

Ontology Summit 2012 Communique

Ontology for Big Systems

Lead Editors: AliHashemi & ToddSchneider

Co-Editors: MikeBennett, MaryBrady, CoryCasanave, HensonGraves, NicolaGuarino, AnatolyLevenchuk, ErnieLucier, LeoObrst, SteveRay, AmandaVizedom, MatthewWest, TrishWhetzel

1. Focus & Scope

The 2012 Ontology Summit, *Ontology for Big Systems*, sought to explore, identify and articulate how ontology, its methods and paradigms, can bring value to the architecture and engineering of Big Systems throughout their full lifecycles.

The term Big Systems was intentionally vague and intended to cover a large scope that included many of the terms encountered in the media and engineering including:

- Big Data and the systems that handle it
- complex systems including those that support processing, physical or information, and socio-technical economic interactions/processes
- intelligent or smart systems

Additionally, though not necessarily explicitly Big Systems, we also included cloud computing and net-centric environments, which represent areas also addressed by systems engineering that will benefit from the use of ontology.

Established disciplines that fall within the summit scope include (but are not limited to) systems engineering, software engineering, information systems modeling, enterprise architecture and data mining.

As is traditional with the Ontology Summit series, the results were captured in the form of a communiqué (herein), with expanded supporting material provided on the [web](#).

2. Summary

The principal goal of the summit was to bring together and foster collaboration between the ontology community, systems engineering community, and stakeholders in Big Systems. The common thread that emerged for Big Systems and Big Data was models and modeling; the status of models as an authoritative source of information for these systems; the need to have models with greater fidelity and interoperability that adequately represent the complexity of the systems and their (operational) environments. The primary driver for a modeling approach to systems engineering and development is complexity, cost, in time/money/maintenance/reuse, and resultant system value. Ontology, both in the guise of ontological analysis and ontologies as artifacts, provides the basis for meeting the complexity and cost challenges of engineering Big Systems and handling Big Data in terms of more explicit semantics - fidelity and verisimilitude to real world – and consistent conceptualizations.

Among the current approaches to mitigate some of the complexity and cost factors associated with engineering are executable architectures and model based engineering. Each approach involves a model to either understand the thing being designed or to provide a predictive base of design. In each case current methodologies and tools often fall short of providing:

- sufficient rigor in their ability to adequately represent the system for the needs of the entire engineering lifecycle and its environment,

- adequate ontological analysis of the domain or its constituent parts
- explicit semantics (usually only in the minds of the modelers and therefore prone to variation between modelers and inconsistency across disciplines).
- the use of logical inferencing to automate processes.

The lack of adequate fidelity among models and their conceptualizations, and of consistent semantics during engineering phases can cause poor design, mis-communication across the lifecycle and among stakeholders, implementation errors, re-work, and systems that fail to neither meet their expected uses nor cost-effectively be extended to meet unanticipated needs. During operation such systems may be difficult to maintain, including simple maintenance, updating, or even extensions. Moreover, there is a growing expectation for systems to be more 'intelligent', in the sense of being able to adapt, or at a minimum be adaptable, to new needs without incurring large costs.

The information age has resulted in the production of unprecedented amounts of information and data - Big Data. Accompanying, or causing, this abundance of data and information are Big Systems that create or attempt to handle it and provide something akin to "knowledge". Something that integrates the best, most appropriate information, something that not only reflects the vast quantity of data, but also its meaning and authority, including the meaning, authority, and intention that may be derived from the context in which it originates.

But the growth of these 'Big' things are outstripping the capacities of current engineering practices and tools. Ontologies and ontological analysis are vital parts of a solution addressing the problems of architecting and engineering Big Systems and Big Data. Whereas data models and conceptual schemas typically only provide local and/or ad hoc semantics, ontologies explicitly represent real-world semantics of the systems. Ontologies can be used to:

- make explicit and accessible the implicit yet vital assumptions about nature and structure of engineered systems and their components
- help people better understand and disentangle the complexity of big engineered systems and their social, economical, and natural environment
- enable integration among systems and data through semantic interoperability
- allow humans to delegate more of the mundane processing and computing to machines (than was previously possible)
- improve models and modeling, their adaptability and reuse, and resulting design
- reduce development and operational costs
- enhance decision support systems
- aid in knowledge management and discovery
- provide a basis for more adaptable systems

Finally, as we move into the knowledge age with Big Systems and Big Data, there is a growing expectation that our systems will be more self-describing and intelligent. While for smaller systems, it may be viable to rely on implicit semantics and manual modifications, the scale, complexity, heterogeneity, and costs of Big Systems or Big Data often exclude such an option. Rather, the semantics must be made explicit and machine readable, because there are multiple communities, users and developers who are involved throughout the lifecycle. In order to engineer and operate such systems cost effectively, allow intuitive use and meet expectations of all stakeholders, a more consistent and complete use of ontologies and ontological analysis must be made.

3. Introduction

In the past decade, more data has been collected, more video has been produced and more information has been published than in all of previous human history. At the same time, with the advent of the computer, digital representations, and the Internet, it has been possible to model more of the complexity of systems, connect more people, and connect more (information) systems. With all this new information (aka Big Data) and all these new

systems (aka Big Systems), there has also been an attendant growth in the complexity and scale of systems that model physical phenomena and handle information, their size, scale, scope and interdependence.

To address the problems that have arisen during the current period of information and knowledge growth, we need novel tools and approaches. Some of the major challenges facing Big Systems stem not only from their scale, but also their scope and complexity. At the same time, there are novel challenges for Big Systems when different, dispersed groups work together toward a common goal, for instance in understanding climate change. This leads to a need for better solutions for interoperability among federated systems and for fostering interdisciplinary collaboration.

Given the broad scope of this year's theme, *Ontology for Big Systems*, the summit was organized along three tracks and two cross-track initiatives. This communiqué seeks to distill and construct a whole from the activities that occurred within each track and throughout the summit. The interested reader is encouraged to visit the synthesis and community pages for further information. In addition each of the meeting pages, containing links to the presentations, audio recordings, and chat sessions is also available for review. The tracks were as follows:

- [Big Systems Engineering](#)
- [Big Data Challenge](#)
- [Large Scale Domain Applications](#)
- [Quality Cross Track](#)
- [Federation and Integration of Systems](#)

3.1. Big Systems Engineering

Engineers and designers have always used a variety of models as part of their disciplines. Designing a car, a power plant, information application, or a transportation system relies heavily on creating a model of the system. Similarly, models are used extensively in trying to understand how complex systems such as the human body or climate works. In the computing age, it has become far easier to create and share these models, and given the scale and complexity of the systems being modeled, these models are becoming the authoritative source.

However these models carry an (often implicit) ontology, expressing a theory or a set of assumptions, about the world or some part of it. But different fields create and use models of varying sophistication whose underlying conceptualization and/or intended semantics is often implicit or governed by inconsistent conventions. But the reuse of these models is hindered by these differences. So a gradual shift to explicit semantics and consistent conceptualizations is underway, first in engineering and slowly in other fields.

Within engineering, the various disciplines are evolving from using informal modeling to using formal languages to model their systems; to underpinning these languages with explicit semantics; to recognizing the importance of understanding the underlying ontology of modeling primitives. This ontology is based on real world characterizations and categories, not just the local semantics of structural data models. Ontological analysis helps to ensure proper shared understanding of fundamental relations such as “component”, “sub-class”, or identity-preserving properties that persist through time as designs and implementations change.

There are various standardization efforts underway to advance the semantic and ontological foundations, from the development of [ISO 15926](#) (a standard for data integration, sharing, exchange, and hand-over between computer systems), to providing formal semantics for the [Unified Modeling Language](#). Similarly, groups are working to build repositories of ontologies, or libraries of [ontology patterns](#) - snippets that formalize important aspects of reality such as “part-of” or “is-a”. Additionally, domain-specific languages such as Haskell and other functional programming and modeling languages require firm grounding in explicit semantics that can only be provided by formal ontologies.

3.2. Big Data & Applications

A key component of the current explosion of information is the proliferation of vast amounts of raw data. With greater computing power there is an increased ability to create and track data. Whether it be encoding an organism's DNA, tracking Internet usage, tracking credit usage, the experiments at the Large Hadron Collider or weather satellite data, each of these activities creates a staggering amount of data. A future in which the ability to analyze and

extract information from large, diverse, and disparate data sets:

- accelerates the process of scientific discovery and innovation
- promotes new economic growth
- leads to new fields of research and new areas of inquiry that would otherwise be impossible.

The sheer size and scale of these data sets presents its own challenge, knowing how to first understand the data, garner information and knowledge from it, and then intelligently combine it with other data sets means that there is a need to accurately represent (the portion of) the world this data reflects. This in turn necessitates that each data source adequately represents itself and makes available information that can be interpreted out of its original context, for example units of measurement, time-stamps, or annotations of data elements with terms from reference ontologies. To effectively reuse and combine data from different sources and contexts in novel ways there must be sufficient commonality among the information that describes the data.

For example, [imagine a future](#) where intelligent agents play a more prominent role in the doctor-patient relationship. As a patient describes her symptoms to the doctor, an agent is able to cross-reference these symptoms with aggregated patient data to find similar patient profiles. Unable to determine the exact ailment, the doctor uses this information to prescribe a series of test to further narrow the possibilities. Before the tests are carried out, a new paper is published linking a previously unknown gene to a symptom displayed by the patient. An agent monitoring this publication, extracts this information and flags the patient file for doctor review. The next day, the doctor is alerted to a change and realizes that a number of tests prescribed are unnecessary. Such functionality would be the manifestation of a number of federated Big Systems (patient data, research publication networks, gene information systems).

Realizing this vision will require a multitude of technologies and approaches. One tool currently used to understand how different data sets are related to one another is statistical analysis, but there are limits to statistical analysis. There needs to be a conceptual framework or theory alongside statistical analysis tools. To effectively combine multiple data sets and systems, we need to be able to represent the assumptions and conceptualizations that underpin knowledge in those domains.

To be able to effectively use the data and combine it for other useful ends, data creators and publishers need to make explicit what their data represents together with the context of the data and its creation (e.g., the systems that created and transformed it). This requirement necessitates developing theories about those parts of the world relevant to the data and its contexts. Without such theory and subsequent practice, successful data reuse and adaptability will not be possible.

Of note is the work in bioinformatics, such as the [Gene Ontology](#), and other ontology artifacts found in the [OBO Library](#) or [Bioportal](#), which could annotate Big Data with explicit semantics. These initiatives allow research groups to publish findings on genes, gene expression, proteins and so in a standardized consistent manner.

Another example is the [FuturICT](#) project funded by the European Union. Its ultimate goal is to understand and manage complex, global, socially interactive systems, with a focus on sustainability and resilience. FuturICT will build a Living Earth Platform, a simulation, visualization and participation platform to support decision-making of policy-makers, business people and citizens. Further examples can be found on the [track four teleconferences](#).

3.3. Federation, Integration & Interoperability

The Internet has made it far easier for different people in the different parts of the world to share and combine data, information, and knowledge. If the true potential of this interconnected world is to be realized it means that we need to be able to combine not just our data, but also our systems, models, conceptualizations, and semantics.

As knowledge has become more specialized, different communities have developed their own bodies of knowledge, vocabularies, or interpretations of common terms. Each community (of practice) views and prioritizes parts of the world according to their own viewpoints and interests, having their own implicit semantics, with competing goals. Similarly, within a single enterprise, the same product or data may be viewed differently by each of the marketing,

engineering, manufacturing, sales and accounting departments, each applying their own terminology and possibly different conceptualizations. Ensuring that these views are, if not harmonized, at least aligned so that information can be shared and used effectively entails solving interoperability. Without interoperability, information from these different departments cannot be combined or reused accurately or effectively. Leaving the enterprise stakeholders, including decision-makers at every level, without ready, reliable access to what the rest of the enterprise knows. Attempts to bridge such information or knowledge gaps without explicit semantics can also leave the enterprise weighed down by additional costs, inaccuracies, and latency in creating and maintaining duplicate information sources.

Semantic analysis (understanding the meaning of terms used by different systems or organizations), followed by ontological analysis is a fundamental, essential aspect of federation and integration – providing a consistent interpretation of the (natural language) terms used in systems and data sets. Ontologies, in the form of explicit statement of the assumptions in each sub-field can help identify points of overlap and interest between different communities. The ontologies can serve as tools to facilitate search and discovery. Building value by combining the views of different communities means solving interoperability, and that means negotiating the meanings or interpretations, implicit or otherwise, used by each of these groups.

The [Object Management Group](#) recently released a request for proposals to create a standard to address issues, such as the request for proposal regarding the “Semantic Information Modeling for Federation” ([SIME](#)). Similarly, one example within the systems engineering community is the [ISO 15926](#) standard which aims to provide a capability to support the federation of the design ([CAD](#)), manufacturing ([CAM](#)) and lifecycle ([PLM](#)) systems in industry, business and ecosystem-wide scales. A set of references regarding the subject of this cross track have been compiled and posted to the ontolog wiki, available [here](#).

Another project, the [iPlant Collaborative](#), is building the requisite cyberinfrastructure to help cross-disciplinary, community-driven groups publish and share information, build models and aid in search. The vision is to develop a cyberinfrastructure that is accessible to all levels of expertise, ranging from students to traditional biology researchers and computational biology experts.

3.4. Ontology & Quality

While addressing our main theme, *Ontology for Big Systems*, we can't ignore of course the ontology quality issue. The word “Quality” may be used to describe how “good” something is in some way independent of usage, but “quality” is also used, in industrial quality assurance, to describe how well some deliverable meets its stated requirements. To the extent that it is possible to state things about an ontology which make it “good,” these two definitions may converge, but they should be considered separately.

Quality in its most formal sense refers to the rigorous use of requirement specifications, requirements-centric design, multi-stage testing and revision, and other risk-management and quality assurance techniques. This is a hallmark of systems engineering, distinguishing it from less rigorous systems creation activities and essential to success in developing large-scale and complex systems and managing them throughout their life-cycles. Various sub-domains within systems engineering apply these risk- and complexity- management techniques to systems overall, to system components, to component interfaces, to engineering, interface, and other processes. Quality at any of these levels is defined in terms of the degree to which any one of the system, component, process, etc., meets the specified requirements. Analysis and specification of requirements and functions at each of these levels, along with identification and application of relevant quality measures, is an essential part of good systems engineering.

The key to formal quality in any context is that the requirements must be well-articulated. Ontologies present specific challenges in this regard, starting even at the most basic question: “What is this ontology for?”. There is considerable literature on measures which may be applied to ontologies ‘in vacuo’, which may allow one to make some assessment of how good they are with respect to their general aim, i.e., making a conceptualization explicit, avoiding misunderstandings about a particular term. Less well-developed, but increasingly important, is the literature on how to formally articulate the range ontology characteristics such that for a given application, those characteristics may be

specified as requirements, and the ontology may be assessed against those formal requirements within some formal quality assurance framework or regime.

Ontologies may be used in one or both of two very distinct ways: as a formal computational artifact which forms part of some system, and as a tool for formally articulating business subject matter as part of the specification and engineering of some system. In Big Systems engineering the latter use case starts to come into its own. Ontologies which have been developed to articulate business knowledge within the big system development process may go on to be deployed as computational artifacts within one or more components of that system, again highlighting the importance of articulating the uses to which the ontology is to be put. Independent of such deployment, however, the use of ontologies to represent business subject matter, systems specification, processes, functions and other matters important to the systems engineering process provides a certain rigor while also enabling ontology-based reasoning about those matters either now or in some future application.

In either case, the formal quality of the ontology used matters as much as the formal quality of other systems components and tools. By definition, the formal quality of the ontology is the degree to which the ontology meets the specified requirements. Those requirements are derived from the usage, as a component of an engineered system or as a part of an engineering process. For effective ontology quality assurance, these requirements must be specified.

In practice, however, even in systems engineering projects in which attention is rigorously applied to quality assurance measures for other components and aspects of those systems, often the commensurate identification and specification of ontology requirements and the subsequent validation of the delivered ontologies against those requirements is given little to no attention. Where ontologies have been used specifically as part of the quality assurance process for other system deliverables (that is, using ontologies to articulate the business knowledge that is to drive data model development or systems components), there is often a perception that quality measures do not need to be applied to the ontology itself. This is far from being the case.

In order to validate the quality of an ontology then, it is necessary first to identify what will be required of that ontology in use. A formal approach is required no matter how simple may be the use to which the ontology is to be put. In order to fully describe the formal requirements of an ontology, it is first necessary to articulate what are the things which may be said of an ontology, in order to determine whether or not those are specific requirements for the ontology that is to be delivered in a given project. The kinds of things which need to be articulated include logical formalisms, the treatment of meaning, coverage of the subject matter semantics, the ontological commitments logical characteristics of the ontology and so on. There is a wealth of literature on certain aspects of ontology requirements and quality, but considerably less on other areas such as ontological commitments or the more “semantic” issues (as distinct from requirements which may be validated by some automated means). If this is not addressed there is a danger that ontology engineers are “looking for the keys under the streetlamp” by applying only those techniques which are amenable to some automated treatment.

Issues around real meaning are less amenable to these technical treatments, but not incapable of validation. Some techniques are emerging which provide some means to better address the actual semantics of ontologies, in particular ontology patterns, the use of industry standards, and techniques around competency and coverage. Numerous techniques for validation of models by domain experts exist in practice; documentation of these techniques and evaluation of their soundness are needed.

4. Recommendations & Observations

This section represents a distillation of the discussion in this year’s summit focused on recommendations and observations, beginning with a listing followed by more detailed explanations.

4.1. Recommendations & Observations Summary

Modeling

- Modeling should employ ontological analysis and patterns
- Modeling should employ foundational ontologies to provide consistent conceptualizations

- Modeling languages need to support explicit semantics and conceptualizations grounded in foundational ontologies

Engineering Practice

- The ontology community needs to develop ontology patterns to facilitate adoption and use of ontology in engineering
- Engineering processes have to be expanded to include requirements for ontologies
- Systems engineering should include ontological analysis as part of its standard practice
- Modeling practices should recognize the value of conceptual models and understand the differences between them and logical models and implementations
- Provenance for design rationale and implementation decisions need to be maintained

Ontology Tools & Infrastructure

- Tools for ontology development need to be improved and integrated with tools from other modeling paradigms
- Configuration control tools and processes need to be extended and expanded for ontologies and their artifacts (e.g., provenance)
- Ontology repositories with common interfaces and common metadata need to be readily available

Ontology Quality Practices

- Quality requirements and metrics for ontologies need to be developed and integrated into engineering practices

4.2. Modeling

Most aspects of engineering involve models, many times residing solely in the engineer's mind. In the process of engineering Big Systems there are many (possibly complex) models developed by different disciplines, teams and people that may be geographically, linguistically, and culturally dispersed. But, models from different disciplines have different levels of expressivity or fidelity, different assumptions, different degrees of automation, and are not interoperable in general. Aside from differences in tools and modeling syntax, more fundamentally, different and not necessarily compatible conceptualizations and interpretations exist among the models. At various points in the system's development and operational lifecycle(s) these differences must be resolved and models integrated, or at a minimum, differences bridged, to achieve interoperability, including syntactic, conceptual, and semantic, in order for collaboration and continued development to occur. These efforts to resolve incompatibilities add additional time and costs. Thus the Systems Engineering community gives a strong emphasis on the importance of models and modeling, and explicating the underlying concepts and their semantics.

Models incorporating formal ontologies can deliver additional value by exploiting the application of rules, inferencing and transformations between models. Current approaches to modeling include natural language textual descriptions, mathematical models, free form graphical diagrams (e.g., Microsoft Power Point or Visio), spreadsheets, or specification based notations (e.g., IDEF, Entity-Relationship diagrams, UML). A number of candidate modeling languages were considered in the discussions, alongside their deficiencies in semantic and conceptual clarity (ontology representation languages among those). In each of these cases there emerged a lack of clear conceptualizations or semantics.

Computer based modeling languages provide some built-in support for component modeling and provide facilities for extending the language's ontological commitment, but are usually not sufficient to support formal semantics, logical inferencing, nor expressive enough to take advantage of rigorous ontological analysis.

To mediate at least the possible semantic differences among models there has been a progression in engineering to shift from informal modeling toward more explicit semantics, for instance chalk/white board sketches or textual descriptions, to modeling in formal languages that support more explicit and complete semantics. However, beyond the issues of semantic differences of models, there can be, and are, differences in conceptualizations. These differences may not always be readily apparent and sometimes manifest in modeling languages.

The modeling of big, complex or distributed systems, such as linked open data (LOD), in which information data is

shared and used across organizational, specialty, geographic and even linguistic divides, requires conceptualizations within multiple domains of relevance to the system(s), their use(s), and engineering processes. Ontologies represent conceptualizations of aspects of a domain or its environment. Ontological analysis provides a more thorough analysis methodology for understanding and distinguishing the complexity of Big Systems. Modeling, in all its various guises, is an area where ontology and ontological analysis is starting to be used and has great potential, as exemplified by INCOSE's [ontology for Model Based Systems Engineering](#) effort..

Ontologies can be viewed as patterns for what constitutes a system (with parts, connections, processes or events), the identity, dependence, and unity of systems – models in their own right. Informally a system is an entity that consists of components, where the components are connected in some way such that the system as a whole exhibits some behavior. For engineered systems, it is usual for them to be designed such that the components are replaceable. Key relations like classification, specialization, and whole-part are well understood in the realm of ontology, and see major application in systems engineering.

It was further noted that developing an ontology of a problem space or domain as a referent conceptual model allows an organization to decouple this knowledge from any particular information model or technology implementation. In this way, a technology agnosticism is enabled, allowing the conceptual model to be reused and realized in whichever technology stack is most appropriate.

4.3. Engineering Practice

The intersection between ontology, Big Systems and Big Data spans many communities, disciplines, and levels of depth. Regardless of the community, the success of any ontology intervention requires understanding its intended environment and problem space to be addressed. Clarifying how ontology fits into the larger picture will shape what level of expressiveness and semantics is required and how they may be employed in a project. Not all ontologies need to be reasoned over and rarely are they the end product.

In considering the use of ontology one has to gauge the level of “semantic maturity” of the organization and environment in which the use is proposed. To what degree does the broader organization understand ontology or the application of ontology? To what extent are such technologies already being deployed? Will the shift be incremental or might it be perceived as disruptive? Often, existing infrastructure will support traditional software development far better than large-scale ontologies, developing a migration path that delivers small wins while transitioning towards a more suited infrastructure makes such change easier to manage. Given that no single technology or tool currently provides the best solution across all large system use cases, most implementations should expect to evolve as the technology landscape changes.

Determining exactly *which* ontology is appropriate for an application is an involved task. Ontology patterns allow engineers to construct ontologies incrementally, without committing to reusing an entire ontology, by selecting only those parts which address a limited scope. Selecting the right ontology requires trade-offs in terms of the desired expressivity, comprehensiveness and breadth. To this end, it was recognized that a number of distinct problems are often conflated in the case of procedural artifacts. It is wise to disentangle:

1. The level of expressiveness (representation) it takes to develop the ontology needed for your domain. This is development time expressiveness.
2. The level of expressiveness (representation) it takes to efficiently reason over the ontology at run-time. This is run-time expressiveness.
3. Transformation of the representation of (1) to (2), i.e., knowledge compilation.

Not enough expressivity may mean that it is not possible or cumbersome to represent essential aspects of the problem space. Conversely, allowing extraneous expressivity for reasoning can severely affect run time performance. A vital task for any ontology implementation is to understand the level of expressivity as required by the problem space, while also accounting for performance criteria. Moreover, reasoner and query engine performance are highly dependent upon the exact formulation of the rules and queries. Alternative representations of the same axioms can have significant effects in the performance of reasoning. One observation was that ontologies work best when not

compromised by implementation tradeoffs.

This means that greater work is required to build adequate support frameworks for such tasks, which is currently minimal. When it also comes to the deployment or construction of an ontology, while the target community should be included in the development and evolution of the vocabularies, engineers turned ontologists often don't have the necessary background or skills. That said, it is critical to maintain a strong relationship with the domain experts about the fidelity of the model.

The transition from implicit domain knowledge to explicit encoding requires community consensus, which in turn requires an organizational commitment to create the necessary infrastructure to manage such consensus. At the same time, consensus is not always possible, as different subgroups working on different parts of the same system may have differing views. In these cases, having explicit vocabularies (classifiers) is a necessity in a distributed system.

In those applications where the ontology will impact end users, there is broad consensus that the presentation of the ontology should be relevant to the users' context. For example, in one [successful project](#), ontologies were used as configuration templates which user interface specialists then used to tailor views for their end users.

4.4. Ontology Tools & Infrastructure

Systems engineering is all about understanding the whole and the relationships between the parts. It involves assembly from components and support for the use of the same parts in different systems. This calls for ontologies which can themselves be components of other ontologies and be assembled for an ontology of the whole system. Yet in general ontology developments are one-off with it being rare for ontologies to be reused or be reusable. For ontology to be useful for engineering reusable ontologies to support reusable engineering models will be important.

Big systems have a long life and usually change over that life. They tend to interact with their environment and change state as a result of interaction. This means conceptualizations are needed to model state change and system evolution throughout its lifecycle which in turn means that the ontologies that describe a system need to be able to change, but in a way where the history of changes is not lost. This requires a sophisticated approach to change and configuration management, both in model and ontology creation and maintenance.

When deciding what ontologies to use or implement, there is a consensus that where possible, ontologies should be reused from pre-existing sources. Two such sources were explored, Ontology Repositories and libraries of ontology patterns that represent successful representations of particular relations or "snippets" of a domain. The former have the advantage of providing a more comprehensive solution, while the latter afford greater flexibility and in theory, allow the designer to pick and choose among a variety of patterns to best meet their needs.

Foundational ontologies contain conceptualizations needed for modeling, especially at the enterprise scale. Ontologies to support dynamic concepts such as time ([OWL-Time](#)) and process ([PSL](#), [OWL-S](#)) have also been developed and applied within engineering scenarios. These include processes, events, descriptions, plans, physical quantities, individuals, types etc. Further ontologies provide relationships between the concepts which can be exploited to relate data needed to determine program status. Some enterprises have recognized that ontologies generalize information models and provide better access and organization than traditional data models.

4.5. Ontology Quality Practices

It's also been observed that the proliferation of ontologies has not been accompanied by adequate tools or methodologies to gauge the quality of the ontologies. Quality dimensions/criteria/attributes and measures vary with the specific project at hand. The ontology community currently does not have a clear understanding and virtually no documentation as to how that variation works. Experienced ontologists develop a sense of this, but it is implicit and not made accessible to others.

Are they fit for purpose? Any ontology project should not only pay attention to quality, but develop a quality policy. How would the organization measure the success of the ontology project? While there currently exists no standard methodology, there are some efforts within the literature. Consequently, a more systematic effort is needed. Concurrently, it is important to spread the understanding that ontologies need to be viewed as technical artifacts that

need requirements and quality assurance.

5. Conclusion

Big Systems can garner benefits in many ways from the use of ontology throughout their full lifecycles. To more completely integrate ontology and ontological analysis into the engineering community and its processes, the skills most needed include a combined understanding of a scientific or engineering discipline and knowledge of ontological analysis and ontology-based technologies. To realize this combination a transition based on existing paradigms and tools will need to be exploited in order to create the infrastructure, both technical and social (i.e. human systems integration), needed for quality ontology development and more general use.

In particular, the efforts by the Object Management Group (OMG) to provide a formal semantic underpinning to their Unified Model Languages and its derivatives (e.g., SysML) represent a step in the right direction. Moreover, organizations such as the International Council on Systems Engineering (INCOSE) are already engaged in fostering the use of ontological analysis and ontology in their communities.

The engineering ecosystem and Big Data users have much to gain from the use of ontology and ontological analysis. These capabilities can provide the key to engineer better systems, reduce costs and accelerate the process of scientific discovery and innovation.