# Ontology as Product-Service System
# Lessons Learned from GO, BFO and DOLCE

**Barry Smith**

*Department of Philosophy and National Center for Ontological Research, University at Buffalo, Buffalo, NY, USA*

**Abstract**

*This paper defends a view of the Gene Ontology (GO) and of Basic Formal Ontology (BFO) as examples of what the manufacturing industry calls product-service systems. This means that they are products bundled with a range of ontology services such as updates, training, help desk, and permanent identifiers. The paper argues that GO and BFO are contrasted in this respect with DOLCE, which approximates more closely to a scientific theory or a scientific publication. The paper provides a detailed overview of ontology services and concludes with a discussion of some implications of the product-service system approach for the understanding of the nature of applied ontology. Ontology developer communities are compared in this respect with developers of scientific theories and of standards (such as W3C). For each of these we can ask: what kinds of products do they develop and what kinds of services do they provide for the users of these products?*

***Keywords***

Basic Formal Ontology, Domain Ontology for Linguistic and Cognitive Engineering, product-service system

## Preamble

The success of an ontology can be measured using a range of metrics, including number and variety of associated software applications; quantities of data and literature annotated using terms from the ontology; and number, size and degree of utilization of major databases incorporating terms from the ontology. By any of these metrics, the Gene Ontology (GO) (1) is the world's most successful scientific ontology. In a tutorial presented at the Intelligent Systems for Molecular Biology (ISMB) conference in 2005, Michael Ashburner and Suzanna Lewis formulated a set of "Principles of Biomedical Ontology Construction" (2) extracted from their experience in developing the GO. These principles can, I believe, help us to account for the remarkable success of the GO, which still, after 20 years, continues on its upward trajectory as concerns numbers of users and applications, and scientific influence and utility.

The most important of the Ashburner-Lewis principles for our purposes here are:

- Before you start building an ontology learn what is out there.
- Assess extant ontologies critically and realistically. Do not reinvent. Start building – but not in isolation. Collaborate.
- The computable representation must be shared. Ontology development is inherently collaborative.
- Ensure that there is access to help. Does a warm body answer help email within a reasonable time (say 2 working days)?
- Every ontology improves when it is applied to actual instances of data.
- There will be fewer problems in the ontology and more commitment to fixing remaining problems when important research data is involved that scientists depend upon.

The basic thesis underlying these principles is that an ontology becomes more valuable *to the extent that it is aggressively used*. Ashburner and his associates accordingly devised a multi-pronged strategy designed to maximize GO usage, including:

1. developing a simple ontology editing tool (called OBO-Edit) designed to suit the needs of biologists (3)
2. making the GO easy to find by placing it in the public domain
3. making the results of using the GO in annotating biological literature and data easy to find, by creating the GO Annotation (GOA) database (4) and associated software tools
4. providing a set of evidence codes, which allow literature and data curators to record the types of evidence (for instance experimental, phylogenetic, computational) on which their annotations are based (5)
5. ensuring sustainability (6), for example (i) by ensuring that the ontology provides permanent identifiers (7), that it is updated speedily in light of advances in science and in the needs of users, and (ii) by providing easy and enduring online access to all successive versions of the GO
6. providing assistance in creation and use of persistent identifiers and repositories for ontology content (8)
7. providing online user forums and help desk, and an issue tracker that allows users to report errors or omissions in the ontology and to obtain rapid feedback
8. responding to needs of users by creating new ontologies developed in such a way as to interoperate with the GO and with each other
9. making these ontologies easy for biologists to access by creating an open portal: initially the GOBO (Global Open Biology Ontologies) portal, launched already in 2001 (9);

now the OBO (Open Biological and Biomedical Ontologies) Foundry at http://obofoundry.org.

10. ensuring an effective modular architecture for these ontologies, including providing each ontology with a name (such as "Cell Ontology", "Protein Ontology") that would make it easily findable by new users
11. ensuring an effective division of labor by devising procedures to support resolution of overlaps between ontology modules and an editorial process that allows each OBO ontology to be managed by scientists with corresponding subject-matter expertise
12. contributing to the development of ontology software (for example ROBOT (10) and the Ontozoo tools (11))
13. maintaining a reliable license regime to provide legal certainty for users and reusers of the ontologies.

This strategy has yielded (and continues to yield) a positive snowball effect whereby, when one community encounters gaps or errors when using the GO or one of its sister ontologies, these gaps and errors are rapidly fixed. This increases the value of the ontology, thereby making it attractive to further users, who in turn identify further gaps and errors thereby initiating a new cycle of improvement. Such fixes occur in a way that preserves the integrity of the GO as a resource on which existing and future users can rely as it develops over time and addresses new sets of user needs.

## Product-Service Systems (PSS)

While the role played by ontology services is familiar to all ontology practitioners, such services have not thus far been an explicit topic of theoretical reflection. In the advanced manufacturing industries, in contrast, a set of parallel developments has engendered a new organizational paradigm with considerable impact both practically and theoretically.

The term 'product-service system' was introduced in the early 2000s, when producers of, for instance, aircraft engines or photocopying machines had been experimenting with new business models focused on the marketing of bundled services.

On one model, GE does not sell engines. Rather it charges for the *use* of its engines per hour of flying time. As pointed out in (12), this fosters better alignment of incentives between GE and its customers, both of whom want to minimize the amount of downtime for unscheduled maintenance. This creates a second-order incentive on the part of GE to learn as much as possible about the reasons for engine failure and to ensure that lessons learned in servicing are not only quickly disseminated across its staff of service technicians but also communicated to the designers of the next generation of GE engineers.

GE is thereby stimulated

> to develop sophisticated algorithms for predicting likely sources of future engine failure and the optimal time to service the engine to prevent such failures. The more data and experience GE accumulates, the better these algorithms become, and the more effective GE will become in delivering these services. (12)

Manufacturing and servicing become tied together in ways that allow advances on either side to be mutually reinforcing. This then gives rise to an additional dimension that will be of significance to us here, in that, for GE, as for other advanced manufacturers, there is a cottage industry of smaller companies that offer repair services to its products. The latter may indeed bring benefits. They may be quicker, and cheaper, than GE itself. At the same time, however, such companies contribute to a fragmentation of the data landscape in ways that may bring adverse consequences for users in the future.

### Product-Service System Business Models

The business model underlying the activities of GE, as well as – somewhat later – IBM, Microsoft, Adobe, is that each of these organizations *wants* to provide services. They are, in effect, using the products that they create as delivery mechanisms for services. This model contributes to a move away from the throw-away approach characteristic of manufacturing in the second half of the 20th century, to a situation in which one product is maintained in active use for as long as possible and is continually updated to ensure a maximally productive life.

As Mont points out (13), to achieve this goal

> requires a higher level of customer involvement and education by producers. For producers and service providers, product-service systems mean a higher degree of responsibility for the product's full life cycle, the early involvement of consumers in the design of the product-service system, and design of the closed-loop system.

A product-service system is thus not simply the result of associating products with services. Rather, it constitutes an architecture where artifacts and processes are deliberately designed in such a way that they fit together in a single system, which may involve multiple enterprises and multiple user communities joined together in complex networks.

## The GO Product-Service System

In an interesting parallel to such developments in the advanced manufacturing and commercial software industries, we can now see that the GO developer community has been providing the Gene Ontology to its users as a (free) *product* bundled with a range of user *services*. This reflects the fact that the incentives of the GO developers were from the very beginning aligned with the incentives of its users. Indeed, many of these users formed part of the very same GO Consortium that was responsible for developing and maintaining the ontology itself.

From the very beginning the GO pursued a strategy of assisting its users in solving the problems that arise, for example, when a new release of the ontology involves changes that might disrupt existing workflows, or when new scientific results or new sorts of data arise which need to be accommodated within the GO and GOA frameworks. Both GO developers and GO users want to minimize the amount of downtime of the ontology of the sort that would arise, for instance, if the GO failed to correct errors or to provide terms relating to newly discovered biological phenomena in a reliable manner and with a rapid turnaround time. The GO has helped its users also by developing software, such as the AmiGO browser (14), which enhances the value of the data entered by curators into the GOA database by making these data more easily accessible.

The creation of the UniProt and other databases (15) which use GO terms in annotations of their data provides the GO itself with an important informational advantage over potential competitors. This has served in the biological domain to slow the growth of the sort of cottage industry of small ('lite'), local ontologies that has, unfortunately, been a feature of ontology work in many other domains, a phenomenon which has repeatedly given rise to the sort of fragmentation of the data landscape which ontology development was precisely designed to avoid.

## The BFO Product-Service System

BFO was created to serve as the top-level ontology of the OBO Foundry, and its three principal categories, of (i) independent and (ii) dependent continuants and (iii) occurrents, correspond to the three Gene Ontology modules for Cellular Component, Molecular Function, and Biological Process, respectively (16). The BFO developer community, too, offers not merely the BFO *product* but also a range of *services* analogous to those provided by the GO. Because BFO was established as a top-level ontology (TLO) designed to support the coordinated development of interoperable domain ontologies, it provides both (i) services to those who are using BFO as a starting point for building domain ontologies, and (ii) services to the users of these domain ontologies themselves.

The BFO developer community provides these *services* as a reflection of its conviction that an ontology benefits when it has a user community that is both large and diverse. Many of the services are provided through the OBO Foundry (of which BFO forms a part). However, there are now significant numbers of BFO-compliant ontologies outside the domain of the life sciences (17), and services must be provided to the developers and users of these ontologies also.

The BFO *product* consists of the ontology itself, in both formalized and natural language versions, which is presented to its users as a domain-neutral starting point for ontology creation in a way that brings the advantage of having been employed as TLO in many peer ontology initiatives with a correspondingly broad cohort of experts in BFO-based ontology development.

*Services* provided by the BFO developer community and its collaborators include:

- helping such users
  - to formulate definitions (18)
  - to re-engineer legacy domain ontology artifacts in such a way as to achieve BFO conformance (19, 20, 21)
- providing
  - public dissemination and developer portals (for example ontobee (22))
  - a tracker that enables users to post questions and report issues (23)
  - manuals and 'how to' documents providing guidance on developing and using BFO-conformant ontologies (24)
  - review services for developers and users of BFO-conformant ontologies
  - training videos, including site visits, tutorials, workshops and conferences
- serving as liaison between different ontology communities working with BFO, for example:
  - helping to organize collaborative ontology building efforts
  - helping to negotiate agreements concerning division of ontology coverage (and thus division of labor) in overlapping areas
  - helping to align neighboring ontologies in logically fruitful ways
- updating ontology content wherever needed, while
  - informing users in advance of proposed changes
  - providing software support for making needed updates to domain ontologies using BFO (25)
- promoting sustainability in order to provide users with the confidence that effort invested in using an ontology today will not be wasted because the ontology ceases to be maintained at some time in the future.

There are a number of items which could be added to this list as desiderata, including targeted software tools for checking the BFO compliance of a domain ontology and also tools to assist in the creation of BFO-conformant ontologies following the proposals sketched in (18), (26), and (27). The OBO Foundry provides useful first steps towards the provision of such services, but there is still no single submission point where some level of validation for at least some aspects of conformity could be achieved for BFO-based ontologies.

### BFO Version Tracking

One service that is indispensable to those who need to use an ontology over long periods of time is a traceable revision history. This enables annotations of data to be kept up to date as the ontology used in these annotations changes. For this purpose it must be possible to establish the present meanings of annotations created at an earlier date and using an earlier version of the ontology. Like the GO, BFO has a traceable history of this sort, and it has provided guidance to users of successive versions both to support consistency of use from one version to the next and also to provide the rationale for specific changes (28).

Given the large number of ontology development groups using BFO as their common top level, it is important that updates involve minimal disruption and that they are carefully managed in such a way that they do not disrupt existing workflows. Often, issues can be resolved without any necessary change in the ontology itself, for example through additional commentary on the release document, or through associated changes in an extension ontology such as the IAO (29).

The consequences of proposed changes in BFO are thoroughly evaluated before these changes are incorporated in the next public release. If such changes affect, for example, the ways English-language definitions are formulated, the BFO developers are careful to ensure that terms and relational expressions refer persistently from one version to the next and that examples of usage provided in earlier versions continue to be applicable in later versions.

### Ontologies Reusing BFO

Some 300 domain ontologies have been built using BFO as top level (30). In some cases BFO has been used as a TLO for suites of mutually interoperable domain ontologies that have been developed with the goal of providing benefits analogous to those brought in biological and biomedical domains by the OBO Foundry. Examples include the Planteome Consortium

(31), the Network of Epidemiology Related Ontologies (32) and the Penn TURBO (Transforming and Unifying Research with Biomedical Ontologies) suite (33).

Most recently, the Allotrope Foundation (34) has adopted BFO as its top-level ontology (35). The Foundation is funded by a consortium of the world's major pharmaceutical companies to improve the way its members acquire, share and gain insights from scientific data through standardization and linked data.

BFO's ability to promote interoperability across ontology frameworks has made BFO attractive not only in the life sciences but also in other areas such as industrial engineering (36), manufacturing process modeling (37), and military intelligence (38). Not least importantly for our purposes here are those users of BFO who are developing ontologies of product-service systems as set forth in (39) or (40). Examples of institutions using BFO in the engineering domain include:

- NSF Center for e-Design and the Realization of Engineered Products and Systems (21, 41)
- Engineering Informatics Research Group (42)
- Systems Engineering Research Center (SERC) (43)

In March 2019, BFO was selected, after an extensive review process managed by the National Institute of Standards and Technology (NIST), to serve as top-level ontology of the Industrial Ontologies Foundry (IOF) suite (44).

Finally, in the military and intelligence domains, BFO is being used above all through the Common Core Ontology (CCO) suite (45), which forms a set of mid-level BFO-based reference ontologies covering domains such as physical artifacts, geospatial entities, units of measure, time and events, together with a large set of domain ontologies extending from the CCO covering, inter alia, land, sea, air, planning, operations, and sensor data.

## BFO and ISO/IEC 21838

A further type of service that can be of value in almost any area of organized human activity is the establishment of standards. The International Standards Organization was founded on the idea of answering the question: what's the best way of doing this? (46). It began with units of measure and now embraces, for example, network security standards.

Standard terminologies can promote more effective communication; electrotechnical standards can promote interoperability of hardware and software. They can promote improved understandability of third-party content formulated in accordance with the standard, allow new sorts of quality measures to be developed and applied, and promote transportability of expertise.

Existing ISO standards relevant to ontology include:

- Common Logic (CL, ISO/IEC 24707:2018) (47), a family of languages extending classical First-Order Logic (FOL) with features designed to optimize computational use
- Industrial Automation Systems and Integration (ISO 15926), especially part 14: Data Model Adopted for OWL 2 Direct Semantics (draft dated 2019)
- Process Specification Language (PSL, ISO 18629) (48)
- Standard for the Exchange of Product Model Data (STEP, ISO 10303) (49)

Reflecting the number of Department of Defense (DoD) and Intelligence Community (IC) ontology applications developed on the basis of BFO as top level, the Joint Technical Committee on Information Technology (JTC 1) of ISO and the International Electrotechnical Commission (IEC) initiated in 2016 a process to consider BFO for adoption as an international standard. This proposal led to the development of ISO/IEC 21838, Part 1 of which sets forth the requirements for being a top-level ontology (TLO). Part 2 then documents that BFO satisfies these requirements, *inter alia* by providing formalizations of BFO in both OWL and Common Logic. The requirements specify further (a) that the CL formalization be proved consistent, and (b) that the OWL formalization be proved to be logically derivable therefrom.

Compliance to the ISO rules for the formulation of definitions required also a number of improvements in the treatment of natural language definitions of terms and relational expressions in BFO 2 (50). Accordingly, a new version of BFO, to be called BFO-ISO – roughly equivalent to BFO 2.1 in the conventional enumeration – will be released simultaneously with the publication of Parts 1 and 2 of ISO/IEC 21838. (The backbone is-a hierarchy of this new release is illustrated in Figure 1 below.)

The BFO-ISO framework offers a range of new opportunities for ontology developers, since CL allows greater expressivity than OWL in the formulation of axioms and of definitions of ontology terms. The FOL language from which CL is derived also brings the benefit that it is a more intuitive language when it comes to presenting formal content to human users.

### What is a Top-Level Ontology?

ISO/IEC:21838-1 defines an *ontology* as:

> a collection of terms, relational expressions and associated natural-language definitions together with one or more formal theories designed to capture the intended interpretations of these definitions.

Part of the goal of this definition is to take account of the fact that an ontology – for example the Gene Ontology, or BFO – can exist in multiple successive versions and yet remain one and the same ontology. One solution to this problem views the ontology as a document, analogous for example to a textbook, that exists in several successive editions (51). The ISO definition above relies instead on a common reading of 'collection' (as in 'museum collection') as representing something that – like organisms and organizations – can gain and lose included items (parts) over time.

Top-level ontologies deal with *categories*, which are general classes or types represented by domain-neutral terms such as 'object' or 'process'.

On this basis ISO/IEC 21838-1 defines a *top-level ontology* as

> an ontology that deals with categories shared across a maximally broad range of domains.

The standard must then provide a procedure for determining whether a candidate TLO satisfies this definition. To meet this need, Part 1 of the standard provides a list of types of entities ranging from entities relating to time, space and spacetime, to change and process; qualities and quantities; material and informational artefacts, and so forth. That an ontology O succeeds in covering a 'maximally broad range' is shown by providing documentation demonstrating that, given a type or class of entity of one or other of the listed sorts, either (i) there exists a corresponding parent (or ancestor) term for this type of entity in O itself, or (ii) a definition of this class of entity can be created through logical combination (for example disjunction) of terms from O representing types or classes satisfying (i).

ISO/IEC 21838-1 therefore does not require that the TLO contains detailed treatments of entities under all the mentioned headings. Thus, for example it is sufficient if it is possible to point to some extension ontology which serves this purpose, and which establishes the needed chain of subtype relations to some term or terms in the TLO. Thus, for example there are no terms for information artifacts in BFO – thus no terms for data, signs, symbols, software, and so on. However, many terms in this family are provided in the Common Core Ontology suite (CCO) and in the Information Artifact Ontology (IAO), which provide definitions of the mentioned terms as representing subtypes or subclasses of BFO:*generically dependent continuant*.

## BFO and DOLCE

As Guarino himself outlines in (52), BFO and DOLCE have much in common. They have a common origin: indeed a valuable early presentation of BFO was published in the very same document (53) in which the formalization of DOLCE first appears. Both BFO and DOLCE rest on the same trinity of fundamental dichotomies: between (i) *universals* and *instances*, (ii) *dependent* and *independent* entities, and (iii) *continuants* and *occurrents* (the latter referred to by DOLCE as, respectively, *endurants* and *perdurants*). On the other hand, the two ontologies differ in a number of ways from the point of view of ontology content – above all in the fact that DOLCE, but not BFO, adopts a multiplicative view of continuants. This means that DOLCE, but not BFO, distinguishes physical objects from the portions of matter which they contain at any given time.

DOLCE was launched in 2002 in WonderWeb Deliverable D18 (53), and the latter contains what is still today the definitive formalization of DOLCE. This document forms one major milestone in a stream of important contributions from the DOLCE development team, including the OntoClean methodology (54), an ontological restructuring of WordNet (55), and a series of contributions to domain ontology in areas such as law, engineering, hydrology, and – interestingly for our purposes here – of services science (56). It is worth remarking here also that the DOLCE community contributed to a remarkable degree in the provision of ontology-related services, above all in establishing the FOIS ontology series, the *Applied Ontology* journal, and the International Association for Ontology and Its Applications. While these are services to the broader ontology community, rather than services to the users of specific ontologies of the sort that concern us here, this does not take anything away from their intrinsic value and importance.

### BFO and Ontological Realism

BFO and DOLCE differ also with respect to the issue of ontological realism (57). For the BFO developer community, ontology is an effort to foster consistency in the ways data are described by following a specific methodology which uses reality as benchmark. The goal is to counteract the many tendencies leading to ad hoc and non-interoperable coding of data, and thereby to the formation of data silos, of the sort that have plagued ontology efforts in the past. In many cases, most notably in the case of the Gene Ontology, we can use the results of empirical science as a means of gaining access to those portions of reality that form the needed benchmark. In other cases we may use for this purpose authoritative sources such as industrial or military standards, or codes of law.

The reason for using such benchmarks as the basis for creating ontologies is (simplifying considerably) to arrive at a situation in which there will be just one authoritative ontology for each domain of reality. This goal can be achieved only if we can persuade ontology developers to accept certain shared constraints on how they build ontologies and for this we need to employ a strategy that does not endanger, for example, the flexibility that is needed to keep pace with scientific advance. The OBO Foundry has been at least partially successful in meeting these conditions, and in ways that essentially involve the use of BFO.

### BFO on the Nature of Ontology

In 2008, I defended the proposition that ontologies like the GO, which are created to support the retrieval, integration and analysis of scientific data, are a *part of science* (58). They form what we can think of (again simplifying somewhat) as the terminological scaffolding of a scientific theory. In the case of GO the relevant theory would be molecular biology. The ontologies in question are therefore subject to the same empirically-based methods of evaluation as are those theories themselves. An ontology like the GO must, of course, be associated with implementations satisfying the requirements of software engineering. But the ontology is not the sort of thing that can exist only as embedded in some specific software framework.

A counterpart view, as applied to a top-level ontology such as BFO, would see the latter as providing terminological resources at a more general level, which is to say, above the level of the specific sciences. This is to enable domain ontologies at lower levels to be linked together. What subject-matter experts in, for example, molecular biology or clinical medicine will see are terms in GO, or in some clinical ontology such as SNOMED CT. BFO will remain invisible.

Certain requirements must be satisfied, however, If an ontology is to serve as "terminological scaffolding" of a scientific theory or of some counterpart thereof for example in the field of manufacturing industry (44) or military doctrine (59). For again, this will require services to be provided from the side of the developers of the ontology.

Above all, it will require provision of services of the sort that will give users confidence that the ontology will be reliably disseminated and maintained, and will continue to offer needed serviced, in the future. Only thus can we overcome a range of familiar obstacles standing in the way of adoption of an ontology by new users, captured in complaints for example to the effect that this or that ontology "was not being kept up to date"; that it "was not clear to me how the ontology would fit my particular data"; that it "would not be able to incorporate the terms I needed in time for my funding deadline"; or that "I did not have the confidence that the ontology will still be supported when I need it in the future". Too often, potential new users of an existing ontology are motivated to build ontologies of their own, resulting almost always in ad hoc contrivances with a very short half-life.

**DOLCE and Ontological Realism**

The views of the developers of DOLCE on the topic of ontological realism are formulated as follows:

> the aim of DOLCE is to capture the intuitive and cognitive bias underlying common-sense … DOLCE does not commit to a strong referentialist metaphysics (it does not make claims on the intrinsic nature of the world) and does not take a scientific perspective (it is not an ontology of, say, physics or of social sciences). Rather, it looks at reality from the mesoscopic and conceptual level aiming at a formal description of a particular, yet fairly natural, conceptualization of the world. (60)

DOLCE, that is to say, aims to capture the ontological categories lying behind natural language and human common sense, so that its categories are to be regarded "conceptual containers" and thus as "cognitive artifacts ultimately depending on human perception, cultural imprints and social conventions." (53, 55)

One problem with views of this sort, however, is that they can be detrimental to the goal of using a top-level ontology as a means of promoting interoperability of domain ontologies defined in its terms. Indeed, since 'cultural imprints and social conventions' vary so widely, allowing these to play a role in determining top-level ontology content raises the problem of non-interoperability at this very top level itself.

**DOLCE on the Nature of Ontology**

A more charitable, and I believe more adequate, view of DOLCE, however, sees it as a scientific ontology of human common sense, with needed benchmarks provided, for example, by linguistics, perceptual psychology and action theory. DOLCE can in this way be viewed as a contribution to science, that has served as inspiration for the development of many new ontologies, both domain ontologies based on DOLCE in its FOL version, as well as spin-offs from DOLCE, such as the Unified Foundational Ontology (61).

In support of a view of this sort is the remarkable degree to which DOLCE (FOL) has remained stable across its entire history, reflecting the degree to which human common sense, too, has also – for reasons relating to the evolutionary survival of the species – manifested little change over long periods. (62)

The remarkable stability of DOLCE in its FOL version has however been to some degree overshadowed by the many – in some ontology user circles more conspicuous – artifacts created using DOLCE (FOL) as inspiration, but formalized using OWL, for instance as listed in (63). The proliferation of versions of an ontology is clearly not an unalloyed good in the ontology context, since it will tend to diminish the degree to which the ontology will be trusted by potential users as a resource that can be relied upon to promote interoperability in a sustainable fashion. Guarino has accordingly (in personal communication) referred to the DOLCE OWL artifacts as mere "variants" over and against the one "version" of DOLCE described in Sections 3 and 4 of the WonderWeb deliverable D18 (2003).

**Lessons Learned from BFO and DOLCE**

In the matter of update history, now, BFO lies somewhere intermediate between DOLCE (FOL) on the one hand and DOLCE (OWL) on the other. For BFO has been maintained as one thing *through a series of updates* over time motivated by the experiences of its users. The one "version" of DOLCE, in contrast, has been without update for some 17 years, which means that it has been unaffected by the changes in the field of ontology around it. Thus while DOLCE provided the ontological foundations for important work in a series of multi-partner projects (64) involving the creation of DOLCE-based domain ontologies, none of these collaborations brought about changes in DOLCE itself as a result of discoveries made in its actual use. DOLCE (FOL) has thus not benefited from the sort of virtuous cycle of continuous development through a mutually beneficial interaction with the users of domain ontologies constructed in its terms that has characterized the evolution of BFO.

A possible exception in this regard is the e-Science Knowledge Infrastructure (65), which has taken important steps towards creation of a suite of ontologies in the domain of hydrology. (66, 67). Otherwise, while DOLCE has certainly inspired the creation of appreciable bodies of domain-ontology content, it has not been able to serve in any of the domains where it has been applied as an easily findable, easily learnable, easily teachable hub for the development of mutually consistent extension ontologies in sustainable unitary suites analogous to the OBO Foundry or the Common Core (45).

The domain ontology contributions created on the basis of DOLCE thus survive largely as fragments, documented in scientific papers, rather than as ontological going concerns.

In sum, the various domain ontologies built around the official DOLCE have not been provided with the *services* needed to make the domain ontologies defined in its terms work well together in a sustainable, publicly accessible way. Matters are somewhat different in the case of DOLCE in its OWL formalizations, which have a richer history of usage than the FOL version of DOLCE. This is in part because most users of ontologies work with OWL rather than FOL. But it is in part also because of differences in management policy, reflected in the multiple sorts of user assistance documented for example at (63).

The FOL version of DOLCE was, clearly an unusually impressive piece of ontology design from the very start, and thus there are good reasons why it has survived so long without updates. Its record in this respect is not perfect, however, given that one of the most important spin-offs from DOLCE was the DOLCE-CORE proposal presented in 2009 (68). For the latter describes a number of improvements over the official DOLCE, and represents what it calls a "first step, after the release of the DOLCE ontology in 2002, toward a new version of this ontological system." DOLCE-Core has however not resulted in a new version of DOLCE even though members of the Guarino lab have been using it to build domain ontologies since 2009, as for example described in (69) and (70).

## Conclusion

**1. Ontology as Interdiscipline** There are scientific *disciplines*; and, it is sometimes said, there are scientific *interdisciplines* (71). Interdisciplines work like disciplines. They involve people making scientific contributions – theories, experiments, data, perhaps also ontologies – but in such a way that these contributions cross established disciplinary boundaries.

There is no question that the discipline of ontology as a whole is properly conceived as an interdiscipline, spanning (at least) philosophy, linguistics, engineering, and various branches of computer and information science. We believe, however, that what has been said above suggests a new approach to the question of the interdisciplinary nature not only of ontology, but also of a range of related activities such as scientific theorizing and standards development.

Let us therefore assume that ontology itself is an interdiscipline. What is to be said now of each single ontology? Is there some single type or class whose instances are the GO, and BFO, and IAO, and UFO, and DOLCE, and DOLCE+DnS Ultralite v. 3.31? Or do we rather need something like an *ontology of ontologies* that would have no single root?

**2. Ontology as business enterprise:** Much of the foregoing has rested on what I think is a novel view of ontology developer teams as analogous to business enterprises.

The latter, as we have seen, have as their outputs both products and services, more or less closely bundled together. Products are of two sorts: material products such as laptops and servers; and digital products such as Adobe Acrobat. Some enterprises – such as business consultants – produce no products at all but only services. Some enterprises product products – such as IBM's Watson – but they give them away free, and provide services to support their use.

I conclude merely by noting that this perspective can be applied not only to teams of ontology developers. Standards organizations, too, can be viewed under this heading, given that NIST, CEN, ASME, ISO, IEC, W3C and even HL7 are both production systems, producing *standards,* and *service systems*, providing support for the users of these standards.

And communities of scientists, too, can be viewed as providing product-service systems, containing both production elements – producing scientific results, publications – and service elements, for example training each new cohort of scientists.

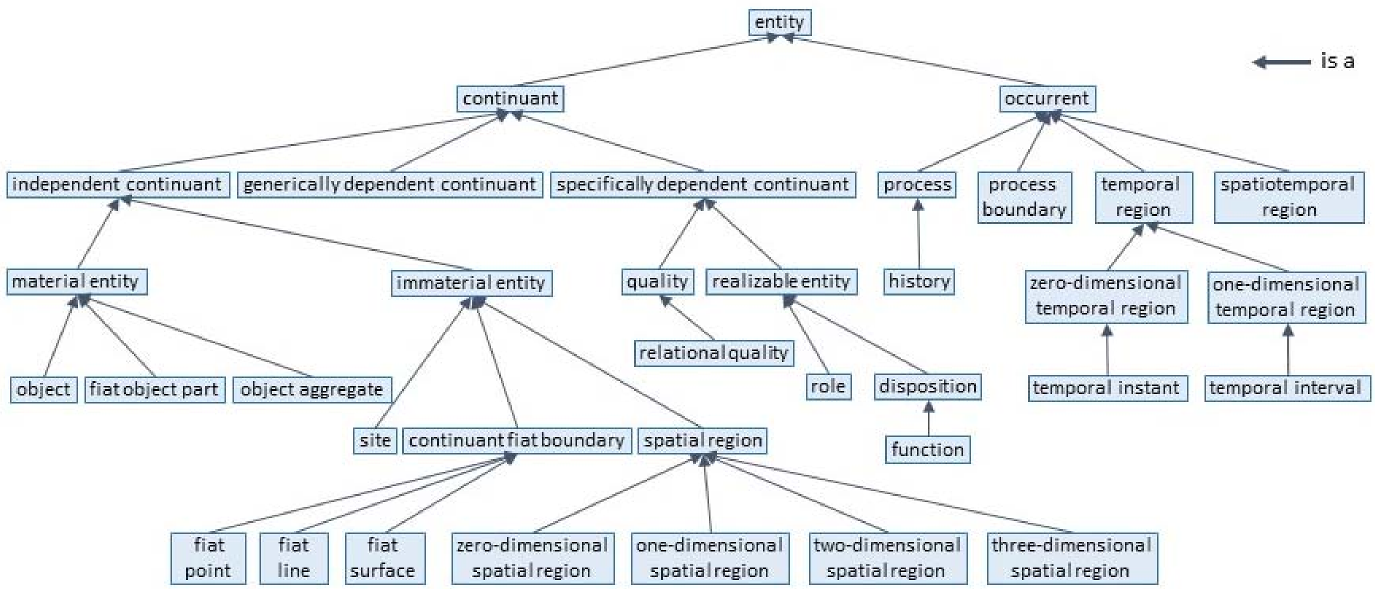


*Figure 1: Is_a Hierarchy of BFO-ISO*

## References


1. Ashburner M, Ball CA, Blake JA, et al. Gene Ontology: tool for the unification of biology. Nature Genetics. 25(1), 2000. p. 25-29.
2. https://buffalo.app.box.com/v/Building-Ontologies. All URIs last accessed April 8, 2019.
3. Day-Richter J, Harris MA, Haendel M. et al. OBO-Edit – An ontology editor for biologists. Bioinformatics, 23:16; 2198–2200, 2007.
4. Huntley RP, Sawford T, Mutowo-Meullenet P, Shypitsyna A, Bonilla C, Martin MJ, O'Donovan C. The GOA database: Gene Ontology annotation updates for 2015. Nucleic Acids Research, 43, D1, 2015, p. D1057–D1063.
5. http://wiki.geneontology.org/index.php/Guide_to_ GO_ Evidence_Codes
6. "Services to support the OBO Foundry standards", https://app.dimensions.ai/details/grant/grant.7071924
7. http://www.obofoundry.org/docs/Policy_for_OBO _name space_and_associated_PURL_requests
8. http://www.obofoundry.org/id-policy.html
9. Hill DP, Blake JA, Richardson JE, Ringwald M. Extension and integration of the Gene Ontology (GO): Combining GO Vocabularies with external vocabularies. Genome Res. 2002; 12(12), p. 1982-1991.
10. http://robot.obolibrary.org
11. http://www.hegroup.org/ontozoo
12. Chesbrough H. Open services innovation, https: //www.innovationexcellence.com/blog/2012/11/04/open-services-innovation/.
13. Mont O. Clarifying the concept of product-service system. Journal of Cleaner Production, 10 (2002), p. 237-245.
14. http://amigo.geneontology.org/amigo

15. The UniProt Consortium. The Universal Protein Resource (UniProt). Nucleic Acids Res. 2008 Jan; 36(Database issue).
16. Munn, K. & Smith, B. (eds.) Applied Ontology: An Introduction, Frankfurt/Lancaster: ontos/Walter de Gruyter, 2008.
17. http://basic-formal-ontology.org/users.html
18. Seppälä S, Ruttenberg A, Smith B. Guidelines for writing definitions in ontologies. Ciência da informação 46 (1), 73-88; 2017.
19. Hagedorn TJ, Krishnamurty S, Grosse IR, A knowledge-based method for innovative design for additive manufacturing supported by modular ontologies, Journal of Computing and Information Science in Engineering, 18(2), 021009, 2018.
20. Arena D, Ameri F, Kiritsis D, Skill modeling for digital factories, IFIP International Conference on Advances in Production Management Systems. APMS 2018: Advances in Production Management Systems. Smart Manufacturing for Industry 4.0, 2018, 318-326.
21. Hagedorn TJ, Smith B, Krishnamurty S, Grosse I. Interoperability of disparate engineering domain ontologies using Basic Formal Ontology, Journal of Engineering Design, in production, 2019.
22. http://www.ontobee.org/
23. https://groups.google.com/d/forum/bfo-discuss
24. Arp R, Smith B, Spear A. Building ontologies with Basic Formal Ontology, MIT Press**;** 2015**.**
25. http://ontobull.hegroup.org/
26. Seyed P. A method for evaluating ontologies – Introducing the BFO-Rigidity Decision Tree Wizard, Formal Ontology in Information Systems (FOIS 2012), p. 191-204.
27. Seppälä S. An ontological framework for modeling the contents of definitions. Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication, 2015, 21(1),
23-50
28. Seppälä S, Smith B, Ceusters W. Applying the realism-based ontology-versioning method for tracking changes in the Basic Formal Ontology, Formal Ontology in Information Systems (FOIS 2014), 2014, p. 227-240.
29. http://www.ontobee.org/ontology/IAO
30. http://basic-formal-ontology.org/users.html
31. http://planteome.org/
32. Pesquita C, Ferreira JD, Couto FM, Silva MJ. The epidemiology ontology: an ontology for the semantic annotation of epidemiological resources. J Biomed Semantics. 2014; 5: 4.
33. http://upibi.org/turbo/
34. https://www.allotrope.org/
35. http://www.prweb.com/releases/2018/03/prweb15337219.htm
36. http://ncorwiki.buffalo.edu/index.php/BFO-Engineering
37. Arena DN. Towards semantics-driven modelling and simulation of context-aware manufacturing systems, Dissertation, EPFL – École polytechnique fédérale de Lausanne, School of Engineering, June 2019.
38. Smith B, Malyuta T, Rudnicki R, et al. IAO-Intel: An ontology of information artifacts in the intelligence domain. Semantic Technologies for Intelligence, Defense, and Security, (STIDS 2013), CEUR, vol. 1097, p. 33-40.
39. Correia A, Stokic D, Siafaka R, Scholze S, Ontology for collaborative development of product service systems based on Basic Formal Ontology, 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC).
40. DIVERSITY/FALCON White Paper: Product Service System (PSS) Ontology for Discrete Manufacturing. Draft 2019. Available from ICT for Sustainable Manufacturing, EPFL CH-1015 Lausanne.
41. http://www.centerforedesign.org/
42. http://infoneer.wp.txstate.edu/
43. Transforming Systems Engineering through Model-Centric Engineering A013 Interim Technical Report SERC-2017-TR-111. https://apps.dtic.mil/dtic/tr/fulltext/u2/1058354.pdf
44. Wallace E, Kiritsis D, Smith B, Will C. The Industrial Ontologies Foundry proof-of-concept project, APMS 2018: Advances in production management systems. Smart manufacturing for industry 4.0 (IFIP International conference on advances in production management systems, 536). Springer, 2018; p. 402-409.
45. https://www.cubrc.org/index.php/data-science-and-information-fusion/ontology
46. https://www.iso.org/benefits-of-standards.html
47. Gruninger M, Katsumi M, Mossakowski T. Revision of ISO 24707 (Common Logic), 2013. http://metadata-standards.org.
48 Grüninger, M. Ontology of the Process Specifica-tion Language. http://bbs.w3china.org/dragonstar/-papers/psl-chapter.pdf
49. Barbau R, Krima S, Rachuri S, et al. OntoSTEP: Enriching product model data using ontologies. Computer-Aided Design. 44(6), 2012, 575-590.
50. https://github.com/BFO-ontology/BFO/wiki
51. Neuhaus F. What is an Ontology? 2018. arxiv 1810.09171.
52. Guarino N. BFO and DOLCE: So far, so close …, Cosmos + Taxis 4 (4):10-18. 2017.
53. Masolo C, Borgo S, Gangemi A, et al. Deliverable D18: Ontology Library, IST Project 2001-33052 WonderWeb: Ontology infrastructure for the Semantic Web. 2003.
54. Guarino N, Welty C. An Overview of OntoClean. S Staab and R Studer (eds.) Handbook on Ontologies, 2009, 201-220,
55. Oltramari A, Gangemi A, Guarino N, Masolo C. Restructuring WordNet's Top-Level: The OntoClean approach, Proceedings of Workshop on Ontologies and Lexical Knowledge Bases (OntoLex'02), 2002, p. 17-26.
56. Ferrario R, Guarino N, Janiesch C, Kiemes T, Oberle D, Probst F. Towards an ontological foundation of services science: The General Service Model. Internationale Tagung Wirtschaftsinformatik, Zürich, 2011.
57. Smith B, Ceusters W. Ontological realism: A methodology for coordinated evolution of scientific ontologies. Appl Ontol. 2010, 5(3-4), p. 139-188.
58. Smith B. Ontology (science). Formal Ontology in Information Systems (FOIS 2008). Amsterdam: IOS Press. p. 21-35.
59. Morosoff P, Rudnicki R, Bryant J, Farrell R, Smith B. Joint Doctrine Ontology: A benchmark for military information systems interoperability. Semantic Technology for Intelligence, Defense and Security (STIDS 2015). (CEUR 1325), p. 2-9.
60. Borgo S, Masolo C. Foundational choices in DOLCE, in R Poli et al. (eds.), Theory and applications of ontology: Computer applications, Springer, 2010, p. 279-295.
61. https://ontology.com.br/

62. Smith B. Formal ontology, common sense, and cognitive science. International Journal of Human-Computer Studies, 43, 1995. p. 641-667.
63. http://www.ontologydesignpatterns.org/ont/dul/
64. http://www.loa.istc.cnr.it/old/Projects.html
65. Brodaric B, Reitsma F, Qiang Y. SKIing with DOLCE: toward an e-Science Knowledge Infrastructure. Formal Ontology in Information Systems, Amsterdam: IOS Press, 2008. p. 208-219.
66. Brodaric B, Hahmann T, Gruninger M. Water features and their parts. Applied Ontology, 2018, 14(1) 1-42.
67. https://www.researchgate.net/project/A-hydro-foundational-ontology-HyFO
68. Borgo S, Masolo C. Foundational choices in DOLCE, in: Handbook on ontologies, Springer; 2009. p. 361-381.
69. Masolo C. Founding properties on measurement. Formal ontology in information systems (FOIS 2010); p. 89-102.
70. Ferrario R, Porello D. Towards a conceptualization of sociomaterial entanglement, CONTEXT: Modeling and using context, 2015. p. 32-46.
71. Newell A. Reflections on the structure of an interdiscipline, in F Machlup and U Mansfield (Eds.), The study of information: Interdisciplinary messages. New York, 1983, p. 99-109.