A First-Order Logic Formalization of the Industrial Ontologies Foundry Signature Using Basic Formal Ontology

Barry Smith[a], Farhad Ameri[b], Hyunmin Cheong[c], Dimitris Kiritsis[d], Dusan Sormaz[e], Chris Will[f], and J. Neil Otte[g,1]

[a] *National Center for Ontological Research,* University *at Buffalo*
[b] *Engineering Informatics Lab, Texas State University*
[c] *Autodesk Research*
[d] *EPFL, ICT for Sustainable Manufacturing*
[e] *Ohio University*, *Department of Industrial and Systems Engineering*
[f] *Dassault Systèmes*
[g] *Johns Hopkins University Applied Physics Laboratory*

Abstract. Basic Formal Ontology (BFO) is a top-level ontology used in hundreds of active projects in scientific and other domains. BFO has been selected to serve as top-level ontology in the Industrial Ontologies Foundry (IOF), an initiative to create a suite of ontologies to support digital manufacturing on the part of representatives from a number of branches of the advanced manufacturing industries. We here present a first draft set of axioms and definitions of an IOF upper ontology descending from BFO. The axiomatization is designed to capture the meanings of terms commonly used in manufacturing and is designed to serve as starting point for the construction of the IOF ontology suite.

Keywords. Basic Formal Ontology (BFO), Industrial Ontologies Foundry (IOF), advanced manufacturing industry, top-level ontology, first-order logic.

1. Introduction

Basic Formal Ontology (BFO) is a small, top-level ontology that is used in a wide range of projects focusing especially on information-driven sciences. BFO provides ontology developers with a common, tested starting point for the formulation of definitions in a way that is designed to promote interoperability. BFO has been used in this manner in the domain of biomedical ontology for some fifteen years, where it serves as the top-level ontology of the Open Biomedical Ontology (OBO) Foundry.

1 The authors wish to thank Evan Wallace and Boonserm Kulvatunyou for their comments.

The OBO Foundry is an initiative established in the wake of the Human Genome Project to coordinate the development and use of high-quality ontologies in the domain of biology and biomedicine in a way that would promote interoperability across species and disciplines. To be admitted to the OBO Foundry, biomedical domain ontologies are required to conform to a series of principles relating to accessibility, quality, scientific accuracy, consistent development, sustainability, and service to users.

In more recent years, BFO has been applied in other areas, including intelligence, defense, and security. As a consequence of these developments, BFO is in the final stages of review to become international standard ISO/IEC: 21838-2.[2]

In addition, BFO is being applied in a series of industrial engineering- related projects, including those documented in [4]–[18]. Following a lengthy evaluation and selection process, BFO was adopted in the spring of 2019 to be the top-level ontology of the Industrial Ontologies Foundry (IOF), an ecosystem of ontology resources designed to promote interoperability in digital manufacturing and related fields.

Like the OBO Foundry, the IOF promotes a principles-based approach to the design of ontologies. The initial focus is on support for the manufacturing domain, and more specifically on ontologies for design, maintenance, supply chain, production, and lifecycle management, or in other words for the successive stages in the canonical manufacturing product lifecycle. It is anticipated that in later stages the service, construction, and extraction industries will be included also within the IOF framework.

It has been clear for some time that the task of developing a coherent set of ontologies covering the manufacturing domain will present a considerable challenge. Manufacturing is not only highly multidisciplinary; it is affected also by the need to address the needs of manufacturing enterprises, who gain commercial benefits when their data is held in proprietary siloes. On the other hand, the increasing importance of outsourcing and of the resultant long supply chains provide incentives for the sharing of data and for the coordinated development of interoperable software, and it is to address these needs that the IOF was formed.

Table 1 is a list of terms provided to us by members of the IOF as part of the proof-of-concept project described in [17]. The latter summarizes the goals and organizational set up of the IOF and presents the results of an initial scope-determining experiment, in which some twenty (in the end: twenty-five) representative terms compiled by subject-matter experts within the IOF.

2 https://www.iso.org/standard/74572.html.

Table 1. IOF Top Twenty(-Five) Term List

[1]	Assembly	[13]	Manufacturing resource
[2]	Assembly process	[14]	Manufacturing tool
[3]	Business process	[15]	Material resource
[3]	Component	[16]	Operation specification
[4]	Customer	[17]	Piece of equipment
[5]	Design	[18]	Piece of manufacturing equipment
[6]	Feature description	[19]	Plan
[7]	Input material role	[20]	Product quality
[8]	Manufacturing machine	[21]	Product
[9]	Manufacturing process	[22]	Quality specification
[10]	Manufacturing process plan	[23]	Task
[11]	Raw material role	[24]	Transport process
[12]	Supplier	[25]	Step

The goal of the present paper is to provide a summary account and illustrative portions of an early draft formal representation of the definitions of these terms and of associated axioms within the BFO framework. The material presented here is extracted from a more extensive formalization,[3] which is still subject to update. It should not be assumed that all of the terms introduced here will be included in the ultimate IOF signature.

2. Syntax

The formalization in what follows employs standard first-order logic (FOL) notation for negation, conjunction, disjunction, material implication, biconditional implication, universal quantification, and existential quantification, using, respectively: $\neg$, $\wedge$, $\vee$, $\rightarrow$ $\equiv$, $\forall$, $\exists$. Variables t, t', … , range over temporal regions.

For the sake of readability, initial universal quantifiers are suppressed. Hence, instead of writing:

$$\forall(x)\ [\text{business-process}(x) \rightarrow \text{planned-process}(x)]$$

we write:

3 Available at https://buffalo.box.com/v/IOF-Signature.

business-process(x) → planned-process(x)

In addition, we sometimes abbreviate

instance-of(x, y, t)

by writing simply: 'y(x)'. For an account of this treatment of predication, see [3], chapter 5.

Terms from Table 1 in what follows appear in **bold**. Other terms necessary for the definition of these terms appear in roman face. Some terms are marked as primitive. This signifies that they are too basic in our vocabulary to receive definitions because there are no more basic terms which could be used to define them. Two sorts of primitive terms are distinguished. First, are primitives, such as 'disposition', 'role', 'process', and 'material entity'— treated in the BFO 2.0 Specification and User Guide (Almeida, et al.). Other terms, such as 'artifact' and 'information content entity', are treated in the Information Artifact Ontology (IAO)[4] and in the Common Core Ontologies (CCO), a suite of mid-level ontologies conformant to BFO 2.0.[5] Terms derived from these sources and used in definitions will be indicated as such on first occurrence by use of the corresponding namespace ID.

3. Universals and Defined Classes

Some terms in what follows refer not to universals but to what BFO calls 'defined classes'. Consider, for example, the term 'lawyer'. This does not represent an extra entity instantiating a universal in its own right. Rather, it connotes that some already classified particular entity (some instance, in this case, of the universal *homo sapiens*) has a certain *lawyer role.* The latter *is* an extra entity, and thus BFO admits a corresponding *role* universal. The term 'lawyer' then represents the defined class consisting, at any given time, of all those entities (human beings) that have the *lawyer role*. Defined classes may also be defined disjunctively – for example, in the definition of 'agent' as 'person or organization'.

4. BFO-IOF-FOL

In this section we document a representative fraction of the *is-a* (aka *subclass of*) relations in our first-order logic axiomatization. A somewhat compressed version of this *is-a* hierarchy, which leaves out IAO and CCO terms for the sake of readability, is presented in Figure 1. We provide also selected *is-a* axioms from BFO-IOF-FOL, and associated comments, including:

1. **planned process** subclass of process.

planned-process(x) → BFO:process(x)

2. **manufacturing tool** subclass of object.

manufacturing-tool(x) → BFO:object(x)

4 http://www.obofoundry.org/ontology/iao.html

5 The Common Core Ontologies (CCO) are available at: https://github.com/CommonCoreOntology.

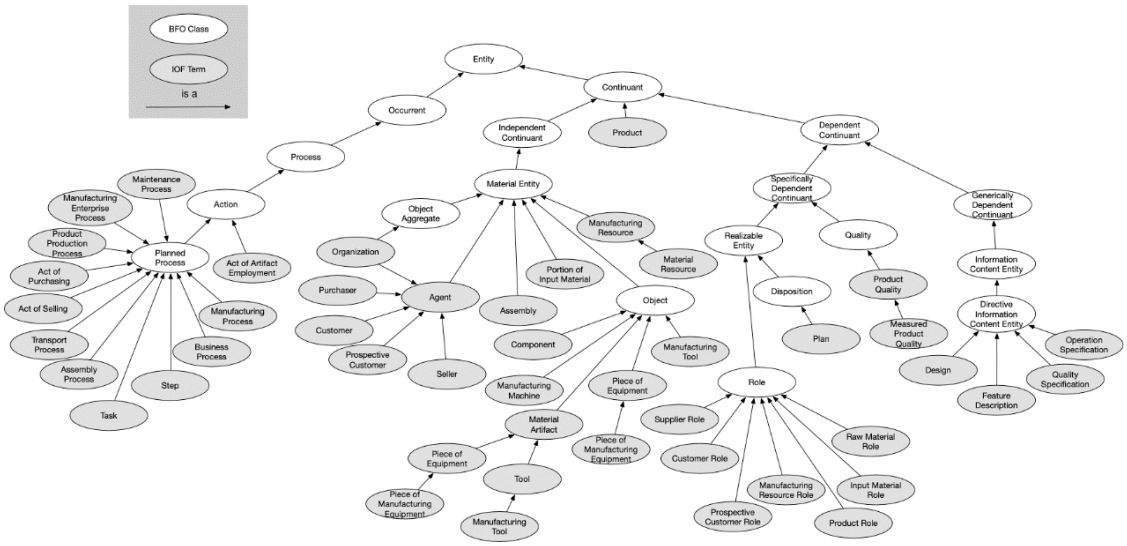

Figure 1. Fragment of the BFO class hierarchy aligned with classes bearing labels from the top twenty-five terms provided by the IOF. An expanded version incorporating terms from IAO and CCO is provided at https://buffalo.box.com/v/IOF-Signature

Comment: The BFO term 'object' comprehends material entities possessing one or other type of causal unity. In addition to material artifacts such as laptops, objects include: solid portions of matter. [21]

3. **product** subclass of continuant.

product(x) → BFO:continuant(x)

Comment: 'Product' is a defined class combining manufactured products and information products. The latter will be addressed in a later version of this axiomatization.

4. **material resource** subclass of manufacturing resource.

material-resource(x) → manufacturing-resource(x)

Comment: This term is here defined as meaning, not raw material, but rather resources available to the enterprise that are made of matter (thus buildings, vehicles, equipment, and so forth, as contrasted with intellectual property, software, and so forth).

5. Definitions and Axioms

A definition of a term T is a statement of jointly sufficient and individually necessary conditions which an entity must satisfy if it is to be an instance of the universal or class referred to by T. In the ideal case we would provide equivalent definitions of all terms both in natural language (using '=def.') and in FOL syntax (using the biconditional '≡'). In some cases we fall short of this ideal, for instance for primitive terms, where we can provide only necessary (but not sufficient) conditions (labelled using '→'). Primitive terms are marked as such in what follows. In some cases we can provide English language definitions but not equivalent FOL definitions (for instance because FOL does not include the

resources to capture possibility or necessity). Note that where the definitions here presented use terms from BFO, IAO or CCO, the corresponding definitions – available at in [22], [23], and [24] – will be presupposed.

5. **product** =def. BFO:continuant that has a product role.

product(x, t) =def. instance-of(x, continuant, t) & $\exists r$ product-role(r) & (has-role(x, r, t)

Comment 1: This definition leaves open the possibility that immaterial continuants – for example pieces of software or real estate – may be products.

Comment 2: The BFO term 'role' is used extensively in this formalization to do justice to those cases where general terms hold only for certain phases in the existence of the relevant entities. Thus, for example, a given material entity may be correctly describable as a *prototype* at one phase in its existence and as a *product* in a later phase.

6. has-agent: a primitive relation between a process and an agent, which holds when the agent participates in the process and plays a causal role in bringing about the process.

has-agent(x,y) → (instance-of(x, agent, t) & instance-of(y, process, t))

Comment: The inverse of has-agent is agent-in.

7. action =def. process that has-agent some agent.

instance-of(x, action, t) ≡ $\exists y$ (instance-of(y, agent, t) & has-agent (x, y, t))

8. agent =def. person or organization (Defined class)

instance-of(x, agent, t) ≡ (instance-of(x, person, t) ∨ instance-of(x, organization, t))

Comment: This axiom implies that 'agent' is not a role term in the BFO sense. It implies also that it is not a phase sortal.

9. Every **manufacturing tool** bears a function that, if realized, is realized in a manufacturing process.

manufacturing tool(x) ≡ material-entity(x) & $\exists f$ (BFO:function(f) & has-function(x, f) & ∀y(realizes(y, f) → manufacturing process (y)))

10. **planned process** =def. process that occurs as the result of one or more intentions to realize a plan and where the process successfully realizes that plan.

instance-of(x, planned process, t) ≡ instance-of(x, process, t) & $\exists y$(instance-of(y, plan, t) & realizes(x, y))

Comment 1: This definition implies that every planned process is a process but not vice versa. Thus, 'planned' is here functioning as a specifier, rather than as a modifier analogous to 'cancelled' or 'averted'. Therefore, to say that a process is planned is not to say that it has not yet taken place; rather, it is to say that it is (was or will have been) protocol-driven, instruction-driven, command-driven, or software-driven (or some combination thereof). A planned process remains a planned process even after it has occurred. Hence, the contrast is with *accidental*

processes or with processes that did not turn out as planned (for example because a fire broke out in the engine room) or with incidental processes not part of the realization of a plan (such as the lathe operator smoking a cigarette while operating her lathe).

Comment 2: 'Planned' means 'protocol driven'. Protocols may be written, spoken, or simply thought – as when upon waking up, we plan, for instance, what to eat for breakfast.

Comment 3: 'Plan' is a synonym of 'plan-specification.' An active plan is a plan that is associated with an intention-to-perform on the part of some agent.

11. manufacturing enterprise =def. organization whose function is to engage in manufacturing processes.

manufacturing-enterprise(x) ≡ organization(x) & ∃f ((function(f) & has-function(x, f)) & ∀y (realizes(y, f)) → manufacturing enterprise process (y))

12. manufacturing resource role =def. role that inheres in an independent continuant where that continuant bears a disposition that, if realized, is realized in a manufacturing enterprise process.

instance-of(x, manufacturing-resource-role, t) ≡ ∃y(instance-of(y, continuant, t) & has-role(y, x, t) & ∃d(instance-of(d, disposition, t) & has-disposition(y, d, t)) & ∀p((process(p) & realizes(p, d)) → manufacturing-enterprise-process(p)))

13. **manufacturing resource** =def. continuant that bears a manufacturing resource role.

instance-of(x, manufacturing-resource, t) ≡ ∃y(manufacturing-resource-role(y) & has-role(x, y, t))

14. **manufacturing material resource** =def. manufacturing resource that is a material entity.

instance-of(x, material-resource, t) ≡ instance-of(x, manufacturing-resource, t) & instance-of(x, material-entity, t)

15. **piece of equipment**: primitive term that refers to a material artifact that is used in an operation or activity.

Comment**:** 'Equipment' is a synonym of 'piece of equipment'.

16. **piece of manufacturing equipment** =def. piece of equipment that bears a function where any process that realizes that function is a manufacturing process.

piece-of-manufacturing-equipment(x) ≡ piece-of-equipment(x) & ∃f(has-function(x, f) & ∀p(process(p) & realizes(p, f) → manufacturing-process(p))

17. supplier role =def. role inhering in an agent that, if realized, is realized in some act of selling.

supplier-role(x) ≡ ∃y(agent(y) & has-role(y, x) & ∀p((process(p) & realizes(y, p)) → act-of-selling(p))

18. customer role =def. role inhering in an agent and realized in an

act of purchasing, and which comes into exist at the point in time when a purchasing act has been initiated through to completion.

instance-of(x, customer-role, t) ≡ ∃y, z(agent(y, t) & has-role(x, y, t) & ∃w(instance-of(w, act-of-purchasing, t) & agent-in(y, w, t)))

19. prospective customer =def. agent capable of performing an act of purchasing.

prospective-customer(x) ≡ ∃y(capability(y) & has-capability(x, y) & ∀p(realizes(y, p) → act-of-purchasing(p))

20. has-specified-output: a primitive relation between a planned process and an entity where the entity satisfies the process endpoint specification in the plan specification.

21. **manufacturing process** =def. planned process that is an occurrent part of a product production process in which one or more material entities that will be part of a manufactured product are modified.

manufacturing-process(x) → ∃y(product-production-process(y) & occurrent-part(x, y))

Examples: Drilling a hole on an engine block, making a shaft (via milling, turning, and drilling, assembly process (including contract manufacturing).

Comment 1: There are also types of processes which are complements of or auxiliary to manufacturing processes. Examples are: adjusting a drilling machine in preparation for drilling a hole, changing a tool on a drilling machine, handling of objects (moving a part from one location to another by a robot), inspection of the manufacturing process / line / equipment).

Comment 2: The proposed definition of manufacturing process presupposes that the outputs of a manufacturing process are in every case material entities. Processes analogous to manufacturing with digital outputs such as pieces of software will be treated at a later stage.

22. product production process =def. planned process that has specified output some product that leaves the production facility for distribution and sale, where the product did not exist prior to the planned process.

product-production-process(x) ≡ ∃y(continuant(y) & has-output(x, y) & ∃t,t'(exists-at(y, t) & precedes(t', t) & ¬exists-at(y, t')))

Comment 1: A product production process is distinct from a maintenance process in that, in the latter case, the product exists both before and after the process occurs.

Comment 2: A product production process has several planned processes as parts (sub-processes) including manufacturing process, packaging process, transportation process, etc. Note that for example, maintenance is not a product production process.

23. product role =def. role inhering in an entity that is the specified output of a product production process.

Comment 1: Naturally found entities such as seashells are not products, according to this definition; they become products only if, for example, they are packaged in a certain way.

Comment 2: Parcels of real estate, according to this definition, are excluded from the realm of products.

24. maintenance process (primitive): planned process that has the same manufactured product as both specified input and specified output and that involves some material transformation of this product.

maintenance-process(x) → planned-process(x) & ∃y(manufactured-product(y) & has-specified-input(x, y) & has-specified-output(x, y)

25. **transport process** (primitive): planned process in which a material entity is moved from one site to another.

transport-process(x) → planned-process(x) & ∃y∃t∃t'∃s∃s'(material-entity(y) & occupies(y, s, t) & occupies(y, s', t')) & earlier-than(t, t') & ¬overlaps(s, s')

26. requirement (primitive): an entity that is specified in a requirement specification.

Comment: This is a defined class (almost certainly to be defined by enumeration).

27. **design** (primitive): directive information content entity that has product requirements as parts.

design(x) → directive-information-content entity(x) & ∃x_1, x_2, …, x_n∀i(product-requirement(x_i) & part-of(x_i, x)))

Comment 1: Note that this is merely a statement of necessary conditions. It is not intended to provide a full definition of 'design'.

Comments 2: 'Prescribes' is a primitive relation. A prescribes B means: A is some information content entity that tells us how the world has to be for it to conform to A. For example, a command prescribes how you should behave in order to conform to the command. A quality specification prescribes how a product has to be in order to conform to the quality specification.

Comment 3: 'Prescribes' can be understood in terms of Searle's idea of a world to mind direction of fit – where prescribing occurs there is a portion of reality that involves something like an attempt by an agent to make the world fit what the agent intends. Mind-to-world direction of fit occurs where a mind tries to make itself fit – for example, in its assertions or beliefs – the world [19].

28. specification =def. directive information content entity that prescribes some part or feature or some outcome of a planned process.

Examples: action specification, objective specification, plan specification, quality specification, requirement specification.

29. **quality specification** =def. specification that prescribes one or

more qualities.

quality-specification(x) ≡ specification(x) & ∃Q(specifies(x, Q) & ∀q(instance-of(q, Q) → quality(q)))

Comment: A quality specification specifies a quality as something that is to come into existence as a result of a planned process. It specifies this quality generically, that is to say, on the level of general type or universal.

30. objective specification =def. specification that specifies an intended process endpoint.

Comment: When an objective specification is part of a plan specification, the BFO:concretization of the latter is realized in a planned process in which thebearer tries to affect the world so that the process endpoint is achieved. This is another example of mind-to-world direction of fit.

31. **plan** =def. intention-to-perform processes on the part of an agent as prescribed by a plan specification.

Comment 1: If the agent of a plan is an aggregate of persons (for example an enterprise, a team), then the intentions will be relevant intentions will rest on the intentions of the persons involved. Where persons of different levels of authority are involved, respective plans and subplans will be correspondingly nested.

Comment 2: A plan is a BFO:concretization of a plan specification, namely the concretization in which the intention-to-perform (disposition) is first established. Where the plan specification is generic – for example because it leaves open the time when the plan should be realized – the plan itself may incorporate further specificity.

32. production plan specification =def. plan specification that prescribes a manufacturing process.

33. **production plan** =def. plan that is specified by a production plan specification.

34. **operation specification** =def. specification that specifies some manufacturing enterprise process and the resources required to perform the process by specifying some partially ordered sequence of steps.

35. operation =def. process that is specified in an operation specification.

36. **assembly process** =def. planned process whose plan specification specifies an endpoint that includes a completed assembly as output.

37. subassembly =def. assembly that is intended to become part of a further assembly.

∃x(instance-of(x, subassembly, t) ≡ instance-of(x, assembly, t) &
∃y∃z(instance-of(y, assembly-process, t) & (instance-of(z, assembly-process, t) & ¬(y=z) & has-process-part(z,y) & has-specified-output(y,x) & ¬(has-

specified-output(z,x)))

38. component role =def. role inhering in a material artifact that is designed to be a proper continuant part of some material artifact.

39. **component** =def. material artifact that bears a component role.

instance-of(*x*,component, *t*) ≡ instance-of(*x*, material-artifact, *t*) & ∃y(instance-of(*y*, component-role, *t*) & has-role(*x*, *y*))

40. machine =def. material artifact that has a mechanical system as part.

41. mechanical system =def. engineered system that realizes its function through the use of power to apply forces

42. system =def. object aggregate whose member parts are causally integrated.

Examples: solar system, digestive system, forest ecosystem, hydraulic system, subway system.

43. manufactured system =def. engineered system that is the specified output of a manufacturing process.

Comment: 'Engineered system' is broader than 'manufactured system' – the Milan subway is an engineered system but in addition to manufactured parts it includes also personnel and real estate.

44. **manufacturing machine** =def. machine whose function is realized in a manufacturing process.

instance-of(*x*, manufacturing-machine, *t*) ≡ instance-of(*x*, machine, *t*) & ∃*y*(has-function(*x*, *y*) & ∀z(realizes(*f*, *z*) → manufacturing-process(*z*)))

7. Conclusion

Disciplines such as mathematics, physics, and chemistry have long been subject to powerful incentives toward uniformization of terminology, and the effects of these incentives are manifested also in the SI system of units and other standards. In manufacturing, in contrast, the corresponding incentives have been rather weak, so that very many of the definitions provides here – above all, of terms such as 'machine', 'tool', 'plan', 'task', 'operation' – will be contested by one or more potential users of the IOF framework. To address this issue, we propose that the IOF signature and its IOF-BFO-FOL formalization be conceived as having a (weakly) legislative function. That is, we are *recommending* that, in IOF contexts, the terms set forth here be used with the meanings specified in the definitions set forth in this document. Where IOF-associated persons or groups need to accommodate alternative entrenched uses, appropriate mappings should be created, for example by use of equivalence relations and appropriate namespace IDs to signify that a given usage is associated with a specific community.

References

[1] Almeida, M. et al. (2015). *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]

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

[3] Munn, K. & Smith, B. (eds.). (2008). *Applied Ontology: An Introduction*, Frankfurt/Lancaster: ontos/Walter de Gruyter.

[4] Otte, J. Neil, Dimitris K, Munira Mohd A, Yang, R, Zhang B, Rudnicki R, Rahul R, Smith B. (2019). An Ontological Approach to Representing the Product Life Cycle. *Applied Ontology* 14 (2):1-19.

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

[6] Rudnicki, R. (2016). *Modeling information with the Common Core Ontologies*. Available at: https://github.com/ CommonCoreOntology. [Online; accessed 25 April 2019].

[7] Kulvatunyou B, Wallace E, Kiritsis D, Smith B, Will C. (2018). The Industrial Ontologies Foundry proof-of-concept project, *APMS: Advances in production management systems. Smart manufacturing for industry 4.0* (IFIP International conference on advances in production management systems, 536). Springer. p. 402-409.

[8] Furini F, Rai R, Smith B, Colombo G & Krovi V. (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), Charlotte, North Carolina, USA.

[9] Suwelack S, Markus Stoll M, Serf M, Bursac N, Albers A, Bendl R, Dillmann R & Speidel S (2016). Towards Cognitive Computer Aided Engineering, *Proceedings of the International Association for the Engineering Modeling, Analysis and Simulation Community*, NAFEMS Americas Conference, June 2016.

[10] Correia A, Stokic D, Siafaka R, Scholze S. (2017). Ontology for Collaborative Development of Product Service Systems based on Basic Formal Ontology, *International Conference on Engineering, Technology and Innovation* (ICE/ITMC).

[11] Hagedorn TJ, Krishnamurty S & Grosse IR (2018). A Knowledge-Based Method for Innovative Design for Additive Manufacturing Supported by Modular Ontologies], *Journal of Computing and Information Science in Engineering*, 18(2).

[12] Bone M, Blackburn M, Kruse B, Dzielski J, Hagedorn T & Grosse IR (2018). Toward an Interoperability and Integration Framework to Enable Digital Thread, *Systems*, 6 (4), 46.

[13] Mesmer L & Olewnik A, Enabling Supplier Discovery Through a Part-Focused Manufacturing Process Ontology], *International Journal of Computer Integrated Manufacturing*, 31(1), 87-100.

[14] Arena D, Ameri F, Kiritsis D, Skill Modelling for Digital Factories (2018) *IFIP International Conference on Advances in Production Management Systems* : *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0* (APMS 2018), 318-326.

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

[16] Otte JN, Kiritsis D, Ali MM, Yang R, Zhang B, Rudnicki R, Rai R, Smith B (2019). An Ontological Approach to Representing the Product Life Cycle, *Applied Ontology*, 14 (2), 155-177.

[17] Karray H, Ameri F, Hodkiewicz M, Louge T (2019). ROMAIN: Towards a BFO compliant reference ontology for industrial maintenance, *Applied Ontology*, 14 (2), 179-197.

[18] Hagedorn TJ, Smith B, Krishnamurty S, Grosse IR (2019) Interoperability of disparate engineering domain ontologies using Basic Formal Ontology, *Journal of Engineering Design*, June 2019, https://doi.org/10.1080/09544828.2019.1630805.

[19] Searle, J.R. (1985) *Expression and Meaning: Studies in the Theory of Speech Acts*, Cambridge University Press, Cambridge.

[20] Bratman, Michael. (1987). *Intention, Plans and Practical Reason*, Cambridge, MA: Harvard

University Press, 28-30.

[21] Smith, Barry. “On Classifying Material Entities in Basic Formal Ontology” (2012). *Interdisciplinary Ontology. Proceedings of the Third Interdisciplinary Ontology Meeting*, Tokyo: Keio University Press, 1-13.

[22] The Common Core Ontologies (CCO). Accessed August 5, 2019 at https://github.com/-CommonCoreOntology/CommonCoreOntologies

[23] Basic Formal Ontology (BFO). Accessed August 5, 2019 at https://github.com/BFO-ontology/BFO

[24] The Information Artifact Ontology (IAO). Accessed August 5, 2019 at https://github.com/information-artifact-ontology/IAO