Architectures for Complex Semantic Models

Robert B. Allen and Yoonmi Chu
Department of Library and Information Science
Yonsei University
Seoul, Korea
rballen@yonsei.ac.kr, yoonmichu@gmail.com

Abstract— **We have been exploring applications for complex semantic models such as rich-media science and history digital libraries. In this paper, we consider in more detail a range of services which could be implemented as well as technical details for those implementations. Many semantic tools are now available but these have rarely, if ever, been applied across broad and dynamic sets of complex instances. Modeling detailed histories involves complex entities interacting in complex ways. We consider architectures such as the Basic Formal Ontology and object-oriented models and we apply them in hybrid implementations using Jena/Java and Slate.**

Keywords—Abstraction, Agents, BFO, Community Models, Digital Humanities, Instances, Object Models, Programming Languages, Model-Oriented Semantics, Qualitative Simulation, Jena, Semantic Microworlds, Slate, Societal Models, State, Upper Ontologies, Verbs

I. The Challenge of Complex Dynamic Semantic Models

Our goal is to provide a structured yet flexible framework for interaction with complex sets of instance data such as those from human and natural science. There is a massive amount of such data. Having a clear notion of structure seems essential for effectively mining text and other media. We favor integrated models and thus go beyond loose linking of metadata.

While we have explored applications in geology [10], much of our previous work has focused on descriptions of human history. Specifically, we have been interested in information organization for the contents of digitized historical newspapers. As a strategy for that we have proposed modeling the community on which the newspaper reports [5, 6, 7]. When applied to individual towns, we term this strategy developing "community models" but the same issues apply to broader societal modeling and there are similar issues for natural science histories. Our model-oriented approach emphasizes modeling content directly rather than providing links to information resources.

Because we emphasize the description of content, we focus on qualitative models which are often based on natural language but data from other media may also be incorporated. In terms of information organization, we believe that the modeling of content rather than of the information objects that hold the documents all for a new generation of digital libraries [7, 9].

In terms of methods, our work touches many other streams of research. Perhaps it is closest in spirit to the OpenClinical semantic knowledge management and decision support project in medicine. [1] OpenClinical builds on the OpenGALEN [2] framework. It is also related to work on requirements for information systems [30], agent models for business process engineering [41], and to intelligent tutoring systems [18].

Here, we explore broad issues for developing representations for complex environments which change across time. We start by considering services and the requirements for representations to support them. We then consider approaches to implement those representations based on ontologies and object-oriented models. Finally, we discuss cognitive and linguistic approaches and additional elements to support user interaction.

II. Services and Requirements

We distinguish description from prediction. While much work on modeling is directed at prediction for what might happen, following the traditional role of historians we are interested in accurately recording what happened. Of course, we recognize that the evidence is not always a clear reflection of what happened and that it is quite possible for there to be more than one interpretation of the evidence. In short, our primary approach is descriptive modeling by which we mean developing models which are useful for description rather than for prediction.

We also distinguish description from explanation. Description might be limited to simple chronologies but explanation might go beyond that to include the attribution of causal relationships. Furthermore, explanations may be adapted suit the audience. We are interested in both description and explanation though in this paper we emphasize description since that can be modeled more directly. Explanation requires more interpretation than prediction but explanation can provide more context and support for the descriptions.

There are many higher-level services and applications such as scholarly analysis and hypothesis building, exploration of various versions, tutoring, guided tours, summarization, search and indexing, question answering, and digital gaming. Yet

[1] http://www.openclinical.org/

[2] http://www.opengalen.org/

other options are whether users can get an overview across time or whether they step only forward through time. And, the system should support multiple views of the same events. Moreover, there should be administrator tools and tool to "debug" models which show inconsistencies.

Consider a possible interface to support the exploration of a community as described in a historical newspaper; users who are unfamiliar with the town might start with an overview of the history and context of the community. Other users might be interested in a specific dimension (e.g., economic development, sports, religion, etc.) or maybe with specific people. Ideally, the interface would provide personalized views for these different types of users. The presentations could include automatically generated text and other types of context such as illustrations and maps. Links to relevant evidence and versions should also be supported. For some applications we might tolerate limited errors but advanced users should also be able to repair them.

Supporting services requires varying mixes of representations and capabilities. Some of the issues concern the development and management of the content. For instance, we should be able to coordinate across many types of resources (e.g., multiple newspapers, oral histories, and data tables). We should allow different types of inferences based on the type of evidence. There are many levels of inference some are based on logical necessity and some are highly normative inferences (e.g., that a person breathes or eats). However, other inferences seem so improbable that they can be accepted only with strong documentation. A related issue is the extent to which the original descriptions are able to be faithfully represented within the structured framework. Because we are open to human judgment in complex cases, we need to allow the ability to override system inferences.

We distinguish between qualitative and numerical models. We have focused of qualitative models because they are close to text descriptions and there are an increasing number of tools to support them. We believe the qualitative models can be extended to numerical and probabilistic representations later.

III. ARCHITECTURES

A. Ontologies

We start with ontologies for representation of semantics. In addition to providing an overall structure, the ontologies support logic-based inferences.

1) Upper Ontologies and the BFO

Upper ontologies cover a range of high-level structures and concepts. Several upper ontologies have been proposed such as BFO, SUMO, and DOLCE. The BFO (Basic Formal Ontology) because it is relatively well developed and because it is a "realist" ontology. We believe that basic histories are also largely "realist" and we have proposed applying BFO to full-text and rich media digital libraries [9, 14]. As a result of this approach, BFO describes the world with universals rather than with concepts. BFO proposes a hierarchy of entities which are divided into SNAP and SPAN [20]. These are 3D (Continuants) and 4D (Occurrents) approaches respectively.
SNAP universals are further divided as Dependent Continuants and Independent Continuants. SPAN Occurrents allow changes across time.

Below the top level, domain ontologies may be based on the typology of entities specified by the BFO. An especially wide range of ontologies based on BFO have been developed in biology.[3] Domain ontologies are themselves sometimes divided into several levels of detail of domain ontologies ranging from "upper domain ontologies" to more focused ontologies.

2) OWL

OWL, the Web Ontology Language, is the well-known formalism for specifying ontologies along with constraints and axioms associated with those ontologies. OWL is based on RDF which is composed of triples. While there are proposals for allowing n-ary relationships[4] those are not standard parts of OWL. In addition, OWL ontology-based development environments support a variety of inference engines. Typically, the inferences are based on properties of RDFS and OWL relationships such as whether they show transitivity. Given such properties, the inference engines can find relations that were not explicitly stated. There are several levels of OWL and for some of these no robust general purpose inference engine has been developed.

3) Relationships between Entities

Entities in ontologies are connected by relationships. Thus, the sets relationships are integral to the utility of the ontology although the close connection between the two is often not recognized. The Relationship Ontology (RO)[5] is closely, though not totally, coordinated with BFO. Relationships such as is_A, part_Of, and instance_Of are particularly noteworthy. Is_A is often associated with inheritance. Part_Of is associated with complex entities (see Section 4). As for instances, while ontologies are often considered to emphasize the universals associated with a domain, they can potentially be used to provide a knowledge structure for instances. Indeed, a distinction is commonly made between T-boxes and A-boxes. The "T" refers to "terminology" (i.e., statements with universals) in a domain while the "A" refers to "assertions" (i.e., statements or facts about instances).[6] In addition to specifying relationships between entities with triples, there are several ways to implement constraints and rules such as axioms and rule languages.

Sometimes SNAP and SPAN are treated as different ontologies and they are said to be linked by a distinct set of

[3] http://www.obofoundry.org/

[4] http://www.w3.org/TR/swbp-n-aryRelations/

[5] https://code.google.com/p/obo-relations/

[6] It is possible to have assertions about the relationship of universals but for A-boxes there the assertions are about instances. For an instance every attribute would have a

Trans-ontological and meta-ontological “signatures” rather than relationships [43]. However, more recently, SNAP and SPAN are considered as sub-ontologies rather than separate.

4) Time, Events, Processes, Verbs, and State Changes

The representation of time, processes, and events with ontology languages is a continuing controversy (e.g., [21, 28]). In BFO a Process is an Occurrent which is ongoing activity (e.g., running).[7] Processes often occur in chains. In biology an organism goes through developmental stages. Even running at different speeds, if explicitly acknowledged, may be considered a chain of distinct processes. For some processes which follow regular patterns, such as the beating of a heart, Process Profiles may be developed [44]. We also identify Procedures which are a chain of Processes and which implement a specified sequence of state changes. For instance, chemical reactions may be seen as Procedure. There can be instances of a chemical reaction and also abstraction of the reactions according to families of reactants.

For dynamic environments which are often described with text, the representation of actions with verbs is central issue. In some cases verbs are related to the relationships which link entities but those relationships are not verbs. For a mother to give birth implies the relationship MotherOf but the action is different from the relationship.

BFO also has Occurrent Processurals for initiating a Process and ending a Process. These might be used for describing state changes. However, more complex structures are needed to fully represent verbs because many verbs connect multiple entities. For instance, the FrameNet project uses frames to implements many concepts including state-changing verbs [7].[8] Perhaps structures such as named graphs and frames could be used to enhance OWL but adding such complex structures would necessarily undermine the virtue of the simplicity of triples.

While ontologies may describe state changes, they do not actually implement them. Our explicit modeling of state changes differentiates our work from others on representing human histories (e.g., [28, 33]). Thus, we discuss programming languages below (Section 3.B) when we focus on models of dynamic situations. OWL-related projects such as SPIN and OWL-S support state changes although those are for restricted applications.

5) Potential of the BFO for Modeling Complex Social Situations

We have been exploring the application of BFO for the description of complex dynamic environments such as communities [14]. BFO seems particularly appropriate because it supports both 3D and 4D models, BFO is widely used in biology, and is relatively well developed but it has not been applied previously in many areas beyond biology. While some entities in communities (e.g., families) are nuanced social objects, many other entities such as those which appear in newspapers are relatively mundane (at least to the extent that they are typically described in the newspapers). In addition, [14] explored the application of BFO, Processes, and Procedures in a description of the Roman Constitution as it is described in Gibbon’s “The Decline and Fall of the Roman Empire”.

B. Object-Oriented Programming Languages

While ontologies can readily describe individual events, models of complex environments based on such descriptions will be difficult to implement. It might be possible to create different copies of entities for every different state and then update pointers to them but there would be so many combinations that this does not seem practical on a large scale. Its implementation would likely end up as equivalent to a state-based program (cf., [35]). Remarkably, state-based programming languages, in particular, object-oriented programming languages have had relatively little impact on ontological descriptions. This may be because programming languages are often associated with simulation rather than description but, as discussed earlier, the boundary between description and modeling is fuzzy and, indeed, we seek to bridge them with descriptive models.

Several concepts are bundled together under the rubric of object-orientation (e.g., [15, 21]). Among the central concepts is the notion of message passing as a way objects communicate and the related notion the close association of methods with specific classes. Another important concept is abstraction in which common aspects of functionality are abstracted into higher-level objects. Ideally, these techniques should lead to better designed, more reusable, and more manageable code. They have also been adopted for Business Process Engineering (BPE) to describe and analyze the entities in well-structured organizations. The Unified Modeling Language (UML) is a set of interlocking sub-languages which describe the entities and processes underlying a complex environment. However, UML

specific value (e.g., "red", "green", or "unspecified"). For a universal the that same slot might set the domain of the possible values as "red", "green", or "unspecified" but there would not be any value for that attribute.

[7] This usage for the term Process can be confusing. In BFO, a Process is more like the present participle of a verb than an ongoing procedure or workflow.

[8] While we may explore the theory of frame semantics later, our immediate application uses the FrameNet corpus only as a convenient resource. The verb-based FrameNet frames provide a useful set of “methods” (in the object-oriented sense. Natural language verb hierarchies provide a type of inheritance. However not all verbs and not all verb senses are covered by the current FrameNet corpus. Moreover, with structured representation, we may want to develop methods that go beyond state-change verbs in natural language.

is not a true programming language; there is no way to execute a set of UML statements. Research specification languages such as TELOS [30] make an even more direct link to requirements.

There is a natural relationship between object-oriented approaches and well-articulated semantic structures such as those in ontologies [15]. Indeed, the Object-Management Group (OMG)[7] has developed a conceptual framework for objects in object-oriented models. Beyond design, object-oriented languages support dynamical modeling. While Java and C++ are the best known object-oriented languages, they have several features which have been implemented primarily to support efficient coding. JavaScript and Smalltalk are other well-known object-oriented languages. As described above, object-oriented approaches associate data and methods with specific classes. In addition, those data and methods can be inherited and specialized from more abstract classes (objects). Inheritance can be simply a data compression device but it often reflects a semantic class hierarchy. In order for the program to be executed, the classes must be instantiated. Object-oriented languages can also support part_Of relationships by instantiating the parts within the parent classes. Depending on the whether the language has early or late binding, the parts will have to be fully specified ahead of time or they may be specified only at run-time.

C. Application-Driven Representations

While ontologies and programming languages provide detailed frameworks from which we may build complex structures, some application have developed ad hoc representations. Examples of the latter include the representations used in digital games [38], in narrative [8, 16], in intelligent tutoring systems [18], and in agent-based models for business processes [41]. There are also similarities to many types of semantic linguistic processing systems such as those which implement question answering [26], summarization [30], and discourse [27].

IV. Complex Entities

Beyond the traditional view of entities in data modeling in which they are relatively simple and associated with a specific attributes, the models we consider are composed of many layers of sub-entities and all of these interact in complex ways and many evolve through time. Different types of representations may be applied to match the requirements of different applications. Indeed, purely qualitative models have been employed by the mental modeling community for describing the closed system of steam boiler (e.g., [23]) and that level of description may be adequate for some of our applications.

A. Structure of Complex Entities

A few types of complex entities have been considered by the BFO based on principles of "causal unity" [43]. These principles include internal physical forces and an engineered assembly of components. We believe there are several other important sources of unity such as interlocking functionalities. For object-oriented approaches, complex entities are roughly comparable to objects. We might regard the functioning of some of the elements as a type of "encapsulation". Part_Of relationships are central to both approaches. However, there are many nuances in the way such relationships are used. For instance, a proper_Part_Of says that an entity cannot be part of itself. And, a direct_Part_Of is an immediate part_Of the higher level. In addition, some proposals have been made to model layers of granularity [36]. This matches the common notions about layers in an organism such as its organs or cells.

Because complex entities are made up several sub-entities their attributes can be can be interdependent. If the entities physically connected, they will all change location at the same time. Or, if they are linked in pathway, they may all be active together but only when that pathway is active. Thus, the description of the complex attributes needs to be coordinated.[8]

Because a complex entity may have a great many subentities (e.g., molecules of water in a glass of water) we will not be able to model them all. Thus, we might focus on only one or two of them as examples and suggest that their behavior generalizes to all the others but the descriptive system needs discourse structures (Section 7.D) to handle that in a consistent way. In addition, when a complex entity has several layers we may only deal with entities in one or two layers while ignoring the entities at the lower layers. In some cases, the lower layers may be composed of qualitatively different types of entities. For instance, a car engine may be run by gasoline, diesel fuel, or electricity and we may not know or care which one it is. In addition, we may have expectations about the low-level details but not be overly surprised if turns out to be something different (see the discussion of expectations in Section 7.C).

B. Complex Entities as Systems

While systems may be static, we are most interested in systems which are dynamic – those which have processes and procedures which fulfill specific roles and functions (see [14]). Ultimately, as part of our understanding of complex entities, we need to incorporate insights from System Dynamics, General Systems Theory, and Complex Adaptive Systems (CAS) into our models.

As systems, complex entities may have internal dynamics. In some cases these dynamics are largely independent from interaction with other complex entities. For instance, a person breaths without any external stimulus and in most cases it does not affect our models. Other internal processes such as a person's hunger or the need to sleep may have more overt behavioral manifestations and may need to be explicitly modeled. While individuals can be viewed as complex entities, at another layer groups of people form communities and those communities are complex entities. Some aspects of

[7] http://www.omg.org/

[8] Some of the difficulties encountered with multiple inheritance and attribution of causation are due to a failure to differentiate among the effects on parts or functions of a complex entity.

communities are relatively routine and well structured. We believe these can be well modeled by ontologies (e.g., [14]) and in some cases by agent-based models (e.g., [41]). Other aspects such as the relationships between people in families are idiosyncratic and hard to model. Moreover, just as there are a variety of approaches to account for the behavior of individuals, there are a variety of theories to explain the social structures such as institutions. To add to the complexity, communities evolve as knowledge, technology, and culture change. Moreover, there are many social activities beyond communities such interaction in cities, in mass social groups, and as part of nations. Ultimately, our models may be extended to apply to all of them.

C. Instances of Complex Structures

The usual notion of instantiation can be difficult to apply for complex entities. We may say that a town exists (i.e., instantiate a high-level description) without knowing much about it. Thus, we need partial instantiations.

V. HYBRID IMPLEMENTATIONS

We seek to bridge the traditional distinctions between ontologies and programming languages (cf., [35]). In histories of complex scenarios, ontologies can have two levels of description. The descriptions available in T-Boxes can be part of broad knowledge structures and can have an effect across a large number of entities or methods as when a new technology is introduced. The A-Boxes could state a location or the roles played by an individual.

An object-oriented program applies classes to instances. Effectively, the entire program is the dynamic model of history. Here, we explore two approaches. The first is based on Jena[9] which is a Java-based library which allows the implementation and validation of sets of RDFS statements and OWL. The second architecture uses the Slate object-oriented language.

A. Jena Java

Jena allows the developer to add, delete, and modify RDFS and OWL statements in an ontology model. In addition, Jena provides a flexible framework for supporting inference engines reasoners for the validation of the models and querying the models. However, these tools have varying restrictions and capabilities. For instance, the low-level Jena RDFS Reasoner creates an Inference Model that supports indexing inheritance. However, the OWL Reasonser is needed to make inferences using even simple part_Of relationships. SPARQL queries are implemented with Jena ARQ.

Jena statements can be intermingled with Java statements, and we can use the Java to manually create and modify entire ontologies through Jena. We can also use methods implemented in Java to make the state change. This is related to the approach introduced by [6]. Both the manually created and the programmatically updated ontologies can be validated using the Jena inference tools. We envision a cyclic process of implementing the state change(s) to the 4D entities followed by validation of the resulting ontology.

Moreover, just as we extend by OWL in building ontologies, we can enhance or replace the inference engines. The Jena reasoners work by expanding all combinations of statements allowed by the relationships. A major problem is that this readily leads to a combinatoric explosion (too many combinations). We can control the scope of the validation and inferences to restricted set of statements (see Section 6.C). Similarly, tests of specific conditions and constraints could be implemented.[10]

B. Slate

The second hybrid model we consider is based on Slate which is a research prototype object-oriented language [40] which is descended from Smalltalk. Slate is known as a relatively pure implementation of object-oriented principles. Potentially, the strength of its approach to object-orientation will make it a better platform than Jena/Java. However, there is no RDFS/OWL library comparable to Jena. Indeed, while a basic set of programming tools has been implemented [37], the development environment is not very extensive.

Slate is an interpreted language with dynamic binding at runtime. Slate supports prototypes for inheritance and it allows multiple inheritance. Thus, Slate programs are extremely reconfigurable. In addition, Slate allows multiple dispatch which allows methods to be triggered according to the "roles" of entities. Multiple dispatch may be important for complex interactions such as for multi-agent interactions. This role-matching makes it somewhat like production systems which have been widely used in cognitive modeling. Another potentially useful feature of Slate is that macros may be applied across entities in complex objects [32] such as are common for structures models of complex environments (e.g., [1]).

Several features of the underlying object-oriented program are suitable interacting with structured models such as we explore in this paper. For instance, methods associated with an object can be readily updated. Thus, if a person acquires a new role and that role introduces new privileges and responsibilities, the methods associated with those can readily linked to the entity. A major drawback for Slate is that there is no RDFS or OWL support. A library comparable to Jena would have to be developed. Relationships would need to be introduced to link the classes and then validation and inferencing tools built.

VI. COGNITIVE AND LINGUSTIC PROCESSES

Human cognition language use has some effective, though not perfect, semantic models. We examine here some of the possibilities for employing some aspects of those representations in machine-based models bearing in mind the varying requirements for different tasks discussed at the beginning of the paper.

[9] https://jena.apache.org/

[10] Tests of pre-conditions and post-conditions such as those in the Eiffel programming language might be helpful.

While the BFO is a realist approach, many of the issues are related to human linguistic and cognitive representations. The structure of natural language is closely intertwined with human information processing (e.g., [24]). Moreover, accounts of mental processes are reported in texts and may need to be represented. Some of the insights of DOLCE might be used and, perhaps, blended with BFO's approach to incorporate cognitive representations [22].

A. Symbol Processing and Categorization

There is considerable evidence that human beings use distributed representations rather than processing traditional symbols such as we have emphasized here. While we believe that there is a sub-symbolic layer in human cognition and that may be useful to model to determine nuances such as word senses, it appears a great deal can be done with the consistencies in the symbol layer. A further debate concerns prototype approaches to categories and is discussed below (Section 7.C).

B. Analogies

Although analogy is seen as a cognitive process [17], when ontologies such as the BFO are coupled with abstraction and inheritance as models, they should be compatible with symbol-based accounts of the use of analogies.

C. Scoping, Episodes, and Assemblies

Presumably people do not consider all of their world knowledge when making judgments or checking plausibility. After all, a person's total world knowledge is enormous and considering all possible connections would result in a combinatoric explosion. The downside, however, is that people's knowledge may be siloed and that some unexpected remote connection actually has an effect.

Scoping and limiting options is well recognized in models of natural language understanding [20]. Similarly, events in stories are grouped into episodes [16]. There may also be fluid assembles of entities (e.g., people on a dance floor) which function as an ad hoc entity.

For interacting with the broad synthesized entity-event fabrics we have been considering, it will be helpful to implement some sort of scoping tags which control the level of processing of collections of related instances but limits broader processing.

D. Goals versus Subsumption

Much information processing in artificial intelligence is based on establishing goals attempting to follow them. We tend to believe that goal-based reasoning is often unnecessary and could be eliminated with Occam's razor in favor of subsumption and a behaviorist approach. Nonetheless a wide variety of goal-based models and some of those may be useful. For instance, [13] proposed the GOMS (Goals, Operators, Methods, and Selection Rules) model. This was developed to describe HCI tasks but it might be considered as an approach to specifying the activities which compose processes.

E. Modeling Natural Language

We can move from modeling specific components to a unified framework for describing dynamic environments with interacting complex entities. Taken together, research on advanced information structures, research programming languages, and linguistics research on frames and on discourse provide a useful foundation for modeling semantic interactions. Thus, natural language can be modeled with complex entities acted on by complex state-change methods. Moreover, the entire set of entities and state changes are shaped by human activities and cultural context.

VII. Explanations

Explanation attempts to answer the question as to why something happened [34]. While we have emphasized description over explanation, ultimately the two need to be closely coordinated. Robust explanation of everything in history is not feasible but we can consider the issues and apply them to narrower cases.

A. Causation

Explanation is closely associated with the notion of causation [12]. Effective explanation may be personalized for individuals and can be integral to tutoring systems (e.g., [18]). Causation is controversial notion in some circles. Yet, we believe that causation is integral to narrative [2] and scientific [10] explanations. In addition, the notion of causation is [not] well accepted by the BFO. One issue about causation is the difficulty of determining "the cause" from a set of causal factors. Picking a single cause often depends assumptions about what which was the most exceptional or unexpected factor and that can be very subjective. Causal factors can be linked into threads and may be useful for generating narratives plots and explanations along with other contextual information. Interwoven narrative threads may help to reinforce the fabric and to provide multiple constraints. Because causation implies a temporal sequence, perhaps the relative order is more important than absolute time stamps.

B. Theory of Mind

Some complex entities have especially complex internal states which are difficult to observe. To the extent that we attempt to explanations of the behavior of such complex we need to attribute mental states. In judging such internal states, we may apply a "theory of mind" [11] about cognitive processes. In some cases, this may be based on principles such as information processing with bounded rationality. In other cases, it may be more of a folk psychology. In still other cases, we may model what a person says about their mental state.

C. Prototypes, Expectations, and Constraints

While we do not know exactly how people represent concepts in the world there is considerable evidence that they develop prototypes (e.g., [39]) and their reasoning depends on expectations. While we may know that an entity is an automobile, we may have expectations about many details about it but not be sure. Similarly, we may know in general terms that an event occurred but not know many details about it. If we are explaining the reasoning and decision processes of agents, we may need to model their mental events and expectations. Cognitive prototypes and concepts are not directly supported by BFO's realist entities. Although they can be modeled indirectly with BFO, this is cumbersome. A related issue is the extent to which we use probabilities (modal logic) to generate explanations.

Explanations often explain why something happened but they may also try to explain why something did not happen. To the extent that we want to include such explanations into our models, we need to include constraints. Representing all possible constraints (i.e., context) in a general way is a daunting prospect but some limited constraints can be wired-in and used for limited inference.

D. Evidence for Claims and Assertions, Argumentation, and Extended Composite Hypertexts

There several types of assertions. Some are logical, some may be based on expectations, and some based on reports by observers. The latter would refer to entities and events which are themselves part of the fabric. Metadata are usually considered to be assertions about information resources but in a knowledge-base, each assertion can have its own metadata.

In previous work (e.g., [2, 6]) we discussed the need for discourse tags and for supporting those discourse tags with composite hypertexts. The broad range of applications (Section 2) along with the combination the complex structures we consider allows us to extend traditional approaches to composite hypertexts. Most previous work on composite hypertext systems have focused on simple composites such as guided tours or supporting argumentation with a small number of discourse tags. We envision families of broader combinations of composite models. For instance, we might introduce branching, personalization, and contextual annotations to guided tours. Similarly, we might develop extended argumentation systems.

Potentially, all assertions are falsifiable and need to be supported by evidence and can be disputed. For instance, generalizations are assertions about a set of instance entities or events. A generalization states that there is a commonality among the entities that is not part of the definition of those entities. Such assertions may need to be supported by structured argumentation which might explain away exceptions and challenges to the evidence.

Rhetorical structures such as the nucleus and satellites described in Rhetorical Structure Theory [27] may be considered as conceptual units (Section 6.C). Systems of XML tags have been developed for these but they have not been fully integrated into ontologies.

VIII. Discussion

We have emphasized entities which are complex, dynamic, and instantiated. These entities may interact with others, perhaps like agent-based models. In addition, the complex entities can be embedded as collections of complex entities (e.g., as part of a community or a society). These are far more complex – often multi-layered -than entities which are usually considered in data modeling. Indeed, there could be a single top-level "history-of-the-universe entity" which would be an Occurrent temporal region. Potentially, this would encompass all physical and social events. It is debatable whether it would also include claims or scientific laws (e.g., that the speed of light cannot be exceeded) and definitions. These may be seen absolute and existing outside of history but perhaps even they could be included. While we believe strongly that the speed of light cannot be exceeded, that remains an empirical conclusion.

A significant issue for modeling with the BFO is the large number of different entities which must be identified. Even scales such the temperature are considered qualitatively, each gradation is represented by a different entity. Similarly, each different chain of events may be a different entity. Indeed, ad hoc groups of interacting individuals may be thought of as distinct entities. Yet, the new ontologies may be only slightly different from previous versions so an ontology management framework might be used rather than developing separate ontologies. Because we allow many versions and views of the ontologies, and because they apply to dynamic situations, we need policies and tools for managing them.

Potentially this work could provide a common framework for the range of advanced applications described in the introduction. While we have focused on natural and social histories, the topics discussed in this paper may apply more broadly to semantic representations involving natural language.

References

[1] R.B. Allen, "Developing a Knowledge-base to Improve Interaction with Collections of Historical Newspapers", IFLA WLIC, 2011, San Juan, PR. http://boballen.info/RBA/PAPERS/IFLA2011/IFLA2011.pdf

[2] R.B. Allen, Visualization, Causation, and History, *iConference*, 2011, doi: 10.1145/1940761.1940835

[3] R.B. Allen, "Model-Oriented Scientific Research Reports", D-Lib, May 2011. doi: 10.1045/may2011-allen

[4] R.B. Allen, "Rich Linking in a Digital Library of Full-Text Scientific Research Reports". Research Data Symposium, 2013. http://boballen.info/RBA/PAPERS/RDS/RDStext.pdf

[5] R.B. Allen, "Model-Oriented Information Organization: Part 1, The Entity-Event Fabric", D-Lib Magazine, July 2013. doi:10.1045/july2013-allen-pt1

[6] R.B. Allen, "Model-Oriented Information Organization: Part 2, Discourse Relationships", D-Lib Magazine, July 2013 doi: 10.1045/july2013-allen-pt2

[7] R.B. Allen, "Frame-based Models of Communities and their History". Histoinformatics, 2013, Kyoto, LNCS 8359, 2014, pp. 110-119. doi: 10.1007/978-3-642-55285-4_9

[8] R.B. Allen and Acheson, J.A.," Browsing the Structure of Multimedia Stories", ACM Digital Libraries San Antonio, TX, June, 2000, 11-18. doi:10.1145/336597.336615

[9] R.B. Allen and Y.M. Chu, "Towards a Full-Text Historical Digital Library", ICADL, 2014, LNCS 8839, 2014, pp. 218-226. doi: 10.1007/978-3-319-12823-8_13.pdf

[10] R.B. Allen, Y.J. Wu, and J. Luo, "Interactive Causal Schematics for Qualitative Scientific Explanations", ICADL, 2005, LNCS 3815, pp. 411-415. doi: 10.1007/11599517_50

[11] S. Baron-Cohen, "Precursors to a Theory of Mind: Understanding Attention in Others". In A. Whiten (ed.), Natural Theories of Mind:

Evolution, Development and Simulation of Everyday Mindreading, Oxford: Basil Blackwell, 1991, pp. 233-251.

[12] J.K. Campbell, M., O'Rourke, and H.S. Silverstein (eds.), "Causation and Explanation", MIT Press, 2011.

[13] S. Card, T.P. Moran, A. and Newell, "The Psychology of Human Computer Interaction", .Erlbaum, 1983.

[14] Y.M. Chu, and R.B. Allen, "Structured Descriptions of Roles, Activities, and Processes in the Roman Constitution", IRCDL, 2015, in press.

[15] A.H. Eden, "A Theory of Object-Oriented Design". Information Systems Frontiers 4(4), 2002, pp. 379-391.

[16] D. Elson, "Modeling Narrative Discourse". Ph.D. Thesis. Department of Computer Science, Columbia University, New York City, 2012, http://www.cs.columbia.edu/~delson/pubs/Modeling-NarrativeDiscourse_Elson_R4_Condensed.pdf

[17] D. Gentner, "Structure-mapping: A Theoretical Framework for Analogy", Cognitive Science, 7(2), 1983, 155-170. http://csjarchive.cogsci.rpi.edu/1983v07/i02/p0155p0170/MAIN.PDF

[18] A.C. Graesser, K. VanLehn, C.P. Rose, P.W. Jordan, and D., Harter, "Intelligent Tutoring Systems with Conversational Dialogue". AI Magazine 22 (4) 2001, pp. 39-52, http://www.memphis.edu/graesser/publications/documents/BSC503_graesser.pdf

[19] P. Grenon, and B. Smith, "SNAP and SPAN: Towards Dynamic Spatial Ontology", Spatial Cognition and Computation, 4, 2004, pp. 69–103. http://ontology.buffalo.edu/smith/articles/SNAP_SPAN

[20] B.J. Grosz, and C.L. Sidner., "Attention, Intentions, and the Structure of Discorse". Journal of Computational Linguistics, 12(3) 1986, pp. 175204.

[21] C.A. Gunter, and J.C. Mitchell (eds.). "Theoretical Aspects of ObjectOriented Programming Types, Semantics, and Language Design", MIT Press, 1994.

[22] J. Hastings, W. Ceusters, M. Jensen, K. Mulligan, and B. Smith, Representing Mental Functioning: Ontologies for Mental Health and Disease. International Conference on Biomedical Ontology, 2012. http://kr-med.org/icbofois2012/mfo/

[23] J.D. Hollan, E.L. Hutchins, and L. Weitzman, "Project STEAMER: An Interactive Inspectable Simulation-Based Training System", AI Magazine, 1984, pp 15-27, http://aaaipress.org/ojs/index.php/aimagazine/ article/viewFile/434/370

[24] C.S.G. Khoo, and J.C. Na, "Semantic Relations in Information Science" Annual Review of Information Science and Technology (ARIST), 40(1), 2006, pp 157-2006. doi: 10.1002/aris.1440400112

[25] K.U. Krieger, A Detailed Comparison of Seven Approaches for the Annotation of Time-Dependent Factual Knowledge in RDF and OWL. Joint ACL-ISO Workshop on Interoperable Semantic Annotation, 2014, http://www.dfki.de/lt/publication_show.php?id=7333

[26] J. Lin, and D. Demmner-Fushman, "Methods for Automatically Evaluating Answers to Complex Questions", Information Retrieval, 9(5), 2006, pp. 565–587.

[27] W.C. Mann, and S.A. Thompson, "Rhetorical Structure Theory: Toward a Functional Theory of Text Organization", Text, 8(3), 1988, pp.243281.

[28] A. Meroño-Peñuela, A. Ashkpour, M. van Erp, K. Mandemakers, L. Breure, A. Scharnhorst, S. Schlobach, and F., van Harmelen, "Semantic Technologies for Historical Research: A Survey", Semantic Web Journal, 2014, http://www.semantic-web- journal.net/system/files /swj588_0.pdf

[29] J.C. Mungall., A critique of temporalized relations, 2013, pp. 1-10. https://github.com/cmungall/trel-crit/raw/master/trc.pdf

[30] J. Mylopolous, A. Borgida, M. Jarke, and M.. Koubarakis, "TELOS. Representing Knowledge about Information System", ACM TOIS, 8(4), 1990, pp. 325-362. doi: 10.1145/102675.102676

[31] A. Nenkova, and R. Passonneau, "Evaluating Content Selection in Summarization: The Pyramid Method", NAACL-HLT, 2004. Pp. 145152.

[32] J. Nobel, and B. Foote, 2002, "Attack of the Clones", Asian Pacific Conference on Pattern Languages of Programs. Conferences in Research and Practice in Information Technology, 13. http://crpit.com/confpapers/CRPITV13Noble.pdf

[33] D. Oldman, M. Doerr, G. de Jong, B. Norton, and T. Wikman, "Realizing Lessons of the Last 20 Years: A Manifesto for Data Provisioning & Aggregation Services for the Digital Humanities (A Position Paper)", D-Lib, July 2014, doi: 10.1045/july2014-oldman

[34] J.C. Pitt, Theories of Explanation, Oxford, 1988.

[35] C. Puleston, B., Parsia, J. Cunningham, and A. Rector, "Integrating Object-Oriented and Ontological Representations: A Case Study in Java and OWL", Semantic Web Conference, 2008, pp 130-145, doi: 10.1007/978-3-540-88564-1_9

[36] A. Rector, J. Rogers, and T. Bittner, "Granularity, Scale and Collectivity: When Size Does and Does Not Matter". Journal of Biomedical Informatics, 39(3), 2006, pp. 333-349. doi:10.1016/j.jbi.2005.08.010

[37] B.T. Rice, and L. Salzman, "The Slate Programmer's Reference Manual". 2004. http://slatelanguage.org/documentation/

[38] M.O. Riedl, and V. Bulitko, "Interactive Narrative: An Intelligent Systems Approach", AI Magazine, 34(1), 2013, pp. 67-77. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2449

[39] [E. Rosch, "Cognition and Categorization", Erlbaum, 1978.

[40] L. Salzman, and J. Aldrich, "Prototypes with Multiple Dispatch: An Expressive and Dynamic Object Model", ECOOP, LNCS, 3586, Springer, 2005, pp. 312–336, doi: 10.1007/11531142_14

[41] M. Serhius, W.J. Clancey, and R.J.J. VahHoof, "BRAHMS: A Multiagent Modeling Environment for Simulating Work Processes and Practices". International. Journal of Simulation and Process Modelling, 2007. http://www.dagstuhl.de/Materials/Files/07/07122/07122.

[42] B., Smith, "Beyond Concepts: Ontology as Reality Representation". International Conference on Formal Ontology and Information Systems (FOIS), Turin, 2004. http://ontology.buffalo.edu/bfo/BeyondConcepts.pdf

[43] B. Smith, "On Classifying Material Entities in Basic Formal Ontology", Interdisciplinary Ontology, 2012, pp. 1-13, http://ontology.buffalo.edu/smith/articles/Material_Entities.pdf

[44] B. Smith, "Classifying Processes: An Essay in Applied Ontology", Ratio, 25(4), 2012, pp. 463-488. doi: 10.1111/j.1467-9329.2012.00557.x

[45] B. Smith, and P. Grenon, "The Cornucopia of Formal-Ontological Relations", Dialectica 58(3), 2004, pp. 279-296, http://ontology.buffalo.edu/smith/articles/cornucopia.pdf