

---

[20] Compare with Andersson et al. (2016).

> *Product Role* =def An economic good role that inheres in some artifact and is realized in some act of appraisal.

Note that we do not assert that a material entity becomes a product only when it is actually valued by means of some act of appraisal. Rather, we say that the product role is *realized* by such an act. But this role, like all roles, in BFO, need never be realized. For example, an object may *per accidens* have the product role – e.g. in virtue of its sitting openly on the shelf in the supermarket with a price tag affixed – without ever being appraised by any consumer.

Note further that the term 'act' here is used in a general sense of 'process performed intentionally by some agent'. Thus it does not have the connotation of a single thing that one does in one single performance; an act of maintenance, for example, may continue over several days with multiple pauses between each step in the act.

All economic good roles that inhere in material entities inhere at a time. For this to be the case there must also be some market context in which the material entity has this role. The same item, say a high pressure valve, can be a product during the time when it exists in the machine parts store, and a component of a machine at a later time when it is being used. Assigning roles to products, components, facilities, and so on, when the latter undergo transformations as they occupy, for example, different roles in the supply chain can facilitate more efficient querying for PLM purposes.

With these definitions in place, we can now create two defined classes that treat our principal economic entities:

> *Product* =def An artifact that bears a product role.

> *Commodity* =def A material entity that bears a commodity role.

Note that these definitions are not circular, since 'product role' and 'commodity role' have already been defined above without reference to, respectively, 'product' or 'commodity'. This definitional priority of role to thing bearing the role may appear puzzling, but it is a direct consequence of the BFO principle to look at what exists in reality. There are people in reality; people are extra entities, in addition to, for example, buildings and chairs. But there are no lawyers in reality; lawyers are not extra entities; they are just people who enjoy a certain role in society in virtue of which they are able to perform certain lawyerly deeds. These roles, however, *are* extra entities in reality, and they begin to exist at certain points in the lives of those persons who bear them. And similarly a certain forged copper cylinder is one entity in reality that enjoys different roles in certain phases of its existence, for example as logged item in a warehouse, as product under transshipment, as component of a tattoo machine gun, or as portion of raw material in a scrap metal yard.

'Product', 'commodity' and similar terms – in contrast to 'person' and 'role' – represent in BFO what are called 'defined classes'. These are classes created in the ontology as a matter of convenience especially for the human user; but they do not represent any extra entities in reality, whether at the level of universals or of instances. They are mere *façons de parler* about those entities which do exist, analogous, in other domains, to terms like 'pet' or 'Christmas present'.

Note that this use of roles to create defined classes in our PLC ontology cannot be applied straightforwardly to digital goods in the realm of software. This is because software programs are in BFO terms *generically dependent continuants*, and such entities are precluded from bearing roles. We can, however, provide an account of at least some of the ways in which digital entities participate in economic relations by defining what we might call a *stasis of digital ownership* as a stasis of ownership during which an information content entity is owned by some agent. *Stasis of ownership* would then be the more general class which obtains, for instance, wherever an entity of any sort is purchased or gifted.

This illustrates a more general strategy of the Common Core Ontologies which, for purposes of reasoning with OWL, cannot use the three-place relations which obtain where, for example, one entity is owned by a second entity over a certain interval of time. By employing stases, CCO can accommodate such relations, using two-place statements such as owner *participates_in* stasis1, owned thing *participates_in* stasis1, statis1 *occupies* temporal region, and so forth.

9. Services

A further family of economic entities are services, which are processes that realize a certain capability on the part of the process provider that is of some benefit to the process consumer. The relevant capability, which is a subtype of BFO:*disposition* may be formed by training a staff member to use a new machine.

Services are of relevance to the project of creating a PLC ontology in virtue of the fact that manufactured products are increasingly being purchased by consumers in ways that involve the signing of a service agreement that obligates the seller to perform certain services when requested by the consumer. Here a single purchasing event results in a customer owning both the product and a claim over the seller to provide upon demand a service relating to that product.

Guarino (2018) presents a definition of services – and in particular of public services – which gives a central place to the idea of commitment. If we generalize Guarino's proposed definition to include not merely public services but also services provided by individuals or businesses, then we arrive at a definition according to which a service is

> the sum of all activities that realize a commitment to make available to individuals, businesses, or other public authorities some capabilities intended to answer their needs, giving them some possibilities to control how and when such capabilities should be manifested.

A commitment as thus described is indeed required wherever a service contract is signed. But a commitment of this sort seems not be necessary for a service to exist. When, for example, a business is testing a novel service on its own employees, it is still thereby providing that service, yet there is no commitment of the sort mentioned in Guarino's definition.[21]

To formulate a more adequate definition, it is useful to consider a list of things characteristically referred to as services, for instance in compilations of national income statistics. These include: hairdressing, consulting, nursing, taxi transport, automobile maintenance, teaching, and symphony performance. The common characteristic in all of these examples that a service is such that its production and consumption coincide. The service is, as it were, used up in its provision, so that there is no *product* in the realm of services, since there is nothing left over that would be available to be consumed at a later time, and nothing available to be stored, or rented, or gifted. We thus propose:

provision of a service =def. process that is

(1) intentionally produced by some agent1,

(2) realizes some capability of agent1, in such a way as to

(3) address the needs or wants of some further agent2,

(4) is such that this process occurs at the request of and is to some degree under the control of agent2 as to how and when this capability is realized, and

(5) in such a way that the provision of the service and the consumption of the service coincide.

From the BFO perspective, therefore, the term 'service' refers first and foremost to instances of service-provision as thus defined. These instances form types (hairdressing, teaching, and so forth), which are also referred to as *types of services,* by which is meant *types of service provision*. For in contrast to what is the case for material goods, a service and the provision of a service are one and the same.

The 'capability' referred to in clause (2) of the definition, may include multiple sub-capabilities, for example of the different personnel involved in providing complex services. These capabilities may involve the use of machines, and may indeed be produced by the machine, which then itself serves as service provider. The (direct) recipient of the service, too, may be a machine (for instance an automobile that is being maintained), but then agent2, the relevant beneficiary, is not the machine but rather its owner. Both agent1 and agent2 may be teams or groups of

---

[21] We see a number of further problems with this definition. For example, the act of advertising a service seems to fall under the heading of 'all activities' as used in this definition – advertising is indeed a standard next step once the mentioned commitment has been arrived at. Yet advertising is not a part of a service, as the definition would require. Novel services may be offered on the market which address not *needs* on the part of purchasers of these services, but rather satisfy novel desires.

persons, or combinations of persons and machines. Whole enterprises themselves may be in the business of providing services, so that agent1 is here the entire relevant management and staff of the enterprise that is involved in any given act of service provision, the recipient of which may be some second enterprise, or a city block (where agent2 is then the set of owners of the buildings, or of the residents in those buildings), and so on.

With these classes in place, we may now turn to 'economic good', which is the most general term in our sub-ontology of economic entities. Economic goods are commonly understood to be material entities or services that satisfy human wants and provide utility.[22] 'Economic good', too, is a defined class, that is the disjunction of the classes of products, commodities, digital goods, and provisions of service.

10. Plans and Processes

The immediate subclass of BFO: *generically dependent continuant* in CCO is *information content entity*.[23] Every instance of Information Content Entity is a generically dependent continuant that is about some entity. The subclasses of Information Content Entity primarily fall into three subclasses based on the primary function of the informational entity: directing, describing, and designating. Whereas all information content entities stand in a relation of aboutness to some entity, these subrelations are about other entities in particular ways. For instance, all Directive Information Content entities prescribe some entity, where the relation of prescribing is a sub-relation of *is about*. Prescribed entities may include, for instance, desired or specified qualities, as well as specified steps to be taken in achieving a particular objective. Examples of prescribing include: the relationship between a blueprint for a house and the qualities (such as shape) later to inhere in the house, as well as the relationship between a recipe for a cake and the actions to be carried out in sequence in order to realize the objective of producing the cake.[24]

The latter example is also a plan specification. Plan specifications, whether written on paper or existing in the minds of agents, are a kind of Directive Information Content Entity, which the Common Core Ontologies define as prescribing processes through which an agent expects to achieve an objective. Such plan specifications are to be distinguished from other kinds of Directive Information Content Entity, including quality specifications, which prescribe the qualities that should inhere in some material entity. Importantly, all these specifications are also to be distinguished from *plans*, where this term is used to denote a disposition in some agent or agents to carry out a set of actions prescribed by a plan specification.

11. The Product Life Cycle (PLC) Ontologies

The PLC Ontologies include six mid-level ontologies employing a hub-and-spokes design.

- *Commercial Entity Ontology*: covering: products, commodities, and services, and so forth.
- *Design Process Ontology*: covering: act of design, act of prototyping, product model, and so forth.
- *Maintenance Ontology*: covering: act of maintenance, scheduled and unscheduled maintenance, part replacement, and so forth.
- *Testing Process Ontology*: covering: diagnostic testing, destructive testing, and so forth.

---

[22] Peter Hill's objection to defining goods as including services insists on the same ontological distinctions (what he calls 'logical categories') that we insist on here. Thus, our dispute with him appears merely terminological (Hill, T. P., 1977).

[23] CCO here draws on the BFO-conformant Information Artifact Ontology (IAO).

[24] Informally, we might characterize the 'prescribes' relation as holding between information content entities and those actions they direct, guide, or command, or those qualities which they declare should be in such-and-such a way. In practice, the 'prescribes' relation is frequently used to query for qualities or actions that presently exist or existed in the past, alongside the information content entities that prescribed them. When it comes to information content entities that are—speaking loosely—about entities that may exist in the future, there are varying approaches to representing their content.

- *Manufacturing Process Ontology*: covering acts of production, additive manufacturing, curing, polishing, and so forth.
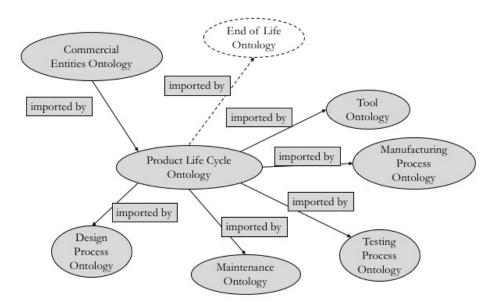- *Tool Ontology*: covering: instruments, machines, robots, computers, and so forth.

Figure 5. The Product Life Cycle (PLC) Ontologies and their dependencies.
The EOL is included for completeness but is presently in development.

The following examples illustrate the types of queries which require the sort of cross-domain integration achieved by our ontology framework in order to be answered. They are drawn from the set of competency questions developed specifically for the datasets provided by our use case.[25]

"Find all failure modes where pounds-per-square-inch was less than or equal to X for less than one hour when the ambient temperature was between Y and Z".
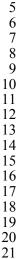
"Find all failures for parts on platform W having launch times within X hours of sunrise at location Y and that have been serviced by a provider Z within a specific date range".

Our third example query

"How do the parts supplied by suppliers affect testing outcomes of assembled products?"

is used in Figure 6 to illustrate the sorts of interrelations between different sorts of data (representing both universals and particulars) captured by our PLC Ontology framework:

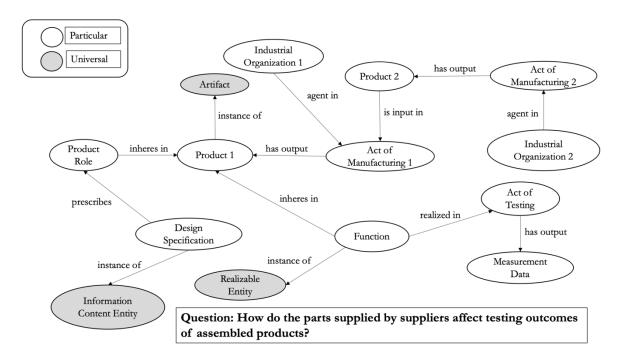---

[25] Described at https://bit.ly/2SppcQ9.

Figure 6. Ontology coverage for sample query. Note that where two or more particulars instantiate a single universal, numbers are assigned to distinguish them.

12. Future Work

We believe that incorporation of the PLC Ontology into the daily business practices of an organization can improve efficiency and productivity by creating more consistent flows of information. The ontology approach helps to solve systemic communication problems resulting from inconsistent taxonomies, nomenclatures, and processes by allowing the development of common data model and report formats using the ontology as basis. This makes the same information more readily available through different interfaces, and serves thereby to promote the degree to which the underlying systems are interoperable. We also hope that the alignment of these business processes and their data stores to a single ontology framework will assist in the process of closing the loop between the design and field-test phases and thereby enhance the ability of engineers to improve the reliability, efficiency, productivity and sustainability of the product development process.

In addition to providing representation for EOL processes, we are working with collaborators on the development and integration of a number of ontologies for industrial materials and their properties. This includes an ontology of materials properties, an ontology of functionally graded materials, an ontology of additive manufacturing, and an ontology of laminate composites. Representing the domain of materials is necessary for understanding and performing various kinds of analysis across the life cycle, though it is particularly critical to design, budgeting, and manufacturing. However, the promises and challenges raised by the materials science domain are substantial, and require significantly more discussion than is available here.

We hope, also, to contribute to the general acceptance of ontology technology within the industrial domain, and are collaborating in this regard with the Industry Ontologies Foundry (IOF) initiative (Wallace et al. 2018).

References

Almeida, et. al. *Basic formal Ontology 2.0: Specification and User's Guide.* Published June 26, 2015. Available at: https://github.com/BFO-ontology/BFO/raw/master/docs/bfo2-reference/BFO2-Reference.pdf. [Online; accessed 03-Mar-2019]

Andersson, B., Guarino, N., Johannesson, P. & Livieri, B. (2016). Towards an Ontology of Value Ascription. In; *Formal Ontology in Information Systems: Proceedings of the 9th International Conference* (FOIS 2016), IOS Press, 331–344.

Arp, R., Smith, B. & Spear, A. D. (2015). *Building Ontologies with Basic Formal Ontology*, Cambridge, MA: MIT Press.

Bandrowski, A., Brinkman, R., Brochhausen, M. et al. (2016). The Ontology for Biomedical Investigations, *PLoS ONE* 11 (4).

Booth, M. S., Stark, J. M., & Rastetter, E. (2005). Controls on nitrogen cycling in terrestrial ecosystems: a synthetic analysis of literature data. *Ecological Monographs,* 75(2), 139–157.

Dörfler, Isabel & Oliver Baumann (2014). Learning from a Drastic Failure: The Case of the Airbus A380 Program. *Industry and Innovation* 21(3), 197–214.

El Kadiri, S., & Kiritsis, D. (2015). Ontologies in the context of product lifecycle management: state of the art literature review. *International Journal of Production Research*, 53(18), 5657–5668.

Furini, F., Rai, R., Colombo, G. et al. (2016). Development of a Manufacturing Ontology for Functionally Graded Materials", *Proceedings of International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* (IDETC/CIE 2016), August 21–24, Charlotte, NC. Available at: http://philpapers.org/rec/FURDOA

Guarino, N. (2018). Services as Activities: Towards a Unified Definition for (Public) Services. *IEEE 21st International Enterprise Distributed Object Computing Conference Workshop*, 102–105.

Hastings, J., Ceusters, W., Jensen, M., Mulligan, K., & Smith B. (2012). Towards an Ontology of Mental Functioning, *Proceeedings of Workshop at the Third International Conference on Biomedical Ontology* (ICBO 2012), 1–5.

Hill, T. P. (1977). On Goods and Services. *Review of Income and Wealth*, 23: 315–338.

Kiritsis, D. (2011). Closed-loop PLM for intelligent products in the era of the Internet of things. *Computer-Aided Design*, 43(5), 479–501.

Kiritsis, D., Bufardi, A., & Xirouchakis, P. (2003). Research issues on product lifecycle management and information tracking using smart embedded systems. *Advanced Engineering Informatics*, 17(3-4), 189–202.

Masud, L. et al. (2010). From Data to Knowledge – Visualizations as Transformation Processes within the Data-Information-Knowledge Continuum. *14th International Conference on Information Visualisation*, 445–449.

Ali, M. M., Rai, R., Otte, J., & Smith, B. (2019). A Product Life Cycle Ontology for Additive Manufacturing, *Computers in Industry* 105, 191-203.

NIST (2002). *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry*, U.S. Department of Commerce, Gaithersburg, Maryland.

Rudnicki, R. (2016). *Modeling information with the Common Core Ontologies*. Available at: https://github.com/CommonCoreOntology. [Online; accessed 25-Oct-2017].

Rosse, C., Kumar, A., Mejino, J. L., Cook, D. L., Detwiler, L. T., & Smith, B. (2005). A strategy for improving and integrating biomedical ontologies. *AMIA ... Annual Symposium proceedings. AMIA Symposium, 2005*, 639–643.

Smith, B. et al. (2007). The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration. *Nature Biotechnology* 25.11: 1251.

Spear, A., Ceusters, W., & Smith, B. (2016). Functions in Basic Formal Ontology. *Applied Ontology* 11 (2):103-128.

Terzi, S., Bouras, A., Dutta, D., Garetti, M., & Kiritsis, D. (2010). Product lifecycle management–from its history to its new role. *International Journal of Product Lifecycle Management,* 4(4), 360–389.

Wallace, E., Kiritsis, D., Smith, B. & Will, C. (2018). The Industrial Ontologies Foundry proof-of-concept project. *IFIP International Conference on Advances in Production Management Systems*, 402–409.

Whetzel, P.L., Brinkman, R.R., Causton, H.C. et al. (2006). Development of FuGO: An ontology for functional genomics investigations, *Omics: A Journal of Integrative Biology* 10(2), 199–204.

Witherell, P., Kulvatunyou, B., & Rachuri, S. (2013). Towards the Synthesis of Product Knowledge Across the Life Cycle. *Proceedings of the ASME 2013 International Mechanical Engineering Congress & Exposition*. November 13-21.