Ontology-based fusion of sensor data and natural language

Erik Thomsen [a,*], Barry Smith [b]
[a] *BlenderLogic, Cambridge, MA*
[b] *University at Buffalo, Buffalo, NY*

Abstract. We describe a prototype ontology-driven information system (ODIS) that exploits what we call Portion of Reality (POR) representations. The system takes both sensor data and natural language text as inputs and composes on this basis logically structured POR assertions. The goal of our prototype is to represent both natural language and sensor data within a single framework that, in addition to axiomatic reasoning, directly supports computations and is capable of discovering and representing new kinds of situations and thematic roles, (e.g., roles such as agent, patient and instrument), based on new compositions of existing representations. We applied our prototype in an intelligence analysis use case to test the hypothesis that a framework of this sort can produce usefully structured information from combined natural language and sensor data inputs. We further tested our hypothesis by adding an enhanced US Air Force ontology framework to our ODIS in order to (1) process a collection of sensor data, intel reports, and mission plans; (2) build composite POR representations from these data; and (3) machine analyze the fused results to infer mission threats.

Keywords: Ontology-driven information system (ODIS), data fusion, sensor data, natural language understanding, intelligence analysis, Basic Formal Ontology (BFO)

1. Introduction

To address information-intensive tasks such as the dynamic re-planning of a military mission or the on-going assessment of threats to the operations of a large corporation we need to be able to (1) make sense of incoming data, whether in the form of sensor feeds, database records, or natural language texts; (2) reconcile these data with stored knowledge; and (3) respond accordingly.

Ontologies are increasingly leveraged for all of these tasks. Current systems typically match text elements (for instance, nouns and verbs) with ontology terms and relations. Increasingly, such systems involve the use of suites of ontology modules linked together through a top-level ontology such as BFO (Arp, Smith, and Spear, 2015), DOLCE (Gangemi, et al., 2002), or SUMO (Niles and Pease, 2001). Here, reflecting requirements flowing from the Air Force ontology framework within which this work is being carried out, we focus on ontologies conformant to Basic Formal Ontology.

Ontology-Driven Information System (ODIS) applications of relevance here start by mapping strings as they appear in texts to terms as they appear in ontologies. The system then processes the results of such mappings in actions such as routing alerts to appropriate analysts.

---

[*] Corresponding author. E-mail: ethomsen@blenderlogic.com.

By drawing on the contents of ontologies built for this purpose, such systems produce relatively simple descriptions of entities by interpreting simple sentences or sentence fragments extracted from more complex text (Schoening et al., 2015). Thus far, however, there are no ontology-based information systems that can produce the sorts of descriptions that would represent the world as interpreted by humans on the basis of complex sentences. And, *a fortiori*, there are no such systems that are able to reason over combinations of text inputs with the sorts of quantitative data provided by sensors.

It was in the spirit of extending BFO-conformant ontologies beyond the single term focus in order to enable treatment of whole sentences that the term 'portion of reality (POR)' was coined.[1] As defined by Ceusters and Smith (2015), 'POR' denotes anything that exists, whether simple or complex, tangible or intangible, universal or particular; including more or less complex combinations of all of these. This includes, most importantly for our purposes here, the sorts of combinations – sometimes called 'facts' or 'states of affairs' – to which simple and complex true empirical sentences correspond, including sentences expressing quantitative data. Representations, in whatever form they take, are at best descriptions of certain parts or aspects of a POR.

For the building of an ODIS whose goal is to be able to compose (and then reason with) these more complex representations of PORs, we wish to use the top level of a BFO-conformant ontology (i.e., the level comprising the children of the root term 'entity' or 'thing'[2]) to supply a collection of ontological dimensions that, for the ODIS at least, are common to all POR representations. Doing so allows us to take a combinatoric approach to building an ODIS that relies solely on iteration, recursion and specialization operators for the composition of new POR representations from those that already exist.

1.1. Motivating goal

Our goal is to extend BFO conformant ontologies to the portions of reality represented by natural language sentences and other more complex expressions. To illustrate the challenges of composing these more complex representations, consider the following two sentences, both of which are composed almost exclusively of BFO terms (*italicized*) and relational expressions.

1. Some particular *independent continuant* that is *bearer of* some *specifically dependent continuant* is *bearer of* a particular *role* in an *occurrent* whose terminal *process boundary occurred at* a particular *temporal instant*.
2. Some particular *role* with some *zero-dimensional continuant fiat boundary* is *realized in* a certain *relational quality* in a *one-dimensional temporal region*.

---

[1] Ceusters and Smith (2015) use the machinery of PORs to solve the problem of how two sentences, for example "Trump is President of the United States" and "Trump is President of Russia" can differ in their truth value even though both are made of constituent elements that refer to entities in reality. See also Smith, Kusnierczyk, Schober, and Ceusters (2006) and Ceusters and Manzoor (2010).

[2] In Basic Formal Ontology this level comprises Continuant and Occurrent. In the BFO-conformant ontology we used for our project, there are four dimensions covering time, space, object and process. Our model ontology specializes and allows recombinations of these four dimensions in order to facilitate the machine composition of POR representations. The ontology and its grounding in computable representations that allows for computing with POR representations is described later in this paper.

Sentence 1. is a valid composition of BFO terms because, with an appropriate substitution of specific terms for BFO classes (for example, 'car' for 'independent continuant'), the sentence could represent a POR (for example, a red car participating in a race). The same cannot be said for sentence 2. There is no substitution of specific terms for BFO classes that would make this sentence represent any kind of POR. This is because BFO roles are not realized by qualities.

The specification of the ways in which different ontological terms can be combined to form representations of PORs (e.g., a process boundary must occupy a temporal instant, a role cannot be realized by a relational quality) may be called an ontological grammar. Grammars can be more or less complex depending on the regularity of the rules by which representations of ontology classes may be combined. The fewer the rules and the fewer the classes, the simpler the grammar.

In this paper, we describe an ontological grammar we developed for use by a machine to combine representations of entities of all kinds into meaningful structured assertions about PORs. The goal is to interpret text and sensor data in a way that will, for example, allow immediate comparison of incoming messages with assertions already interpreted at earlier stages some of which may have come directly from sensors.

As part of that effort, we created a compact BFO-conformant ontology for the purpose of facilitating the machine composition of assertions about PORs relevant to our application domain. Our model ontology has four principal ontological classes: Time, Space, Object, and Process, each of which comprehends more specialized ontology subclasses at multiple levels of granularity and generality.[3] Important among these are units of measure classes such as Day or Kilometer, which are subclasses of Time and Space, respectively. The principal ontological classes may not only be subclassed but also combined into composite classes, which may be subclassed in their turn. For example, Speed is a sub-class of a composite of Space and Time. Attribute classes such as Mass, Color, and Texture are likewise built into the ontology as subclasses of composites of the four principal classes.

In addition to the four principal classes in the model ontology we also assert a fifth class comprehending what we call 'computing dimensions'. This comprises as subclasses mathematically computable dimensions such as Boolean, Categorical, Integer, Rational, Array, and Graph. Each simple and composite subclass under the principal ontological classes is then implemented by means of some computing dimension or combination thereof – for example Time may be implemented as a Rational, Speed as the ratio of two integers. An ontological Date-Time dimension that includes a (1) year, (2) day-of-year, (3) hour-of-day and (4) minute-of-hour may be represented for computing purposes by combining four integers. Thus the computational properties (or semantics) for all ontological classes (and their subclasses and composites) are determined by the computing dimensions in terms of which they are implemented.

The PORs within the scope of our application are represented by assertions formed by some combination of terms corresponding to each of the principal classes of the model ontology. In other words, every POR is assumed to have some temporal aspect, some spatial aspect, some object aspect, and some process aspect.

Our system recognizes assertions about the relations between classes, called 'POR class assertions'. Examples are: 'jet fighter is_a plane' and 'jet fighter dodging surface-to-air fire is

---

[3] A mapping from our model ontology to BFO may be found at this URL: https://bit.ly/2Ecv2Cc

a plane evading anti-aircraft fire'. In addition, the system recognizes what we call 'POR instance assertions', which are instance-level counterparts of POR class assertions.[4] Consider:

1. the maximum strike distance of Argonaut Class surface-to-air missiles is 5 vertical miles
2. a fighter jet was shot down yesterday at an altitude of 8 miles by an Argonaut Class surface-to-air missile

Here the information represented in POR class assertion 1. conflicts with observational information expressed as POR instance assertion 2. When our system encounters such a conflict it is forced to attempt some reconciliation. This may involve discovering information, for example, to the effect that the plane was flying lower than stated, or that the surface-to-air missile was of a different class, or that the maximum range of the class *X* surface-to-air missiles was, in fact, greater than had been assumed, and so on. As one of our goals is to recognize when current reality diverges from what prior knowledge deemed most likely, a minimal requirement for our system is that it is able to compare POR class assertions with POR instance assertions.

Given that each ontological class is associated with one or more appropriate computing dimension, the associated operators and compositional rules can be applied also to the ontological classes. Thus, for example, the ability to specify temporal relations such as *before* and *after* between temporal instances of a temporal unit subclass of the Time class is inherited from the ability to specify numeric relations between the instances of a computing dimension such as integer. The grounding of all ontological classes and expressions in terms of computing dimensions and expressions also allows compositions of ontological terms into POR assertions to be carried out by the computer without further effort.

1.2. Motivating scenario

Air Force mission planning occurs on a recurring cycle known as the Air Tasking Order (ATO) cycle (US Air Force, 2016).[5] The information systems that support the ATO cycle allow periodic bulk updates to reflect new information coming into the system from the outside. Among the many tasks required to maintain the relevance and viability of the overall mission plan is the ongoing review of intel reports to identify information that could impact mission success. Ideally, intel reports would be interpreted and reasoned over in near real time, with inference results implying a threat to a planned mission being sent to the appropriate mission planners for immediate inclusion in updated mission plans. Currently, however, there are simply too many generated reports to allow rapid processing by human analysts. For this reason, the Air Force has been exploring the value of using an ODIS to speed up information extraction, interpretation, and reasoning processes.

The functioning of such a system involves the use of standing searches such as:

Select all sentences in intel reports containing references to

---

[4] The difference between POR class and instance assertions is not always relevant. This is the case, for example, in descriptions of ontological grammar, as such descriptions apply equally across class and instance assertions; in such cases we use the more general term 'POR assertion'.

[5] A link to the ATO cycle can be found at http://www.dtic.mil/dtic/tr/fulltext/u2/a451239.pdf

- instances of continuant classes C_1, …, C_n standing in the *located_in* relation to one or more instances of geospatial location classes referenced in the current mission plan
- instances of occurrent classes O_1, …, O_n standing in *has_participant* relation to one or more instances of continuant classes identified in search 1

where 'continuant classes' refers for example to *persons*, *vehicles*, and aggregates thereof, and 'occurrent classes' to *movements* of troops, weapons, and so forth.

To illustrate the level of reasoning that is supported by such ontology-driven term extraction, consider the following simple text fragments, where 'IR' denotes an Intel Report and 'MP' denotes a Mission Planning report:

- IR 1: Seven anti-aircraft missiles seen yesterday transported towards Raqqa
- IR 2: Though verified not operational, SAM launchers seen near *de facto* ISIS capital
- IR 3: Having left Airbase Bravo at 0500, special forces last seen approaching Raqqa *en route* to disabling Raqqa SAM launchers
- IR 4: Hyperspectral analysis shows ISIS fighters hiding in tunnels below the streets and in water tanks on the roofs
- IR 5: Village sentiment no longer on the rise as militants infiltrate tribal leadership
- MP 1 Start time 0500. Send special forces to disable SAMs near Raqqa
- MP 2 Start time 1600. Send squadron to attack ISIS positions near Raqqa

The goal of analysis is then to extract from the IRs all information from which it could be inferred that the success of the missions described in the MPs is at risk.

For MP 1 and 2 the following standing searches might be used by an ODIS to extract the sorts of key information items that need to be routed to appropriate analysts:

Search 1: Select all reports that contain
- Raqqa AND (SAM OR anti-aircraft missile)
- *and* route them to air operations planners

Search 2: Select all reports that contain
- ISIS fighters OR fighting OR resistance
- *and* route them to ground operations planners

Use of such search strategies brings immediate benefits. For example, it provides an easily extendible set of terms, enhanced through synonymy, parent-child and other relations, which can be used for the consistent indexing of very large bodies of IR content in a way that allows rapid discovery of salient data (Smith et al., 2012). By allowing the proper selection of search terms and their Boolean combinations, it can support filtering out of what are likely to be irrelevant reports (for example, search 1 ignores IRs 4 and 5) and thereby present human analysts with smaller piles of potentially relevant reports to sift through.

1.3. Shortfalls of existing ontology-based approaches

Current ODIS software leverages ontologies to identify relevant entities and relations and distribute associated intel reports to analysts. Unfortunately, existing ontology-driven approaches have a difficult time moving beyond such entity- and relation-based processing to support the sorts of inferencing required to alert mission planners about the need to re-plan missions in real time based on the kind of criteria that are in fact used by human analysts. These criteria work not just at the level of single entities, and single search terms, but also at the higher levels of PORs. A human analyst, for example, would recognize that IR 2 *is_about* the same portion of reality as IR 1, a portion of reality involving Raqqa and its relation to ISIS

in virtue of its role as the *de facto* ISIS capital. More generally, human analysts know how to go back and forth between names of the instances existing in their respective domains of expertise and descriptions of these instances linking them to background information.

A human analyst would recognize that IR 2 provides more detailed and more credible information than IR 1 about some POR – in this case about the status of certain observable SAMs. (IR 1 only asserts the presence of SAMs; IR 2 asserts their operational status.) For this capability to be replicated in the ODIS, differences in level of detail (and thus in relative believability) of assertions about a given POR need to be accounted for. Human analysts would also recognize that if certain PORs are logged – for example, that (a) the SAMS near Raqqa are not operational, or (b) a makeshift hospital is near the target – then certain other PORs follow: for example, (a) that there is no need to disable the SAMS and therefore no need to continue with MP 1, or (b) that the mission needs to be re-assessed to avoid unintended civilian casualties. Human analysts would need further to be able to generate warnings where there is a temporal dependency between PORs that is relevant to mission success, for example to infer that a mission to send a squadron to attack ISIS positions near Raqqa (MP 2) should be realized only when (but as soon as possible after) the SAM threat is neutralized (MP 1).

To replicate such reasoning, ontologies would need to be held in the ODIS representing for example the structures of plans and the different ways in which plans may be modified or aborted. The ODIS would also need to be able to extract not just terms and relational expressions from report data but also compose these into POR instance assertions referring to complex PORs, assertions that combine sensor data and the contents of planning documents with times, locations, phases of mission execution, and observed outcomes on the side of reality.

Imagine, therefore, a POR instance assertion representing the portion of reality in Raqqa on a specific day D that unfolds according to IR 1–4 as follows:

p_1 *instance_of* observation process
AND p_1 *has_target* transport process p_2
AND p_1 *results_in* p_3
AND p_3 *instance_of* report-issuing process
AND p_3 *has_output* IR 1
AND p_2 *instance_of* transport process
AND p_2 *has_direction* towards Raqqa
AND p_2 *has_initial_temporal_boundary temporal_part_of* D − 1
AND p_4 *instance_of* observation process
AND p_4 *verifies* p_1
AND p_4 *has_target* missile launchers o_1 *in* Raqqa
AND p_4 *has_content* o_1 are non-operational
AND p_5 *instance_of* hyperspectral observation process
AND p_5 *has_target* (ISIS fighters *with_location* below ground)
AND p_5 *has_target* (ISIS fighters *with_location* on top of buildings)

In our treatment of our motivating scenario, we are accumulating data in the form of POR instance assertions such as these, relating to the sorts of cases where discovery of changes in the zone of operations will imply the need to modify or abort a planned mission because risk to mission or collateral damage is unacceptable. We are designing our ODIS in such a way that it can use such data to draw inferences in the form of warnings of potential mission threats that can be useful to planners and commanders.

2. Related work

2.1. Early work on situations and events

There are many significant prior attempts to represent information, especially from textual sources, in ways that allow the drawing of inferences of the mentioned sorts. One of the earliest such attempts was the situation calculus (McCarthy, 1969), which used 'event' as its central primitive, together with a universal meta-predicate 'holds' signifying that an event occurred at a given time. In this framework, all events have antecedents and consequences.

The approach suffered from a major problem, however, turning on its lack of a mechanism for inferring the continuity of properties. The event calculus (Kowalski and Sergot, 1986) was created, at least partially, to provide a solution to this problem. Kowalski (1986) formalizes the effects of events on objects. This approach distinguished between *events*, which have actors and involve actions, and *states*, which are assumed to hold unless a terminating event takes place. Unfortunately, the assumption that there is stasis simply because there is no explicit representation in the system of a property-changing event itself proved problematic—a case of the now well-recognized frame problem.

The frame problem arises with the question of how to represent, within a logical system, the relevant effects of actions, given the large number of relevant but not obvious potential effects. Today, this problem is no longer considered an insurmountable obstacle to progress in knowledge representation, and various strategies have been presented for addressing it within both open and closed world frameworks.

One prominent example is Reiter (1991), building on work by Pednault (1989) and Schubert (1990). This utilizes the idea of *fluents* as properties that can change over time, as for example the property *operational*, and it requires a completeness assumption for the set of fluents so that it can be determined for each action whether or not it will lead to a specified successor state. Finzi, Pirri and Reiter (2000) demonstrated that this approach could be used to construct a forward reasoning planner for systems under the open world assumption. However, there are substantial efficiency concerns with this approach, and it requires positing a domain-dependent predicate used to filter out ineffective plans.

More recently, Baclawski et al. (2017) develop an approach to situation analysis in an ontology-driven system that includes both a theoretical framework and declarative language support for decision-making processes. The top classes in their ontology are intended to support decision-making within the OODA (Orient, Observe, Decide, Act) loop framework. These classes include: fact, hypothesis, perception, and directive; and the authors provide case examples demonstrating how they may be used to track steps used in decision-making in different domains. However, some of their choices seem questionable (such as requiring that all instances of a class of activity invoke only one instance of a reasoning process, as well as recognizing two distinct top nodes in their ontology). The authors also did not address the idea of incorporating into their framework a top-level ontology that would enable integration with other ontology resources.

Also relevant to our project is earlier foundational work by Schank (1973), Fillmore (1968), and others on frames and slots. Schank (1973) focused not on some general form for a POR representation, but rather on specific composable primitives. He introduced the notion of 'conceptual cases', conceived by analogy with grammatical cases (also called *case roles*). The idea is that each action type comes with a set of associated dimensional slots where each slot denotes a case role, so that when an action of this type is represented in an assertion then each

of these slots must have some value regardless of whether or not this is made explicit in the assertion.

Unfortunately, Schank's work on conceptual cases is not grounded in anything like an ontology of process and time on the side of reality. Rather, it remains always at the level of linguistic representations. As his starting point, outlined in his original seminal paper, he takes a list of fourteen primitives, which were intended to be composed into what he called 'conceptual structures', roughly corresponding to what we are here calling 'POR representations'. On this basis, he showed how many seemingly simple events, such as *buying a car*, involve multiple correlated sub-events (such as *seller exchanging a car for money*, *buyer exchanging money for a car*, *seller wanting a car*, and so on). Schank's framework also recognizes the distinction between mere changes in location and those exchanges that involve a transfer of ownership or control. But his system was restricted by its starting point in grammar, or more precisely, in the grammatical subject-verb-object structure. This made it hard for him to represent PORs that are more complex, for example transformations (as when a baker converts flour, sugar, and butter into cookies).

Schank's treatment of causality, too, which rests on a contrast between physical action and what he called a physical translation, has problems. A physical action such as pushing a bookcase is asserted to *cause* the physical translation that is a change in location of the bookcase. However, it is not the case that there is here a first process – the action of pushing – which then *causes* a second process – the movement of the bookcase towards the wall. The two processes are one and the same.

2.2. Work on Fundamental kinds of events

A valuable contribution to grounding the aforementioned approaches in an ontology of process and time is provided in the work of Galton (2006; 2012) and Borgo and Mizoguchi (2014), who have tried to incorporate what is of value in the situation and event calculi within a broader framework resting on a distinction between events (which are heterogeneous and have beginning and endings) and processes (which, in their idiolect, can in principle endure without end and whose temporal parts have the same structure as the whole).

Galton (2006) offers on this basis a representation of causal relations. For example, in the act of moving a wheelbarrow he identifies four events: *starting to push*, *starting to move*, *stopping to push*, *stopping to move*, alongside two (simultaneous) processes: *pushing* and *moving*, where the latter are seen as being joined together by a causal relation operating, somehow, a-temporally.

We are to understand that hammering is an *event,* where pushing is a *process*, because the temporal parts of hammering are not homogeneous whereas the temporal parts of pushing are. Yet an act of pushing a wheelbarrow may very well include portions of pulling because the wheel gets stuck, or of twisting because the ground slopes, and so forth. One might counter that pushing can in theory, at least, be imagined to be homogeneous whereas hammering cannot. But even then, pushing, if examined carefully, involves a feedback loop where the agent doing the pushing assesses any changes in resistance and thereby changes the amount of force required to push.

More generally, it is far from clear that pushing and moving (or any of the other events or processes discussed by Galton, *et al.*) really lack beginnings and endings. At the very least, they are bounded in time by the existential constraints on whatever or whomever is doing the pushing.

Galton's atemporal Schank-like analysis of causality, too, seems to us to be unintuitive. Pushing a wheelbarrow involves exerting force, gripping, and walking. If pushing fails, moving does not occur. But moving is not *caused by* pushing; rather, moving *is* pushing under certain conditions, as we see from the following more careful analysis of the processes involved (where *process* is to be understand, now, in BFO terms as any occurrent entity extended in time that involves material participants):

t_0 John intends to push and the wheelbarrow is stationary

t_1 John begins to attempt to execute a pushing process against the wheelbarrow; the electrical impulses have not yet contracted his triceps

t_2 John's exertional force exceeds the wheelbarrow's stationery force and the wheelbarrow begins to move; its motion is a part of a successful pushing process

t_3 John's exertional force continues to exceed the force required to move the wheelbarrow and the wheelbarrow continues to move

t_4 John intends to stop pushing the wheelbarrow

t_5 John's exertional forces drop below the level of the wheelbarrow's stationary forces and the wheelbarrow ceases to move.

The fact that John attempted to execute a pushing process against the wheelbarrow at t_1, combined with the fact that the force he exerted in pushing exceeded the inertial forces of the wheelbarrow at $t_2 - t_4$, is, we believe, a more accurate causal assay of a movement of this sort.

Assumptions of homogeneity and of succession of causes and effects may, it is true, be deeply rooted in our natural language. But an acceptable ontology should not conflate linguistic conventions with ontological representations of the underlying PORs.

2.3. FrameNet and related work

Since at least Whorf (Carroll, 1956) and more importantly Fillmore (1968), it has been understood that the linguistic roles (of subject, direct object, and so on) are independent of the thematic roles of an object in an event (for example, that some object is agent or patient). Consider 'Billy hit the tree' and 'The tree was hit by Billy'. Although linguistic roles flip between the two sentences (the tree moves from being object to being subject), the thematic roles are constant (Billy remains agent in both cases). Yet for the past fifty years, efforts to formalize linguistic frames for the purposes of information processing have incorporated some variant of thematic roles (e.g., agent, patient, instrument, theme) without ever integrating the latter into a more general formal ontological framework that integrates other POR information (for example pertaining to time and space). In other words, the treatment of thematic roles remains unconnected to an ontological framework governing the rest of the POR representation. Fillmore acknowledged as much in his seminal paper (Fillmore, 1968).

The problem is clearly illustrated in FrameNet (Baker, Fillmore, and Lowe, 1998), which has its roots in Fillmore's paper (1968), and which is perhaps the most widely used software resource for representing machine extracted meaning from sentences. Using the terms introduced above, FrameNet can be understood as a framework that provides targets for natural language processing by enabling multiple sentences, expressed using different linguistic conventions, to map to a single POR representation.

FrameNet is a verb-centric approach. Each of its over 1000 semantically related complexes called 'frames' is based on a verb or verb class and is comprised of a small number of core and optional thematic roles for nouns. The frames within this collection are then related to each other through a multi-hierarchy descending from a single root frame called 'event'.

FrameNet is built to support semantic extraction processes that associate nouns and noun phrases in sentences with their thematic roles in such events. The event itself is understood as being represented by the sentence as a whole, and thus corresponds to what we are here calling a POR. FrameNet thereby opens the door to statistical approaches to thematic role labeling, also called semantic role labeling, as articulated for example, in Palmer, Gildea, and Xue (2010). One of the first such approaches aligned the results of a Collins parser (Collins, 1999) with FrameNet roles and built a probabilistic model to identify the start and end words in a contiguous sequence of words that collectively mapped to a FrameNet (i.e. a verb phrase) role (Gildea and Jurafsky, 2002).

This statistical work then motivated numerous POR representation projects, including the Simple Event Model (Van Hage and Ceolin, 2013), the Penn TreeBank (Marcus, Marcinkiewicz, and Santorini, 1993), PropBank (Palmer, Kingsbury, and Gildea, 2005), News Reader (Agerri et al., 2014), AMR (Banarescu et al., 2013), ESO (Segers et al., 2015), and more (reviewed in Kalita, 2016). Unfortunately, none of these projects provides an ontological grounding for thematic roles. The Simple Event Model, for example, used also by News Reader, simply assumes static roles for *actor*, *time*, and *place within an event*.

3.   A compositional approach to POR representations

3.1. Overview

The goal of our ODIS is to use compositions of our ontological dimensions to represent all incoming data (e.g., natural language and sensor data), all calculations made with that data (e.g., mapping text into a canonical form, disambiguating text, entity resolution, numeric analyses), any and all background or newly discovered knowledge about the PORs that are the source of the data (e.g., typical range of a particular class of surface-to-air missile) so-called meta information (e.g., how recent the data is, how precise it is, the source from which it came, etc..), and gaps between prior knowledge and current POR representations (e.g., missing thematic roles).

From a software architecture perspective there are four major components to our ODIS as shown in Figure 1:

   (1) The assimilator is responsible for taking new data (e.g., sensor data or natural language), extracting lexemes, associating those lexemes with their possible meanings (expressed in types and schemas), selecting an actual meaning in the context of surrounding lexemes, merging extracted expressions into larger schemas that are then merged into the POR representation manager.
   (2) The POR representation manager is responsible for storing any new POR representations as well as performing all calculations on those POR representations. It is also responsible for reconciling prior knowledge with new information.
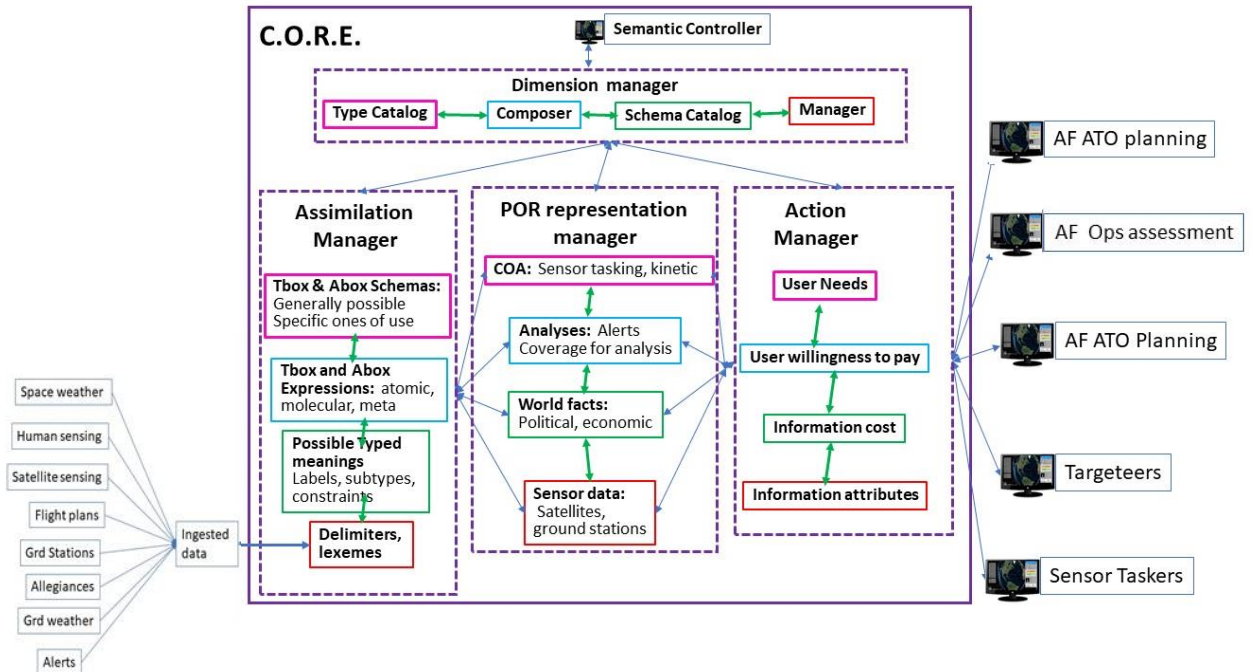
Figure 1. Overview of our ODIS.

(3) The responder produces data that can be consumed by persons or machines.
(4) The dimension manager is responsible for all modifications to any of the three other components (e.g., adding a new term to an ontological dimension, modifying an axiom), and to schemas composed of ontological dimensions.

We here attempt to take the kind of logical rigor currently found in work with axiomatized ontologies such as BFO and apply it to the messier world of data and computations where classifications may be probabilistic, where data sources may disagree, where knowledge has gaps and where data attained at a later point in time may disagree with beliefs acquired at earlier times with no clear method for determining how such differences are to be resolved.

Our ODIS was built to master complex POR representations under these conditions. It works on the basis of the idea of structured POR assertions whose constituent terms belong to combinations of four primitive ontological classes forming the four primitive dimensions[6] of Time, Space, Object, and Process. In addition to these four, we will often refer to the non-

---

[6] Although it is common to see the major branches of an upper ontology referred to as classes (and subclasses etc.), the term 'class' does not adequately capture the full range of computational capabilities we are trying to develop in our ODIS. This is because it is most often associated with Categorical or Set operations such as negation, union and disjunct rather than numeric or graph operations. In software engineering environments, 'type' (as in: Floating point types, Boolean types, Integer types, Array types, and so on) is commonly used. At the semantic, world-referencing level – for example as found in databases providing repositories of information about entities in the world – the most common terms are 'dimension' and 'attribute'. We use here the term 'dimension' because of its neutrality between the computing and world-referencing aspects of our framework.

primitive dimensions of Attribute[7] and Process. Each ontological dimension comprehends a collection of different qualitative (e.g., textual) or quantitative (e.g., numeric) representations. Time, for example can be captured by variables ranging over expressions such as 'today' or 'next year' or by simple or compound quantitative variables ranging over expressions such as (for dates) '1952' or 'June 4, 1952' or (for times-of-day) '13:59:59', and so on. Space, similarly, can be captured by variables ranging over expressions such as 'London', or 'London, Ontario', or over numerical Longitude/Latitude specifications. Information along the Object dimension is indexed by a variable ranging over object instance or object class labels such as 'Emmanuel Macron' or 'bicycle' or 'surface-to-air missile (SAM)'. Information along the Process dimension is similarly indexed with a process variable ranging over expressions such as 'flying', 'fueling', and 'aborting mission'. As with BFO, subclass and part-whole relations are supported for all values of all ontological dimensions.

POR representations may vary in the number of values specified for any particular ontological dimension. Thus, a POR representation may contain more than one Object dimension value (where we have information concerning interactions between multiple objects) or more than one Process dimension value (where we have information that an object or objects participate in more than one process). However, all ontological dimensions included in a POR representation will be drawn from the four we have listed, or from their combinations or from sub-dimensions of one or other of these – as when we distinguish under Space sub-dimensions such as Geographic, Astronomic, and so on.

PORs also vary in the degree to which their ontologically dimensioned variables are associated with values of greater or lesser specificity. At one extreme, are variables whose associated values have a very high degree of specificity, which may be either quantitative (47.366° Celsius) or individuative ('Paul McCartney', 'Eiffel Tower'). At the other extreme, there are no values at all, but only dimensioned ontological variables (captured in expressions such as 'something', 'somewhere', 'sometime', 'somehow').

An ontological grammar specifies which collections of ontological dimensions and values can be composed into a POR representation and under what conditions. In our work here, we start from the simplifying assumption that PORs occupy some space-time interval, thus excluding from our purview not only POR representations of abstract structures such as numbers or geometrical shapes putatively outside space and time, but also POR representations consisting only of time or only of space dimensions. This does not mean that spatial or temporal statements ("Jessie's house is 15 meters wide"; "Yesterday's race was run in less than 10 seconds") cannot be made in isolation. The applicability of a spatial dimension is, however, taken to imply the applicability of a temporal dimension and vice versa.

3.1.1. An informal initial example

Consider Figure 2 which portrays an illustrative POR representation intended to resemble the kind of data table that one might find in any database application (with column headers which identify the ontological dimension; and a row, or perhaps many rows of data underneath). A more formal vocabulary is introduced following the description of a formal grammar in section 3.3.1. POR representations can be *incrementally specified* by successively assigning (which means: filling in) values for their ontologically dimensioned variables. In any given specification step, we add a value to what had hitherto been unspecified – for

---

[7] For our model ontology, the term 'attribute' was introduced because it matches the language of practitioners in the database community with whom we were interacting. 'Attribute' generalizes 'specifically dependent continuant' in BFO.

Time	Space	Object	Attribute	Process
t	*s*	Grace	*a*	*p*

Figure 2. An initial POR representation functioning as a POR template.

example by replacing 'something' in an Object cell with 'Grace', as in Figure 2. We refer to the starting point of each such process as a *POR template*.[8] The results of specification of such templates are what we have already recognized as POR instance assertions. The process is iterative, so that a POR instance assertion in an earlier specification step may itself serve as a POR template for a later step. Formally, whenever a specification step occurs, as in Figure 3, the cells already specified – here: 'Grace' – form the argument. The value asserted (i.e., newly filled in) – here 'Smiling' – forms the logical predicate of the assertion. And in Figure 4, Grace and Smiling both form a part of the argument which might form the backdrop for a sentence such as 'When Grace is smiling she puckers her cheek'.

We are to imagine Figure 2 as the starting POR template for the task: to collect data about a certain object Grace. The Object dimension is accordingly assigned the value 'Grace'; the italicized letters *t, s, a,* and *p* stand in for variables along their respective dimensions.

Figure 3 shows the additional value 'Smiling' assigned to the Attribute dimension as the result of a first specification step (#1). This corresponds formally to the POR instance assertion Smiling(Grace) generated as an interpretation of a text input from an observer, such as "Grace is Smiling". Grey highlighting is used for those values in a POR representation that are asserted – here the Attribute Smiling.

A single POR template may contain one or more unspecified variables. And a corresponding POR assertion may contain one or more asserted predicates (shown in these examples as cells with grey highlighting).

Figure 4 shows the same POR representation as in Figure 3, but now serving as a template for a further assertion in a new specification step (#2), with 'Smiling' de-highlighted.

Time	Space	Object	Attribute	Process
t	*s*	Grace	Smiling	*p*

Figure 3. A POR representation functioning as a POR instance assertion
for specification step #1.

Time	Space	Object	Attribute	Process
t	*s*	Grace	Smiling	*p*

Figure 4. The POR representation asserted in step #1 now serving
as a POR template in step #2.

Time	Space	Object	Attribute	Process
t	*s*	Grace	Smiling	Walking

Figure 5. The POR instance assertion in specification step #2

Finally, Figure 5 shows the POR instance assertion generated in specification step #2; an assertion to the effect that Grace participates in a walking process.

---

[8] The POR representation can be more formally depicted as a mathematical function or relation, the vocabulary for which we will introduce beginning in section 3.3.1.

Formally, the result of combining the POR instance assertions in Figures 3 and 5 can be expressed as a simple conjunction: Smiling(Grace) and Walking(Grace).

3.1.2. Adding sensors

Imagine, now, that our ODIS is connected to a video sensor. A POR representation might incorporate dimensions representing the system time for each instant of a received video as well as the spatial location (and perhaps also other features such as orientation) of the video sensor, as captured in Figure 6.

If the sensor has on-board classifiers for Object and Attribute recognition, the sensor might classify its video imagery at the given time as video of: object Grace with attribute Smiling, thereby generating the POR instance assertion depicted in Figure 7. The representation of the sensor's classification would depend on how the classifier functions were implemented. For example, recognition of Smiling might be dependent on the prior representation of the specific person involved. The formalized representation would then take the form of a sequence beginning with 'Grace(09:32:01, 45.32, 52.63)' and moving from there to 'smiling(Grace(09:32:01, 45.32, 52.63))'. In the former 'Grace' plays the role of predicate, in the latter 'Grace' is part of the argument, reflecting our use of the semantic type freedom approach of Common Logic (Menzel, 2013).

If the sensor is simultaneously connected to a text feed, and also has knowledge that enables it to classify time and space regions not merely quantitatively but also qualitatively (for example, using terms such as 'morning' or 'next fall' or 'my backyard') then this may result in a POR template as in Figure 8:

Time	Space	Object	Attribute	Process
09:32:01	45.32, 52.63	*o*	*a*	*p*

Figure 6. A POR template generated by an ODIS connected to a sensor feed.

Time	Space	Object	Attribute	Process
09:32:01	45.32, 52.63	Grace	Smiling	*p*

Figure 7. A POR instance assertion connected to a sensor feed.

Time	Space	Object	Attribute	Process
Morning	Backyard	*o*	*a*	*p*

Figure 8. A POR template connected to a text feed.

Time	Space	Object	Attribute	Process
Morning	Backyard	Grace	Smiling	*p*

Figure 9. An initial POR instance assertion connected to a text feed.

If the system now receives as input the sentence: "There's Grace smiling in the backyard", then with the POR template in Figure 8 as starting point this would be captured as a POR instance assertion as in Figure 9, formalized (roughly) as Backyard(Grace, Morning) AND Smiling(Grace, Morning, Backyard).

Depending on the purposes served by the ODIS, different POR representations may contain 0, 1, or more Object dimensions; 0, 1, or more Attribute dimensions; 0, 1, or more Process dimensions; and so on. For example, to facilitate the sensor data capture, the ODIS facilitates POR representations based on incremental recognition steps along multiple dimensions such as are involved in solid object detection, heat source detection, moving object detection, and so forth.

3.1.3. Composing POR assertions

The process by means of which our prototype ODIS creates POR representations from text or sensor data passes through three main stages. In a first stage, text or sensor data are mapped to candidate ontological dimensions and dimension values. In a second stage, each candidate ontologically dimensioned value is mapped to candidate POR representations. In a third stage, the most plausible candidate POR representations are selected using an algorithm that looks for the smallest collection of most detailed POR representations required to subsume all input data.[9] The selected representations are then combined together.

This three-stage process runs in cycles (as described in the appendix), with a new cycle being initiated with each new data input. POR templates and assertions identified at any given stage will form the context within which attempts will be made to interpret new information in such a way as to achieve a single consistent collection of POR representations that properly extends the collection yielded at the conclusion of the prior stage. Sometimes, of course, the process of identifying a single most plausible POR candidate will fail. Where this occurs, alternative POR templates (populated through prior POR instance assertions) will be identified
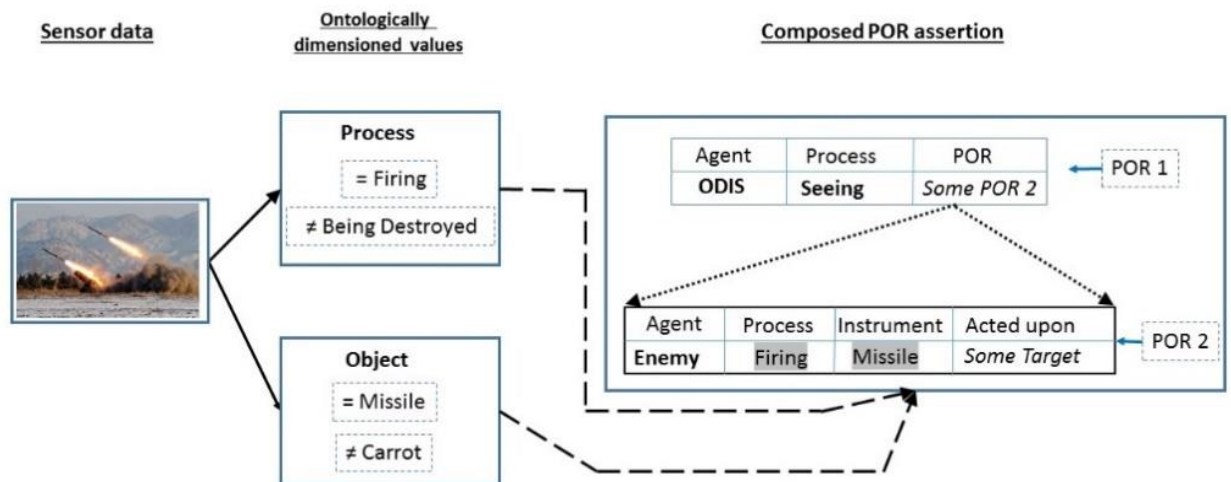

Figure 10. POR instance assertion composed from sensor data

and tested for viability in successive cycles.

---

[9] Our stage-three algorithm is a variant of a similar algorithm for extracting semantics from data for which we have been awarded a patent (https://patents.google.com/patent/US20100169758).

Figure 10 illustrates two further aspects of POR composition. First, it illustrates how one POR representation can serve to capture the contents of an agent's mental acts or states (for instance perceptions or assertions or beliefs). This is indicated here by the small dotted lines with arrows. Second, it illustrates how sensor data – here in the form of imagery – can serve as inputs for POR instance assertions.

The solid lines with arrows connect sensor data with the appropriately matching candidate ontologically dimensioned values. For the Process dimension, Figure 10 shows the selecting of one value ('= Firing') and the rejecting of another value ('≠ Being Destroyed'). What this means is that the classifier determined both that constraints associated with classifying the image as depicting a process of *firing* matched what was found in the image and that constraints associated with classifying the image as depicting a process of *being destroyed* conflicted with what was found in the image. And analogously for the Object dimension ('= Missile' and '≠ Carrot'). In our ODIS described in the appendix there are typically many rejected options.

The dashed arrows track how ontological dimension values selected at an early stage (on the left of Figure 10) are inserted into a POR template to compose a POR instance assertion at a later stage (on the right). Only certain POR template values will be introduced in any given case, either because they are set by the user or because they are inherited from earlier steps; seeded values ('ODIS' and 'Seeing' in POR 1 and 'Enemy' as Agent in POR 2) are shown here in bold.

The ODIS generating the output sketched in Figure 10 is set to work with a reconnaissance sensor observing enemy activities in a specific location. The POR representation variables are shown in italics, with asserted POR values shown in grey highlight. Seeded values in the POR template are used to build a linked pair of POR representations, with POR 1 having the form: ODIS is seeing POR 2; and POR 2 the form: agent of Process in target area has Attribute: enemy. One consequence of this seeding is that, were the ODIS to observe friendly troops firing missiles, it would incorrectly attribute the missile firing to enemy agents. The ray-shaped pair of dotted lined arrows indicate that the value of the POR variable (*some POR*) in the nesting POR representation is specified by the entirety of the nested POR representation. From the POR 2 template for missile firing, the system knows that missiles are fired at some target. The ODIS does not observe/identify the target and so leaves the variable *some target* unspecified.

Figure 11 shows a similar composition process, where an otherwise equivalent POR instance assertion originates with *textual* data. Starting with the sentence "Satellite shows missiles fired" as input, the ODIS matches each symbol to its dictionary of ontologically dimensioned values where, as minimally illustrated here, there would typically be multiple possible senses of each symbol or word each corresponding to a different ontologically dimensioned value. The symbol 'satellite', for example, could mean: a secondary planetary body such as the moon in its relation to the earth, or (as here) a reconnaissance satellite containing an ODIS. Similarly, the symbol 'shows' could mean a plurality of cultural events or (as here) the result of a process of seeing.
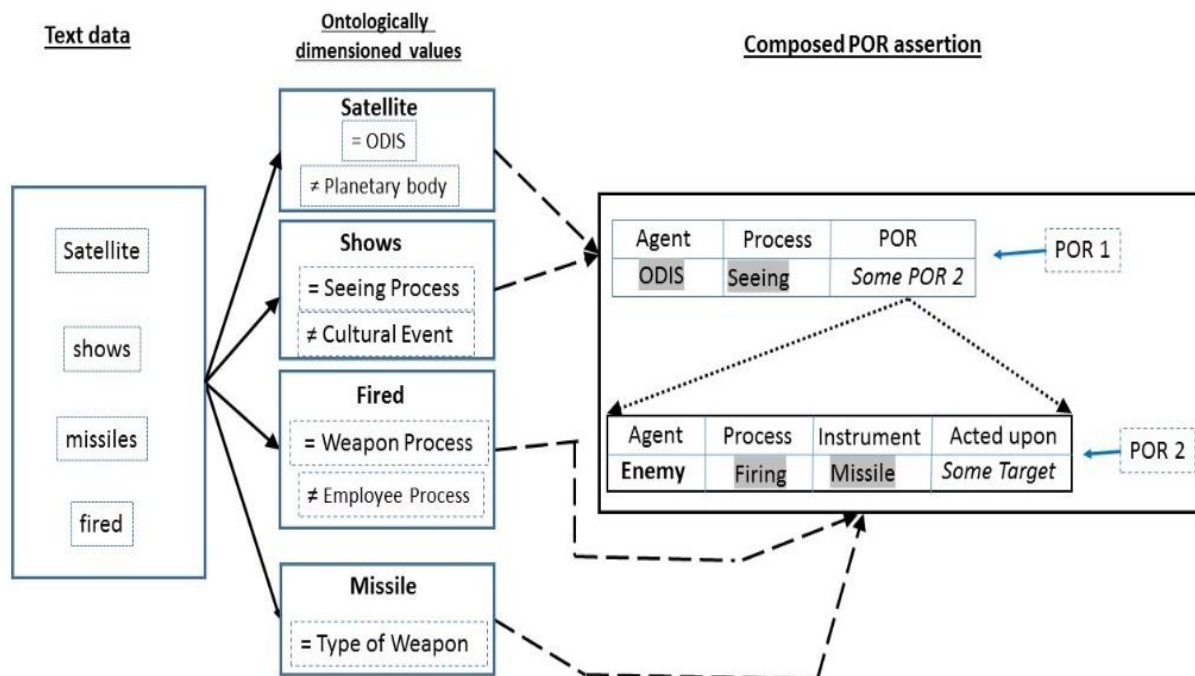
Figure 11. POR instance assertion composed from textual data

As in the process of origination of a POR instance assertion from sensor data, so also here: the ODIS selects ontologically dimensioned values (the values identified with an equals sign) from a repertoire of candidates. These values are then composed into the composite POR instance assertion shown in the right-hand portions of Figure 11. The composite assertions differ between the two figures in the fact that the nesting (upper) portion of the composed POR instance assertion in Figure 10 was seeded with the fixed knowledge that an ODIS was seeing some POR, where in Figure 11, because the source information is text data – in this case, the sentence beginning with 'satellite shows' shown in the left column of Figure 11 – neither the agent nor the process is fixed in advance. Here, it came from processing the words 'satellite' and 'shows'.

Although the kinds of ambiguity that need to be resolved differ between the sensor-based and text-based composition of POR instance assertions, the two resolution processes can still yield matching composed POR instance assertions, which are thus independent of the modality (sensor or text, direct visual observation or signals intelligence) by which they were obtained. At the same time, data relating to the paths by which the two POR instance assertions were arrived at are important. Our ODIS uses POR representations to capture such provenance information also, following the strategy outlined in (Ceusters and Manzoor, 2010). As we shall see, information of this type is of particular importance in those cases where the ODIS yields conflicting POR instance assertions, for example in a case where a sensor classifies the observed missiles as participating in a process of being destroyed (the option marked as rejected in Figure 10).

3.2. Ontological dimensions of a POR representation

We adapt the term 'dimension' in the previous section from the language of multidimensional databases (Thomsen, 2002). Our 'POR instance assertion' is an abstraction of the concept of a 'tuple in a database' as the latter phrase is used, for example, in relation to a database of products whose 'dimensions' might be: Name, Category, Line, Size, Price, and so on (Date, 2003). As already noted, the same dimension can be involved multiple times in a single POR instance assertion. In the firing of a missile, for example, there is a composite spatial dimension incorporating geolocations for both source and destination.

The ontological dimensions used to compose POR representations – Space, Time, Object, Process (and Attribute as a non-primitive dimension)*,* are built in a manner that is conformant with BFO,[10] so enabling us to import other BFO-conformant information. However, while conformant, the ontological dimensions we developed for this project extend beyond not just the upper level terms found in BFO, but also beyond what are typically called ontological items in general. This is because we were looking for a unified treatment of all information (and all associated computations) from sensor data to general knowledge as encoded in natural language.

In our approach, the Space dimension encompasses all possible spatial units and their compositions, and the time dimension encompasses all possible temporal units and their compositions. Similarly, the object dimension encompasses all possible object sub-classes and compositions of subclasses, and the process dimension encompasses all possible process sub-classes and compositions of subclasses. In addition to compositions within a dimension (e.g., a composition of time units to form a date, or a composition of one-dimensional spatial lengths to form an area), our approach supports compositions between dimensions (e.g., a spatial unit and a temporal unit to form a velocity)

In our model ontology, Space encompasses BFO's notions of both spatial regions and sites and 'Attribute' is in effect a generalization of BFO's 'specifically dependent continuant'. For BFO, and for us here, Attributes may be either unary or relational. Unary attributes are for example the *mass* of a vehicle or the *range* of a missile; relational attributes are for example the *ownership* of object A by person B, or the *hostility* shown by group C against group D.

Ontological dimensions naturally constrain expressions. Through suitable changes of instance or of unit any temporal value can be transformed into any other; for example, '2 minutes' into '120 seconds'; '2018 + 1 year' into '2019'. But no such transformation can convert a temporal unit into a spatial unit or into a unit of mass. Similarly, through changes in attributes of size, material composition, shape, and so on, any object representation can be converted into any other. But an object representation cannot be converted into a process representation.

On common parlance, we distinguish between qualitative and quantitative values – corresponding roughly to the opposition between text and sensor outputs. Qualitative values used in the ODIS may come from natural or technical languages or from standard coding systems, for example when places are designated using ISO Country Codes. In many cases, we are dealing with common nouns, which are used as labels for determinate or determinable classes, for example, when the value along the Object dimension is specified as 'Artifact', or 'Ball', or 'Cricket Ball', and so forth.

---

[10] Details are provided in the BFO-CORE-S mapping document at https://bit.ly/2Ecv2Cc .

On closer examination, however, we can recognize that the qualitative-quantitative opposition is not so simple, since qualitative values are often associated with numerical information.[11] Most commonly this is implicit, as for example where common nouns are associated with an implicit count datum, with count = 1 for singular nouns and > 1 for plurals. Sometimes this is explicit, as in 'Seven anti-aircraft missiles'. Sometimes it is more subtle, as when an assertion such as "Mary is arriving later" is followed up with a question – "Do you believe she will get here before 7pm?" – indicating that the information associated with the qualitative term "later" may be compared with the quantitative term 7pm.

To take account of such associations we treat qualitative and quantitative values in a uniform way by extending the term 'unit' to apply not merely to measurement units in the traditional sense but also to what we can think of as atoms or minima on the qualitative side (Bittner and Smith, 2003). For any given context, such atoms or minima are always involved in the process of providing information about reality, for example because of the physical limitations of the measuring devices used. Thus, when formulating qualitative data along the Attribute dimension, for example relating to *color* or *mass*, we employ in each case a granular scale such as the RGB cube for color or grams for mass.

3.3. Making ontological dimensions computable

The ontological dimensions we have discussed point outward into the world in the sense that they correspond, via the POR representations in which they are composed, to PORs that live outside the system of representations. Each POR instance assertion consists of an argument (one or more values of known ontological dimension – for example: SAM launcher) and a predicate (one or more values of another known dimension – for example: moving towards Raqqa). As we see them on the screen (or in the figures that follow) argument and predicate appear as strings of symbols. Behind these symbols, however, is an additional layer – pointing inward to the system itself – that includes computing operators such as '+' applied to values of Integers or Rationals.

Axiomatized ontologies are computable up to a point. The major dimensions (i.e., the hierarchies that stem from the root encompassing in BFO continuant, independent continuant, material entity, and so on) support computing using generalization, specialization, and using the parthood and whatever other relations are built into the axioms of the ontology. This is sufficient for navigating between terms and for testing whether the axioms associated with a term are consistent with the overall body of axioms. But one cannot use such an axiomatized ontology, for example, to calculate the average across a collection of sensor data or to compare this average with prior knowledge to decide whether to trigger an alert. Computations of these sorts are not performed by traditional ontology software. They do however form part of the properties of PORs represented by our ODIS, which adds to traditional ontology precisely the capacity to view entities through a much broader range of types of computation to which they are susceptible. One area of land, for instance, can be ten times larger than another. But the color of a shirt or the name of a town cannot be ten times larger than some other color or name. The following kinds of calculations, similarly, belong outside the realm of ontology software as currently conceived.

1. Operational status of enemy controlled SAM in 6 hours = operational status now + chance of repairing SAM in 6 hours
2. Position of vehicle in one hour = inside a radius of 100 miles based on current position

---

[11] In the next section, we reduce the quantitative/qualitative dichotomy at the ontological level to differences in fundamental computing dimensions and the comparisons they support.

For the ODIS to work on real data it must be concretized in some computing machinery, with inputs deriving (for example) from observations, outputs consisting (for example) of messages to the commander, and a central processing unit (CPU) that executes instructions related to the information being processed. These instructions might be to store a new expression, to compare a new expression with an existing expression, to infer a new expression from a given one, to execute an expression – if it has the form of a command, and so forth. From the perspective of this CPU, received expressions may trigger one or more CPU actions (store, compare, execute, and so on).

Suppose, for example, that the ODIS receives a field report concerning an observed satellite. The CPU converts this report into a POR instance assertion p and then executes operators to compare the dimension values that make up the argument of p with dimension values in POR instance assertions that the ODIS derived from satellite imagery relating to the same time and place. It will then compare the dimension value(s) making up the predicate of p with the dimension values found in the stored instance assertion and, assuming that what the ODIS has stored is of higher confidence than p, yields the value TRUE if there is a match between the two sets of dimension values.

Each ontological dimension in a POR representation reflects a corresponding computable dimension in the computation layer that is built out of physical (which means here: computable) dimensions (such as Boolean, Categorical, Integer, Rational, Array, Hierarchy, Graph) and combinations thereof.

Computable dimensions define corresponding operators, values and compositional rules realized within the context of computational processes. All values of computable dimensions are generatable by their operators: Booleans by assertion or negation; Integers by addition or subtraction; Categoricals for example by joint negation (which defines for a Categorical dimension with n distinct values an $n+1$st value $v_{n+1} = \text{Not}(v_1 \text{ or } v_2 \text{ or } \ldots \text{ or } v_n)$), and so on.

By examining how information is provided to the ODIS in terms of computable dimensions and associated units, we can now identify the computable dimensions corresponding to different sub-dimensions of the four principal ontological dimensions of our POR representation framework (plus two commonly composed dimensions of attribute and process). Table 1 illustrates examples of such computable dimensions for sub-dimensions of Space, Time, Attribute and Process. In each case, units (given in curly brackets) are associated with the computable dimensions used in formulating the associated information. Note that units may be the result of combining other units, as in 'Meters/Second'.

Important for our motivating scenario was the testing of the hypothesis that, by factoring an upper ontology such as BFO into primitive dimensions within which comparisons could be made but between which comparisons could not be made, and by grounding that refactored upper ontology in a computing dimension layer, we can provide a compositional grammar that allows machine composition and analysis of POR assertions.

The constraint that needs to be met is that the units of every sub-dimension can be calculated from the units of some other sub-dimension(s). For example, spatial units of Kilometers can be calculated by

Table 1. Examples associating ontological and computable dimensions via ontological sub-dimensions and units (R = Rational, I = Integer, C = Categorical)

Ontological Dimension	Ontological Sub-Dimension and Associated {Unit}	Computable Dimension

Space	Geography $\{M^2\}$	R^2
	Geography $\{100M^2\}$	R^2
	Longitude {minutes}	R
	Latitude {minutes}	R
	Geography {county}	Categorical
Time	Time {Month-of-Year}	Integer Mod12
	Time {Year}	Integer > 1800
	Time {Year, Day-of-Year, Hour-of-Day}	Integer Composite[12]
Attribute	Soil_Temperature {°C}	R > K
	Soil_Moisture $\{Grams/M^3\}$	R ≥ 0
Process	CH^4 flux $\{NG /M^2 /HR\}$	R ≥ 0

multiplying spatial units of Meters by 1000. A Euclidean metric of M^2 can be calculated from a spatial unit of Meters by composing the Cartesian product of two spatial units of Meters.[13]

Along the Object dimension we can distinguish sub-dimensions for: *warfighter*, *vehicle*, *weapon*, and so on. Each of these may generate further sub-dimensions (for *weapon* these might be *handgun*, *shotgun*, and so on). Along the Process dimension sub-dimensions might be: *logistics*, *operations*, *intelligence*, and so on.

Sub-dimensions of Time have been discussed already. To enable computation with those varieties of time information that are associated with Integer-based computable dimensions we leverage the Allen Interval Calculus (Allen, 1984), but modified to allow computations over multiple resolutions or scales (Thomsen, 2002). Such computations may involve a variety of concrete temporal units, from nanoseconds to millennia. They may also be linear (as in the case of hours and years) or cyclical (for days and months).

We also need to support computations that pertain to mixtures of both temporal intervals and points. Thus, when we ask how many riots occurred *in* 2016, we are treating '2016' as referring to an interval. When we ask how many riots occurred *before* 2016, we are treating the same data element '2016' as proxy for a temporal instant (roughly: midnight on December 31, 2015). We need further to support computations that relate to composite representations as in the case of a date – such as 'February 15, 2016' – composed of three conjoined representations of instances of *month-of-year*, *day-of-month*, and *year*, and incorporating combinations of both cyclical and linear units. We also need to compute effectively with ellipses as in 'February 15' or 'the 15th' or 'tomorrow' by inferring missing information from the wider context.

---

[12] Formally speaking, the computing dimension is more complex. When we refer to a year, for example, we typically employ an integer with a constraint identifying an earliest and latest year that defines an outer interval. Day-of-year is an Integer Modulo 365 or 366 and needs its own composite time to define the constraint (thus if year is divisible by 4 then Day-of-year is Integer modulo 366 Else modulo 365). Hour-of-day, similarly, is Integer modulo 24 with constraints on the result of 'increment by 1' that reflect, when applicable, the time changes of daylight savings time.

[13] Practically speaking, our approach supports the considerable units work already in the public domain such as is found at Astropy (http://docs.astropy.org/en/stable/units/index.html).

3.3.1. A grammar for composing computing dimensions

Within the context of any given POR assertion, each ontological dimension is associated through its units with a single computing dimension. Each computing dimension is composed out of three elements, namely:

- Computing dimension values (for example the values 2, 3 and 4 of the Integer dimension)
- Computing dimension identifiers (such as 'Boolean' 'Integer')
- Computing dimension processes (for values along the Rational dimension, the process of incrementing (+), decrementing (–), and dividing (/), and so on)

Because computing dimension processes are realized in time and exhibit temporally differentiated states – as the state of *being executable* obtains before the state of *executing* which obtains before the state of *having been executed* – they are fundamentally more complex than values or identifiers. Each realization of a dimension process takes one or more input dimension values and produces one or more output dimension values.

Computing dimension processes thus have two interrelated aspects – a *function aspect*: what function is performed on input values in order to produce output values, and a *relation aspect*: how the output data values are related to the input data values. These aspects need to be treated as distinct for three reasons: (1) natural language distinguishes function and relation words, the former typically capturing a more dynamic perspective, the latter a more structural perspective; (2) sensor data can distinguish temporal patterns (which commonly represent the realization of the *function* aspects of computing dimension processes) from spatial patterns (which commonly represent relations); (3) the ODIS can know the function that will be applied without yet having information about the output, and it can know the relation between input and output without knowing how one value could be converted into the other; thus the way it tests an asserted function is different from the way it tests an asserted relation. To see how this works, consider the sentences:

1. The bunker was bombed into rubble.
2. The rubble has a different hyperspectral signature than the bunker.

To understand 1., the ODIS would need to know that the term 'bombing' denotes a function which, when applied to an input object labeled 'bunker', typically produces an output object 'rubble'. To test the validity of 2., an ODIS might inspect field reports or analyze relevant sensor data.

To understand 2., the ODIS would need to know that 'greater than' is a relation between quantitative values and that entropy is measured as a quantitative value. To validate 2., an ODIS would need to have information about the hyperspectral signature of a bunker and the hyperspectral signature of the kind of rubble that would be produced by bombing the bunker, compare the two, and then compare the resulting comparative assertion with the asserted comparative assertion.

As we have outlined, PORs, when viewed on the level of computation, take on one of two aspects depending on whether what is asserted is a *function* or a *relation*. Where the former captures a more dynamic perspective, the latter captures that structural feature of the POR that is unfolded by a given POR assertion. The following terms are used to formally represent the computing dimension elements in terms of which functional and relational POR assertions are expressed:

Table 2. POR instance assertion forms that can be computationally processed.

Assertion form	Interpretation and examples
ρ(N:v, M:w)	N:v stands in the relation ρ to the argument (M:w) Integer:6 is smaller than (Integer :9) • Object:video_sensor is more reliable than (Object: satellite image) • Object:Mary is commanding officer of (Object:Bill) • Object:Bill is child of (Object:Mark)
N:*v* ◄ f(M:*w*)	Executing the function f on the argument (M:w) produces the output N:v • Integer:9 ◄ +(Integer:6, Integer:3) • Object:Scott ◄ Get_author(Object:Intel Report) • Object:Rubble ◄ Bomb (Object:Building)

- Upper case Roman letter terms 'N', 'M', … are used to refer to computing dimensions for example as in: N = Integer, M = Boolean
- The terms 'N:v', 'M:w', are used to denote values (instances) of dimensions N, M.
- The terms 'N:*v*', 'M:*w*' (with *v* and *w* italicized) are variables along dimensions N, M.
- The complex term '◄ f( )' is used to represent a function-like process (hereafter called a function) that takes one or more dimension values within the parentheses as inputs and produces one or more dimension values to the left of the ◄ for output.

As an example: 'N:v ◄ f(M:w)' means: executing the function 'f' on the argument (M:w) produces N:v as output, where 'v' and 'w' (non-italic) are values. Thus '7 ◄ square root(49)' asserts that 7 is generated as output by executing the function square root on the input 49.

When in addition to the function, dimension value(s) are only given for inputs, the expression as a whole expresses an executable command. Thus 'N:*v* ◄ Square root(49)' means: take the square root of 49 and assign that output value to the variable (with *v* italic): N:*v*.[14]

- A complex term of the form 'ρ(,)' is used to represent the relation aspect (hereafter called: relation) between two or more values of a given dimension.

As an example: 'ρ(N:v, M:w)' means: the argument N:v stands in the relation ρ to the argument M:w. Thus the function assertion 7 ◄ square root(49) could be represented as the relation *square_root*(7, 49).

These two forms of POR assertions amount to what we shall call a grammar of allowable combinations of elements for computing dimensions. This grammar has been implemented in our prototype ODIS as described in the Appendix, below All and only those interpretations of text or sensor data whose

---

[14] Both the function and the relation assertions are, in computer science terms, expressed with equality operators, denoted in C and in Java) by the double equals sign '=='. See for example http://www.gnu.org/software/octave/doc/v4.2.0/Comparison-Ops.html. This is in contrast to executable commands expressed with an assignment operator typically denoted by the single equals sign '='. Note that 'function' in the text is used in the sense in which this term is used in mathematics and in computing, rather than in the (biological and engineering) sense defined in BFO.
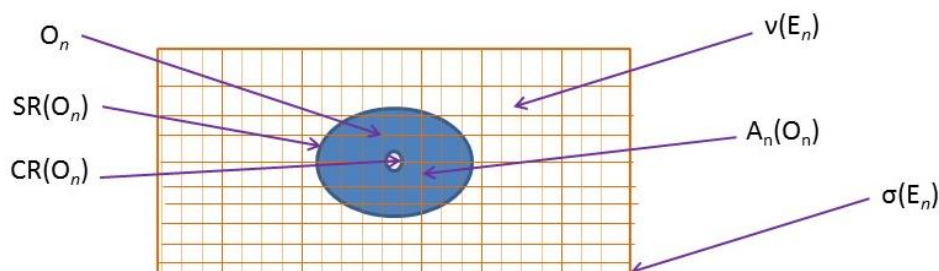
Figure 12. A simple POR representation E_n at time T_n.

corresponding ontological dimensions adhere to these two patterns of computing dimension elements constitute POR assertions within our framework. Table 2 illustrates how these two POR assertion forms can be used to formally represent a variety of sentences covering numeric, sensor and text examples.

Of course, these purely grammatical restrictions for computing dimensions do not exhaust the requirements which combinations of ontological dimension elements must satisfy if they are to compose POR instance assertions. There are, as we shall see, also purely ontological constraints (e.g., subatomic particles have no color; human-sized objects fall towards planetary-sized sources of gravitation). There are also what we might think of as empirical constraints, deriving from the fact that computation will of course standardly occur not over the content of one single POR instance assertion but rather over collections thereof. We might, for example, infer computationally that Mary is commanding officer of (Bill) because we have ingested POR representations of military personnel organized by ranks and units.

3.4. The general form of a POR representation

We focus here for illustrative purposes on representations of PORs of finite and connected spatial and temporal extent comprising one or more objects that may change shape over time. Each object may be represented as having one or more attributes and as participating in one or more processes each having one or more attributes in turn.

3.4.1. Atomic POR representations

Imagine that we are using a sensor to track an object (O_n), for example a SAM, as it changes location and orientation over time. All data representing position and orientation of such an object will have a certain minimal time and space resolution depending on the sensor technology used. This implies that we can conceive of POR representations which are *atomic* in the sense that they represent what is for a given technology the smallest discriminable change along each of the five ontological dimensions of Time, Space, Object, Attribute, and Process.

Figure 12 is the first in a sequence of visually depicted POR representations of a POR labeled E_n, which are atomic in this sense.

Along the Time dimension, $\tau(E_n)$ is the smallest time interval within which the whole of the POR represented in E_n is contained; $\mu(E_n)$ is the smallest temporal unit determined by the resolution of the sensor. Information about $\mu(E_n)$ is required to allow the system to drive the use of clocks to mark off just the right amounts of the right sized units of time. (See Table 3.)

Along the Space dimension, the maximal spatial region $\sigma(E_n)$ observable by the sensor is represented by the entire grid in Figure 12 and the minimal discriminable spatial region $\nu(E_n)$ by the cells in the grid.

Table 3. Syntax for describing temporal and spatial dimensions of POR instance assertions

$\tau(E_n)$	unit-specific representation within E_n of the smallest time interval within which the whole of the POR represented by E_n occurs
$\mu(E_n)$	unit-specific representation within E_n of the smallest temporal unit used in E_n. (This is the information required to allow the system to drive the use of clocks to mark off just the right amounts of the right sized units of time.)
$\sigma_t(E_n)$	unit-specific representation, for each time t in $\mu(E_n)$, of the smallest connected spatial region within which all the objects participating in the POR represented by E:n during t are located
$\nu_t(E_n)$	unit-specific representation, for each time t in $\mu(E_n)$, of the smallest spatial region represented as containing objects existing during t. (Corresponds to resolution, for example of satellite imagery.)

Along the Object dimension, E_n has a single object O_n, whose centroid $CR(O_n)$ and outer surface $SR(O_n)$ are illustrated, respectively, by the center and circumference of the oval in the Figure 12. The POR has one attribute $A_n(O_n)$, represented by the color of the oval.

3.4.2. Minimal POR representation of change

The POR representation illustrated in Figure 12 represents an atomic POR of minimal temporal extent involving *no change* in its object O_n.

Now, however, we can extend E_n to incorporate the case in which exactly *one change* occurs along some single dimension. There are exactly four possible unidimensional changes of this sort for the family of PORs we are treating, of which two are changes involving motion.

The first atomic motion change is a change ΔS in spatial location of the centroid of O_n across some time interval ΔT. Here ΔS is the distance between the positions occupied by the centroid at times T_n and T_{n+1} separated by the time interval ΔT. ΔS can be expressed as the absolute value in some unit of $S(CR_n, T_n)$ minus $S(CR_n, T_{n+1})$. O_n's velocity at T_n can be expressed correspondingly as $\Delta S/\Delta T$.

The second atomic motion change is a change in the orientation of the object O_n while its centroid is fixed, captured as a change in the location of its surface (SR).[15] Assertions about motion of objects salient to intel reports can be represented in these terms by taking advantage of specializations, for example for SAMs, replacing 'movement' by 'movement along flight path such-and-such.' Assertions about the re-orientation of SAM launchers or radar antennae or solar panels can be represented similarly.

The third atomic motion change is a change in some attribute, for example troop strength or morale.

---

[15] Our 'motion of centroid' is roughly equivalent to the 'path predicate' described in Pustejovsky and Mozkowicz (2011). 'Motion about centroid' is however more atomic than their 'manner of motion' expressed using verbs such as 'bike' or 'run', since the latter presupposes a certain complexity of structure. The description of what constitutes a *biking* manner of motion, for example, needs to reference the movement of certain body parts (feet, lower legs, knees, upper legs) in relation to a relatively fixed body part (the torso). A manner of motion can be independent of the act of movement from one place to another. Thus one can bike or run in place, and then the movement is of an object about its centroid in contrast to movement of this centroid. In each case the movement needs to be described relative to some frame of reference, either as a change in the position of the centroid relative to this frame (when the object moves in space) or as a change in the position of parts of the object (for example the surface) relative to a frame of reference (for example when the object rotates on an axis).

The fourth and last sort of atomic motion change is change of identity, exemplified for example when a plane or a building is destroyed, or when a pontoon bridge is constructed out of shallow-draft boats.

We call these four cases of unidimensional change, plus the one case of no change illustrated in Figure 12, the five atomic POR representations.

Based on representational flexibilities inherent in computing dimensions, each of the four atomic changes can be represented in two ways: either as the implicit difference captured by two POR representations each of which represents reality at a different time point, or as that which is represented by a single differential expressed relative to a single minimum time interval. For example, if on day one, enemy troops are of high morale and after two weeks of fighting their morale is low, then the two assessments can be represented as two attribute values each associated with a different time point within a composite POR representation. Such assessments could also be represented as a single differential at a single point in time, as when an intel report taken on day 5 calls out that troop morale is plummeting. Likewise, speed can either be inferred through two measurements of position or it can be measured at a point in time.

As the number and complexity of objects in a POR grows, so too does the number of interdependencies between the objects, the processes in which they participate and the associated attributes within the relevant space-time interval. These interdependencies rest on a series of ontological constraints such as, for example, that inanimate objects cannot be the agents of reproductive processes. Processes that require time to execute cannot be a response mechanism for interacting with an environment that changes at 100 times that rate. Liquid objects cannot provide a surface for solid metals to stand. Bulky objects cannot fit in microscopic space intervals. And so on. We naturally learn these constraints, however approximately, by interacting with the world.

By incorporating such constraints into our ODIS, we can supplement the general form of a POR representation with a range of different sorts of specializations. In this way, we can build a repertoire of increasingly complex POR representations relating to objects, processes and attributes of specific types and, importantly, the specialized roles that different kinds of objects may play in different kinds of processes. Our strategy in building our ODIS is to allow easy creation of new POR representations at any given stage as new sorts of data about new sorts of PORs are incorporated into the framework.

3.5. Composition of POR representations and ontologically grounded thematic roles

To illustrate the feasibility of our approach we have thus far incorporated POR representations mapping over 300 English verbs. We have created for this purpose an ontologically grounded and composable representation of the thematic roles (agent, patient, instrument, and so on) identified by Fillmore (1968) and his successors. This allows us to assign thematic roles to the objects participating in given composite processes and to generate new thematic roles from prior roles in order to deal with new sorts of PORs not adequately represented by the existing POR representations.

Where PORs are comprised of two or more objects participating in the realization of a single process, the mapping of a sentence about that POR to its POR representation in our system requires some method for assigning the objects referenced in the sentence to their respective process roles (such as agent and patient) in the POR representation. This in turn requires a description of the roles themselves (for example that one agent moves first) that can be used for purposes of inference.

The challenge is to develop an account that will allow new thematic roles to be identified as the system goes to work, for example in responding to new kinds of sensor inputs. We will illustrate how this is effected by our system in two stages. First, we will show how two atomic POR representations may be merged to form a single composite POR representation within which new thematic roles naturally emerge. Second, we will provide a series of examples approximating what happens when our ODIS is exposed to new kinds of PORs.

3.5.1. Composition through merging POR representations

Consider a simple case of the construction of a composite POR representation through a process that begins with the merging of data deriving from multiple sources – in our example from:

> (a) a field analyst's report that enemy troops are moving on the ground – illustrated in the upper panel E_1 of Figure 13, representing a unit O_n of enemy troops with two observed positions at two times T_α and T_ω, from which a velocity was calculated for which the directional component is shown with a green arrow,

combined with:

> (b) a satellite report that unarmed civilians are holding 50 miles south – illustrated by the stationery object O_m in the lower panel E_2 of Figure 13.

Composition of these proceeds through a series of steps, each of which must succeed for composition to be successful.

The first step is composition of the Space and Time dimensions of the two representations in order to compare the outer spatial and temporal intervals for each of the two POR representations. If the two sets of dimensions use the same units, or if they can be made to match through an appropriate unit transformation, then the corresponding intervals can be merged without further ado. If the spatial dimensions do not match – as in a case where we have (1) two-dimensional spatial tracking information about a missile but need to use (2) a three-dimensional space for situational awareness – then for composition to succeed the unspecified spatial dimension in (1) must be assigned some value (Thomsen, 2002).
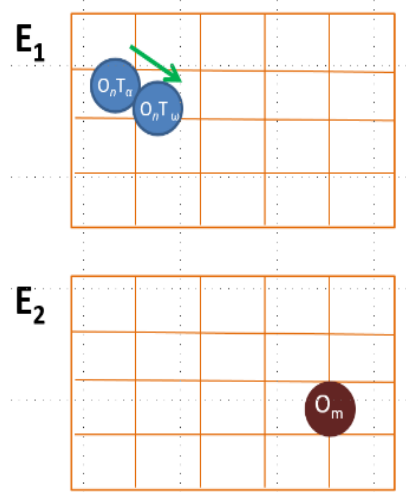
Figure 13. Initial POR representations

In the second step, to which the system proceeds only if the space-time dimensions successfully merge, the information along the object dimension of the two POR representations is tested to see whether it can be integrated without violating any ontological constraints, for example by verifying whether distinct solid objects are represented as occupying distinct spatial locations. If no ontological constraints of this sort are violated, then composition can take place.

Third, is testing whether the merger of attributes violates any constraints. Since the time intervals across the two POR representations may be different, identical attributes need to match only if they are taken from the same time interval. If the time intervals are different, then empirical constraints (for example concerning how fast an attribute like color or size or velocity can change) are used to determine how great a difference in attribute values is composable across disparate time intervals.

Based on the successful matching of the space and time units and intervals of E_1 and E_2 from Figure 13, the two POR representations are composed into a single POR representation shown in the left panel E_{12} of Figure 14, whose containing space and time intervals match those of E_1 and E_2. Finally, the right panel E_{12}, which is inferred by the system, shows the consequences of enlarging E_{12}'s time dimension and the incorporation of predicted inferred
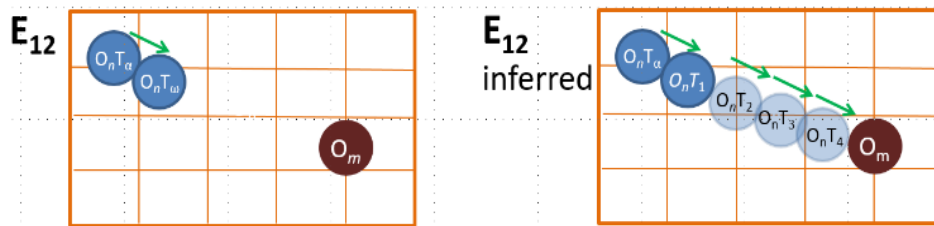

Figure 14. Merger of E_1 and E_2.

positions for O_n and of a predicted time when the enemy troops, O_n, encounter the unarmed civilians, O_m.

While both O_n and O_m were treated in the foregoing as objects with no complexity of structure, the system can infer such complexity over time as further information is acquired. Suppose, for example (incorporating different elements from our motivating scenario in section 1.1):

- Attributes of the second object change: chemical weapons make a village uninhabitable
- The centroid position of the second object changes: troops moved the SAM to a new location
- Orientation about the centroid of the second object changes: the enemy redirected the SAM
- Identity of the second object changes: the missile destroyed the plane.

In the POR representations that would reasonably be constructed to represent these PORs, a new thematic role appears: what is typically referred to as that of being a *patient* or of being *acted upon*.

In the right panel of Figure 14, O_n is the participant in the motion that *results in* contact with the stationary O_m. O_m is from this perspective *acted upon* in its relation to O_n. We call this kind of POR, which represents some object which changes in some way followed by another object which changes (or perhaps does not change) in some way, a *sequenced change*. In a sequenced change, the ontologically grounded distinction between the object that moved first and the resulting contact with a second object then allows further specialization, for example to: O_n *hit* O_m; O_n *pushed* O_m; O_n *spun* O_m, and so on.

Regardless of whether O_n *slid*, or *ran*, or *flew*, the fact that it participated in a process of *change in position of its centroid* means that certain corresponding spatiotemporal information must be specifiable. Depending on how O_n moved, there must be some substance *towards which*, *away from* or *over* or *under* or *through which* it moved. And while these secondary thematic roles, linguistically termed 'prepositions,' are treated by many standard extraction technologies as words that can be ignored, we treat them here as a significant type of ontological glue that binds individual PORs into larger composites.

Many prepositions form part of the common-sense human process vocabulary that presupposes that the POR being represented is within the gravitational field of a planetary body, typically Earth, much larger than the objects depicted in a POR representation. We call the larger object and its associated POR the *Object frame of reference*; we call the smaller object and its associated POR the *referenced Object*. From this point on, we assume the ontological constraints that follow from the presence of gravity, and focus on referenced objects and their associated PORs.

3.5.2. Composition through specification of composite POR representations

We now provide a summary specification of an illustrative series of POR representations that build on the five atomic POR representations described in section 3.4.2.[16]

Each of our example POR representations involves the specification (using the computable dimension grammar we have introduced) of a process composed of simpler processes (either atomic or previously specified in our sequence of examples). We use 'Change(Obj:n)' as shorthand for any atomic change occurring in an object labelled '1, 2, 3 etc.' T0, T1, T01 in what follows are in each case subdimensions of our Time dimension; The letters v, w, and x refer to arbitrary specific values of any subdimension of Time. 'T0:v' then refers to a specific

---

[16] For the interested reader there is a companion document at https://bit.ly/2QGZiX3 outlining 25 of the simplest POR representations that build off the atomic ones.

value along the T0 dimension. 'T0' signifies the time value just before, 'T1' just after, and 'T01' the interval during the execution of a process. We use T01:α to refer to the first time value (of the smallest discriminable units) in the interval denoted by T01; we use T01:v or T01:w for any given value in the interval denoted by T01; and we use T01:Ω to refer to the last time value (of the smallest discriminable units) in the interval denoted by T01. Note that the values associated with T0, T1 and T01 are specific to a given process. Thus, the T0 value of one process may occur after the T1 value of a different process. We use the shorthand Sp to refer to Space and the letters 'q' and 'r' to refer to values of Sp as in Sp:q or Sp:r. We use the shorthand Att to refer to an arbitrary attribute (e.g., color, mass, density, etc.). And we use the letters 'a' and 'b' to refer to values of an arbitrary attribute as in Att:a or Att:b. We use the shorthand Pr to refer to processes and the digits 1,2,3, etc. to refer to specific processes. When referring to the count of objects or processes or attributes, or when specifying a relation between the values of a dimension, we omit the symbol for the value as in: Count(Obj:) = 1, which reads: "The count of objects = 1", or in: ≠((Att: , (Obj:1)), (Att: , (Obj:2,))), which reads: the attribute value for object 2 is not the same as the attribute value for object 1.

We use '◄' as shorthand for '◄ f ( )' to signify that the assertion in which it is included is a function assertion and the ontological term to the right of the ◄ term is the label for the function. When the argument inside the parentheses and the output are each relative to some context, the context is set in its own parentheses (as illustrated in the example below, where the Ontological dimensions for Object and Time provide the context for Attribute). Thus for example:

(Att:b, (Obj:1, T1:w)) ◄ Change(Att:a, (Obj:1, T0:v)) ;

asserts that some change in an Attribute of Obj:1 which at time T0:v has a value of v, results in a change in attribute value from v to w. We use 'ρ(,)' to signify that an assertion of this form is a relation assertion, for example as in:

≠((Att:, (Obj:1, Sp:q)), (Att:, (Obj:1, Sp:r));

which says that the attribute value b of Obj:1 at Sp:q is not the same as the attribute value a of Obj:1 at Sp:r). We use the semi-colon to demarcate the end of a POR instance assertion.

Our goal is to provide an ontological grammar that is fully grounded in computing dimensions so that all aspects of ingesting, interpreting and reasoning over sensor and text data can be carried out by the ODIS. To achieve this goal, we introduce a label assignment computing function that has a POR representation as argument and a labeled POR instance assertion as output. These label assignment functions come in three kinds, assigning

- Function assertion (*FA-*),
- Relation assertion (*RA-*), and
- Thematic role (*TR-*)

Labels, respectively

The major enablers of POR complexity are the number of objects, the number of attributes and the number of processes involved in a given POR (counts being made always relative to a given level of granularity). As we have seen, when there is only one object with one attribute participating in one process, then there are only a small number of possible PORs and POR representations. Accordingly, we begin our series of examples with collections of POR instance assertions where it is the number of each kind of entity in the POR that is being represented. The second collection of examples specifies the object changes unique to a given

POR representation. The third collection of assertions assigns thematic role labels to kinds of processes (object behaviors).

3.5.2.1. A POR representation for sequenced change

Consider a POR involving 2 objects, 3 processes and 0 attributes:

$\geq$(Count (Obj: , 2);
$=$(Count (Pr: , 3);
$\geq$(Count (Att:, 0);
Function:1 ◄ FA-Label((Obj:1, T1:w) ◄ Change(Obj:1, T0:v));
Function:2 ◄ FA-Label((Obj:2, T1:w) ◄ Change(Obj:2, T0:v));
Function:3 ◄ FA-Label(Union(Function:1, Function:2)
WHERE $\leq$ (Time(T1:w, Function:1),Time(T0:v, Function:2)));

The line beginning 'Function:1' asserts that the label *Function:1* is the result of assigning an FA-label to the assertion embedded in the parentheses, which itself asserts that the change applied to Obj:1 at one time yields Obj:1 at a later time. 'Function:1' is thus the simplest possible labeled POR instance assertion that asserts a change in some object that is identity-preserving (for example because it is a change in color).

Taken together with the line beginning 'Function:2' it asserts that the change in question finishes before or at the same time as Function:2 begins (think of a case where we are dealing with real time feeds of sensor data into the ODIS).

The line beginning 'Function:3' then asserts that Function:3 is the labeled union of the realizations of Functions:1 and :2 occurring in sequence.

Participant.1st changer ◄ TR-Label(Obj:1, Function:1);
Participant.2nd changer ◄ TR-Label(Obj:2, Function:2);

are POR instance assertions which label the objects participating in this union of processes using distinct thematic roles. The two roles in a POR representation for sequenced change are the role of the object as *1st changer* (often called *actor* or *agent*) and as *2nd changer* (often called the *acted upon* or *patient*). Both *1st* and *2nd changer* are specializations of the thematic role *participant* which, following BFO, is the all-encompassing *thematic role* for all objects participating in any way in some process.

Sentences such as "The cue ball hit the eight ball (into the pocket)" or "The bomb destroyed the bunker" can be interpreted as POR representations for sequenced change. The cue ball and the bomb are the objects that change first; the eight ball and the bunker are the objects that change second.

3.5.2.2. A POR representation for transformation of one object by another

Objects can transform or appear to transform themselves, as when metals rust or bodies of gas lose entropy. Objects can transform or appear to transform something other than themselves as when a bomb transforms a bunker into rubble or a body of heat transforms ice into water. We use '*transformations of one object by another*' to express POR representations of processes of this sort and classify them as sequenced changes where the bearer of the *2nd changer* role changes its identity.

Imagine the two-object case where some change in Obj:1 occurs at time T0:v prior to an identity change of Obj:2 so that at T1:w, Obj:2 no longer exists but instead there exists some Obj:3.

```
=((Count (Obj:, (T0:v)), 2);
=((Count (Obj:, (T1:w)), 2);
=((Count (Obj:, (T01:x)), 3);

Function:1 ◄ FA-Label((Obj:1,T1:w)◄ Change(Obj:1, T0:v));
Function:2 ◄ FA-Label((Obj:3, T1:w) ◄ Change(Obj:2, T0:v));
Function:3 ◄ FA-Label(Union(Function:1, Function:2))
WHERE ≤ (Time(T1:w, Function:1),Time(T0:v, Function:2))
AND WHERE ≠((Obj:, T1:w), (Obj:, T0:v));
```

The line beginning 'Function:1' asserts that Function:1 is the result of assigning the FA-label to the assertion embedded in the parentheses in a way analogous to our previous example. The line beginning 'Function:2' asserts that whatever Obj:2 was at time T0:v, by time T1:w it had changed in such a way that it transformed into Obj:3.

Function:3 is again defined as the union of Function:1 and Function:2, and the first WHERE clause again asserts that they proceed in sequence. The second WHERE clause asserts that the identities of Objs:2 and :3 are different. Thematic roles are then assigned by the system as follows:

```
Participant.1st changer ◄ TR-Label(Obj:1, T01:v)
Participant.2nd changer.input ◄ TR-Label(Obj:2, T0.v)
Participant.2nd changer.output◄ TR-Label(Obj:3, T1.w)
```

Note the bifurcation of the *2nd changer* (*acted upon*) role introduced in our earlier example. We now have *2nd changer input* and *2nd changer output* roles.

Transformation is at the basis for describing not just destructive acts but also civilian and military production and consumption, as for example, when a portion of meat is transformed into a hamburger, or a portion of cheese and two portions of bread are transformed into a cheese sandwich.

3.5.2.3. A POR representation for motion of centroid between endpoints

   When we have an object with two identified locations, we may speak of *the motion of the centroid of the object relative to these locations*. Consider Figure 15, which depicts simple two-dimensional sensor data that has been classified as identifying locations and objects. Figure 15 displays three snapshots of a single process – one just before a motion-of-centroid process begins, bearing the time stamp $T0_n$; one during this process, bearing the time stamp $T01_n$; and one just after the process, bearing the time stamp $T1_n$. The union of the three snapshots comprises a sensor representation of a POR labeled E_n. Imagine an ODIS had in a given context seen many motion-of-centroid PORs (as described in section 3.4.2) but had never associated the motion of a centroid with locations for the beginning and ending of the motion. This could be important because an analyst might wish to know something about these locations. Had the end location taken fire? How many civilians were left in the start location after Aid agencies removed those they could? Now, with sensor data as in Figure 15, the ODIS is able to catalog the presence of two additional objects or locations in addition to the object whose motion of centroid was observed. The POR instance assertions that follow build on the atomic POR representation for the motion of the centroid of an object by adding beginning and ending locations.

   $\geq$(Count(Obj:,(T01:v)), 1)
   =(Count(Location:,(T01:v)), 2)
   Function:1 ◄ FA-Label((Sp:w, Obj:1, T01:v+1) ◄ Motion-of-centroid(Sp:v, Obj:1, T01:v))
   Relation:1 ◄ RA-Label($>$(ΔS/ΔT01(Obj:1, T01.v), 0)) ;
   Relation:2 ◄ RA-Label(near-or-inside(Space(Centroid(Obj:1, T0:v)), Space(Location:1, T0:v)
   AND (near-or-inside(Space(Centroid(Obj:1, T1:w)), Space(Location:2, T1:w));

   The line beginning 'Function:1' is to be interpreted as in the previous example, except that the function label is now assigned to a POR instance assertion to the effect that the motion-
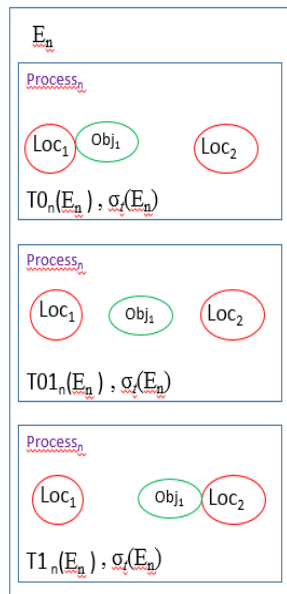

Figure 15. POR representation for motion of centroid between endpoints

of-centroid *function*, when it is applied to on Obj:1 at S:w and at time T01:v, results in Obj:1 being at S:w at time T01:v+1. A similar constraint as imposed by Function:1 could have been imposed through a relation assertion such as:

Relation:0 ◄ RA-Label($\neq$(S:, (Obj:1, T01:v+1)),(S:, (Obj:1, T01:v)))

which asserts that the space position of Obj:1 at time T01:v+1 is not the same as the space position of Obj:1 at time T01:v. The relation assertion expresses the comparison between the space position of the Obj:1 at two times; but it does not point to a process whose execution resulted in the change. It amounts to saying that at those two times the object is not in the same place. Even though the function called 'motion-of-centroid' is not yet associated with ontological attributes such as motion of centroid *by running*, *by flying* and so on, it provides a root placeholder that can be later filled when further information is received. Beyond that, given their basis in spatial comparisons, relational assertions express difference in location but not in motion-of-object (for example an object might disappear from space position 1 and
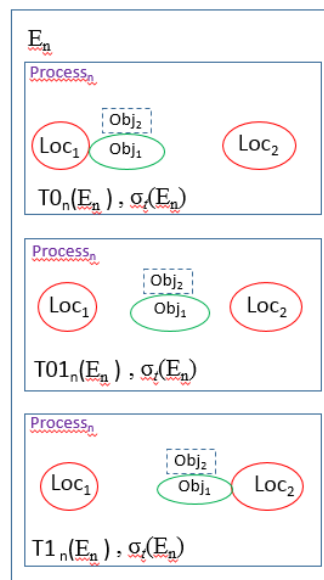

Figure 16. POR representation for Transportation between Endpoints

reappear in some space position 2 without moving). Mathematically, the assertion of a spatial relation is not temporally differentiable. In contrast, a motion of centroid function describes the process of change occurring continuously through time and so is temporally differentiable.

The line beginning 'Relation:1' assigns the relation label to a POR instance assertion to the effect that Obj:1 is moving (i.e., $\Delta S/\Delta T > 0$) during the motion-of-centroid process in S:q and T01:v. The line beginning 'Relation:2' assigns the same label to a POR instance assertion to the effect that Obj:1 was near or inside Location:1 at the beginning of this process and that that same object is near or inside Location:2 at the end of the process. The lines:

Participant ◄ TR-Label(Obj:1,T01:w));
Start location ◄ TR-Label(Location:1, T0:v);
End location ◄ TR-Label Label(Role((Location:2, T1:x);

are to be interpreted along the same lines.

3.5.2.4. A POR representation for transportation between endpoints

Imagine the ODIS has an understanding of the thematic roles involved in the process of motion between endpoints as described, and that its sensors are confronted with a situation captured in the series of snapshots in Figure 16 whose union comprises the POR representation we have labeled E_n. The POR depicted here differs from that in Figure 15 through the incorporation of an additional object – Obj:2 –located on top of Obj:1. Informally we say that Obj:1 transports Obj:2 between Location:1 and Location:2. Sentences such as 'The train carried the troop unit from the city square to the front line', or 'The plane transported the injured from the war zone to safety' would be interpretable in this fashion.

The ODIS might describe the classified sensor data through a series of POR representations as follows:

$\geq$(Count(Obj:, (T01:v)), 2);
$=$(Count(location:, (T01:v)), 2);

Function:1 ◄ FA-Label((Sp:w,:1, T01:v+1) ◄ Motion-of-centroid(Sp:v, Obj:1, T01:v))
Relation:1 ◄ RA-Label($>$(ΔS/ΔT01(Obj:1, T01.v), 0)) ;
Relation:2 ◄ RA-Label(near-or-inside(Space(Centroid(Obj:1, T0:v)), Space(Location:1, T0:v)
AND
(near-or-inside(Space(Centroid(Obj:1, T1:w)), Space(Location:2, T1:w));
Relation:3 ◄ RA-Label(On(Obj:2, T01), (Obj:1, T01))

The simplest description of Obj:2 and of the process in which it is a participant would take the form of an atomic POR relation assertion, namely one of stasis in the spatial relationship of Obj:2 relative to Obj:1. As to the emergent thematic roles, the process of transporting, with the required existence of two or more objects, bifurcates into two sub-processes. As shown in what follows, the generic participant identified in the motion of centroid between endpoint example bifurcates into a participant that transports and a participant that is transported.

Participant.transporter ◄ TR-Label(Obj:1,T01:v))
Participant.transportee ◄ TR-Label(Obj:2,T01:v))
Start location ◄ TR-Label(Location:1, T0)
End location ◄ TR-Label Label(Role((Location:2, T1)

It might seem tempting to branch the thematic roles that emerge during this transport example from those of sequenced changes introduced in our first example, which would mean that Obj:1 here is treated as *1st changer* (*actor*, *transporter*) and the transported Obj:2 as *2nd changer* (*patient*, *transportee*). But such a characterization would impose a temporal sequencing on what is an atemporal relation, namely one of ongoing stasis. Absent additional information we do not know whether Obj:2 controls its spatial relationship to Obj:1 or vice versa, or whether we are dealing with some combination of the two. We do know however that the spatial relation so maintained is the reason why Obj:2 moves in unison with Obj:1, not slightly after as would need to be the case were the act of transport more accurately represented as a branching of sequenced change.

3.6. Conclusion

We hope that through this series of examples of composite POR representation construction for sequenced change, transformation of one object by another, motion of the centroid of an object between endpoints, and transportation between endpoints of one object by another – we have demonstrated that it is possible to ontologically ground thematic roles and to generate new roles from roles already defined by branching (i.e., creating differentiated specializations) from an initial set of atomic POR representations. We do not assert that this particular sequence of definitional assertions amounts to the sole correct account for any of the composite POR representations presented. On the contrary, the entire approach incorporates a great deal of flexibility, corresponding to the flexibility built into the ways in which both natural language and evolving sensor technology represent the corresponding portions of reality and owing to the dual process aspects of function and relation as found in grammar for composing computing dimensions.

We also hope to have shown that it is possible to imbue ontological dimensions with computational capabilities (not just those associated with axiomatic reasoning) thereby enabling ontological representations to reach all the way to empirical data and to the formulas by which data are gap filled, correlated, predicted and otherwise used for decision making. In other words, to produce an information system that is fully driven by ontology (a true ODIS). Finally, we hope to have shown that it is possible to construct a grammar for composing ontological dimensions that is grounded in a grammar of computing dimensions; and that this grounding can enable machines to reason in terms of POR representations.

Appendix: The multi-interpreter prototype ODIS

To implement our POR representation model and test its underlying hypotheses, we built an ODIS called *Multi-Interpreter* (MI). MI provides two distinct subsystems for processing information whether coming from sensors, natural language or databases: a POR assertion slot manipulator that processes
information at the deeper level of computing dimensions, and a POR assertion slot filler that processes information at the ontological (and empirical) level. The ODIS-based process of interpretation passes

Figure 17, a diagram from MI's technical manual, illustrates how a sequence of symbols is processed according to separate slot manipulating (i.e., computing dimension) and slot filling (i.e., ontological dimension) criteria. In the slot manipulating component of Figure 17, E_n refers to an arbitrary expression, L_{nn} refers to the nth argument slot in E_n, C_{nn} refers to the nth predicate slot in E_n. The Trs inside the boxes associated with L_{nn} and C_{nn} represent the dimension roles (called type roles in MI) associated with symbols that are assigned to either argument or predicate slots. The slot filling role component of the Figure 17 (whose detailed description is beyond the scope of this paper), illustrates how surface lexical features or sensor data output (called physical representations in MI) combine with dimension role classifiers for the purpose of associating symbols with slot filling dimension roles.

MI comprises:

Word to symbol mapping: One module in MI consists in a large collection of salient words, each of which is matched with one or more symbols used internally by the ODIS. Each symbol is associated with one or more ontological dimension values (e.g., a time value, a space value, an object identifier). And each conjunctive collection of co-occurring ontological dimension values is associated with one or more POR instance assertion templates corresponding to sentences containing corresponding words. These POR representation templates will in turn *specialize* one or more general POR representation templates. The combination of dimension values and general POR representation templates is called the *ontological sense* of the word.

InformationBase IB: This module stores axioms and definitions from BFO-consistent ontologies and analogous assertions from empirical data related to our intel scenario. Together these are used for language processing and reasoning.

Word Sense Disambiguation: All related information both within the target sentence where the word appears and in nearby paragraphs, as well as salient information from the IB, is used to identify the most likely ontological sense(s).

Simple POR instance assertion Extraction: Simple POR instance assertions may be composed of complex entities (e.g., nations or armies) but they have only one predicate. Because MI is grounded in a small collection of ontological dimensions, there is a correspondingly small number of possible simple POR instance assertions. A single sentence (as shown in Figure 11) may generate one or more simple POR instance assertions. We describe in what follows how MI uses its understanding of possible simple POR instance assertions to look for and extract them from external inputs.

Composite POR instance assertion Construction: Based on our hypothesis that sentences have the same compositional properties as PORs, and that prepositions and logical connectives are the means by which sentences and sentence fragments are bound into larger units of meaning that correspond to larger
and more composite PORs, MI uses ontological relations (e.g., spatial, temporal, control) and logical connectives, identified during word-to-symbol mapping as the links by which the simple POR instance assertions extracted from sentences and sentence fragments are merged into larger, composite POR instance assertions These extracted POR instance assertions are then internally stored as graphs.

POR Assertion Reasoning: A constraint propagation engine is used to merge newly constructed composite POR assertions with the prior time collection of POR assertions and to execute queries against the POR assertions and associated ontologies.

A.1 POR assertions templates and word class sequences

First, MI needs to decide when an N-adic collection of dimensions (e.g., a collection of time dimensions or a collection of object dimensions) represents N-adic aspects of an argument/predicate versus N distinct arguments/predicates. As a general rule, MI treats senses related by granularity (as *hour* is related to *minute*) and space, time and object process senses as N-adic aspects. Thus, in a sentence such as

> *On Thursday, Jan. 15th at 3:00pm on the 10th floor of the Sears tower* an unexpected demonstration took place

the collection of variables in the italicized portion of the sentence all contribute to one location, namely the space-time interval within which the demonstration occurred. So for MI, the spatiotemporal argument would be a single N-adic argument: $f(a_{123})$. In contrast in a sentence such as

Monday, Tuesday and Wednesday at 3:00pm on the 10th floor of the Sears tower an unexpected demonstration took place.
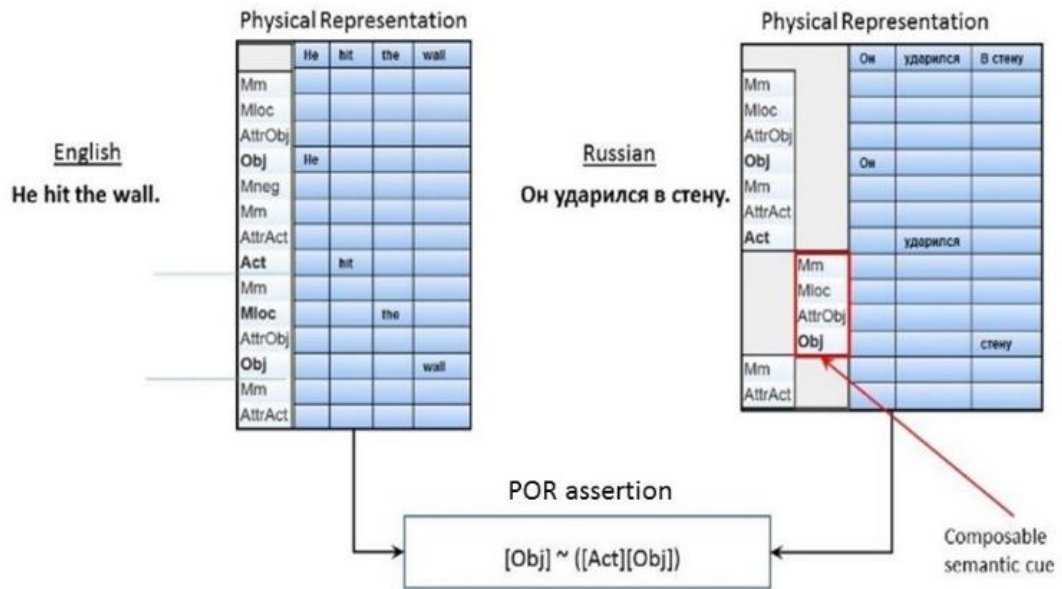

Figure 18. Language Invariant POR assertion

the relationship between days (shown italicized) is not one of granularity and so there would effectively be three arguments, one for each of the three days. Figure 18 shows the similarities of physical representations across surface languages – in this case, English and Russian – and how they can be interpreted by the same POR assertion within the ODIS.

Second, MI understands that prepositions and pronouns such as 'that', 'who', 'where' have more than one sense. Specifically, MI makes use of these terms as markers for determining when a new symbol indicates the presence of a new POR assertion and thus the need to spawn a new physical representation matching process. In response, it determines what kind of link binds the two physical representations. For example, seeing a 'who' or a 'that' might trigger a nested expression link to a new physical representation matching process. Being able to extract and differentially represent nested from nesting information is important because it separates background, implicit assumed-to-be-shared information and from those new statements whose successful conveyance is predicated on the prior, shared information.

Third, MI matches collections of tokens against its simple POR assertions we have described. In implementing our model, we abstracted out from our physical representations those ontological dimensions which could appear in any order in the target language – here English. Time, for example can appear almost anywhere in an English sentence (e.g., Jessie went shopping on Tuesday. Tuesday Jessie went shopping. And Jessie on Tuesday went shopping.) This is why Time does not appear in the POR templates but is a part of the unary

Table 4. Simple POR templates and their word class sequence templates

Abbreviations			POR templates	Word class sequence templates
argument	*arg*	A	(Obj, Attr)	*Arg*-Attr-Obj
predicate	*prd*	B	(Prs, Attr)	Attr-Prs-Attr
object	obj	C	Obj, Rel (A)	(*Arg*-Attr-Obj)-*Prd*-Rel) - *Neg*-(*Arg*-Attr-Obj)
process	prs	D	Attr(A)	(*Arg*-Attr-Obj)-Prd -Attr
attribute	att	E	B(A)	(*Arg*-Attr-Obj)-Attr-Prs-Attr
POR role	evr	F	Obj, Evr(A)	(*Arg*-Attr-Obj)-Prd-(Evr)-(*Arg*-Attr-Obj)

markers. Table 4 shows six of our thirty elementary POR templates and the word class sequence templates they match.[17]

Word sequences found in text are continuously compared with simple composable word class sequences comprised of word classes referring to *objects*, *processes, times* and so forth belonging to different POR representation dimensions. Each such word class sequence is what we called in the main body of the paper a POR assertion. It is a specific physical representation (for example using English) of some POR representation aspect (e.g., of an attribute asserted of an object or of a process and role asserted of an object or of an object asserted of a space etc..) and comes in one of a range of different types called *word class sequence templates*, corresponding to O_n *located in* S_n, P_n *occurs during* T_m, and so forth.

Consider for example the event frame aspect: O_n has attribute A_n during T_m (for example "SAMs undefended at T_m"). When the MI system sees a corresponding word sequence in the text, it allows that the object O_n may come to acquire one or more further attributes that are not referred to in the current assertion, allowing a move, for example, to "Undefended SAMs secured by Blue Force at T_{m+1}," and so on. The system knows also that the object O_n may be pre-associated with attributes, for instance that the undefended SAMs at T_m were enemy SAMs; that they had been positioned to defend an enemy airbase; that they had not been detected earlier, and so forth.

The thirty word class sequence templates are used by the system to classify not merely assertions of an attribute of an object but also assertions about relations between objects, assertions about processes realized by objects, assertions concerning how an object participated in a process, and so on. The templates in addition accommodate different numbers of objects and different numbers of attributes per object (big, old, friendly Bill), and so on. To capture the ways in which sequentially more complex POR assertions are made, the templates involve both logical and ontological markers.

We move from simple POR assertions to complete POR representations by treating those dimensions of a POR representation not filled in the relevant POR assertion as variables (as shown in Figures 2 through 11); thus if we have an IR to the effect that 'airplanes are above Raqua', then the corresponding POR assertion is expanded to a complete POR representation by adding 'at some time', 'moving with some velocity', 'in some direction', and so on. As it sees more text MI will then try to fill in these blanks.

A.2 Illustration of multi-interpreter output and test example

Figure 19 shows the analysis of a sentence in an intel report depicting the matching word class sequence templates and the words that matched them. The POR assertion aspects are arranged vertically across the rows. They are separated by dark horizontal lines. The words are arranged horizontally across the columns with the matches at the intersections. Note that the preposition *with* is part of the original sentence but does not appear matched to a template. This is because MI holds the preposition as a link to be used for merging simple POR assertions into composites

---

[17] The complete set of templates may be found at https://bit.ly/2QGZiX3.
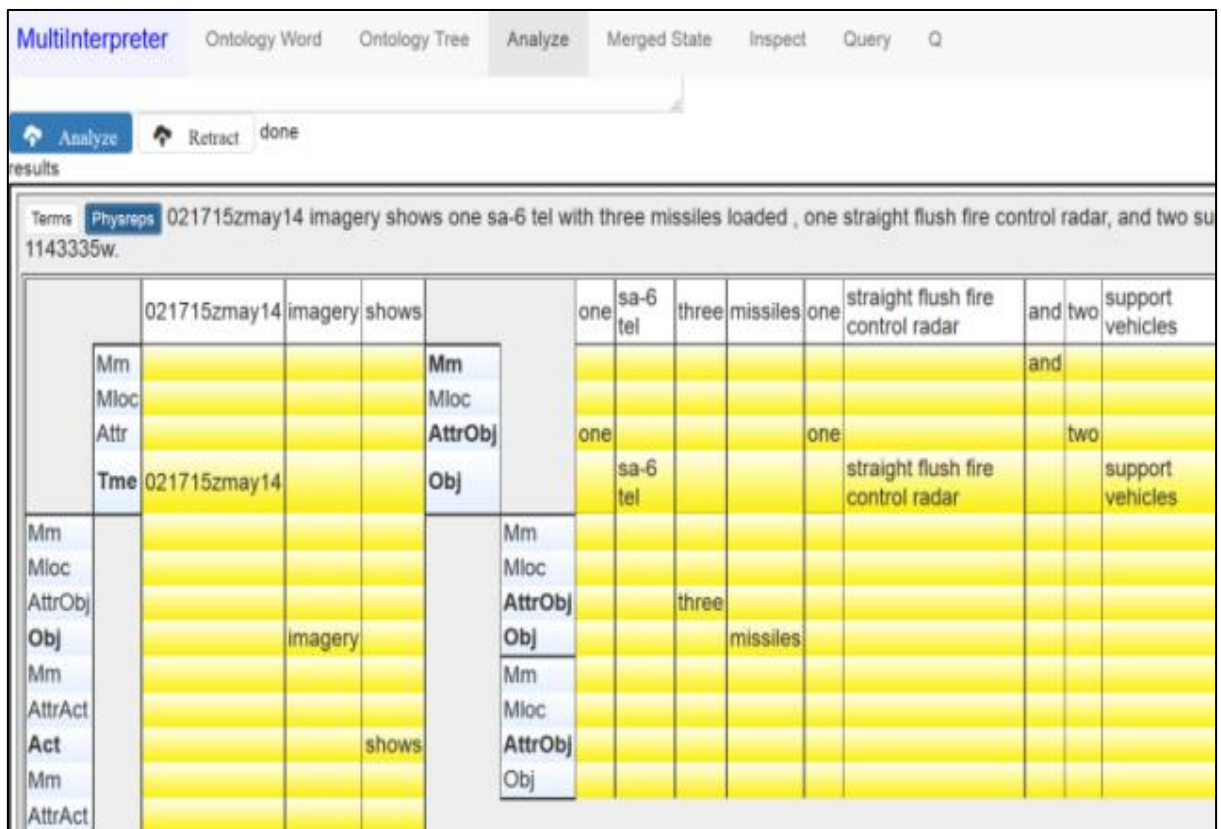
Figure 19. MI screenshot showing partial view of word class sequence matching.

To test MI, we ingested an Air Force ontology generated in OWL and comprising over 1000 entities including red and blue force assets and protocols and built event frames for a synthetic collection of about 200 intel wire reports (of about 3 sentences per report) and for a representative mission plan (containing over 50 mission targets in the form of a Joint Integrated Prioritized Target List or JIPTL) with the intent of discovering whether the activities reported might threaten any impending missions.

The following is representative of an intel wire report. We highlighted prepositions in **boldface** to illustrate their importance for building composite POR assertions

*Recent Reporting indicates **that** enemy is expediting the deployment **of** mobile surface **to** air missiles **to** the front lines. In line **with** this reporting, imagery **from** May141715 shows one SA-T Tel **with** three missiles loaded, one straight flush fire control radar, and two support vehicles **IVO** 12345N 1143335W.*

Potential threats were defined in our ontology as: weapons controlled by the enemy that were in an operational state whose use could destroy an airplane in flight. Actual threats were potential threats located in an area that could attack a planned air mission.

MI successfully interpreted over 90% of the reports and found 8 of the 10 genuine threats and produced two false positives. False positives were produced when unanticipated inferences were needed to understand that a weapon was not in an operational state. (These results were manually verified for evaluation purposes.)

References

Agerri, R., Agirre, E., Aldabe, I., Altuna, B., Beloki, Z., Laparra, E., López de Lacalle, M., Rigau, G., Soroa, A., and Urizar, R. (2014). News Reader project / Proyecto NewsReader. In *Procesamiento del Lenguaje Natural, Revista,53*, 155-158. See also http://www.newsreader-project.eu/

Allen, J. (1984). Towards a general theory of action and time. *Artificial Intelligence*, *23*, 123-54.

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

Baclawski, K., Chan, E., Gawlick, D., Ghoneimy, A., Gross, K., Liu, Z., & Zhang, X. (2017). Framework for Ontology-Driven Decision Making. *Applied Ontology*, *12*, 245-273.

Baker, C., Fillmore, C. F., & Lowe, J. B. (1998). The Berkeley FrameNet project. In *Proceedings of COLING/ACL* (pp. 86-90).

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse* (pp. 178–186).

Bittner, T., & Smith, B. (2003). A Theory of Granular Partitions. In M. Duckham, M. F. Goodchild, & M. F. Worboys (Eds.), *Foundations of Geographic Information Science* (pp. 117–151). London: Taylor and Francis.

Borgo, S., & Mizoguchi, R. (2014). A First-order Formalization of Event, Object, Process and Role in YAMATO. In *Proceedings of Formal Ontology in Information Systems* (FOIS 2014), Amsterdam: IOS Press.

Ceusters, W., & and Manzoor, S. (2010). How to track absolutely everything. In L. Obrst, T. Janssen, & W. Ceusters (Eds.), *Ontologies and Semantic Technologies for the Intelligence Community (Frontiers in Artificial Intelligence and Applications)* (pp. 13–36). Amsterdam: IOS Press.

Ceusters, W., & Smith, B. (2015). Aboutness: Towards Foundations for the Information Artifact Ontology. In *Proceedings of the Sixth International Conference on Biomedical Ontology (ICBO)* (pp. 1–5). Lisbon, Portugal (CEUR 1515), 1-5).

Collins, M. (1999). *Head-driven statistical models for natural language parsing* (PhD thesis). University of Pennsylvania.

Date, C. (2003). *An Introduction to Database Systems* (8th ed.). Pearson.

Fillmore, C. J. (1968). The Case for Case. In E. Bach & R. T. Harms (Eds.), *Universals in Linguistic Theory* (pp. 1–88). New York: Holt, Rinehart, and Winston.

Finzi, A., Pirri, F., & Reiter, R. (2000). Open World Planning in the Situation Calculus. In *AAAI-00 Proceedings*.

Galton, A. (2006). On what goes on: the ontology of processes and events. In *Proceedings of Formal Ontology in Information Systems* (FOIS 2006). Amsterdam: IOS Press.

Galton, A. (2012). States, Processes and Events, and the Ontology of Causal Relations. In *Proceedings of Formal Ontology in Information Systems* (FOIS 2012). IOS Press.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. (2002). *Sweetening Ontologies with DOLCE. Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web (*Lecture Notes in Computer Science 2473).

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, *28*, 245-288.

Kalita, J. (2016). *Detecting and Extracting Events from Text Documents*. University of Colorado, Colorado Springs. Retrieved from https://arxiv.org/pdf/1601.04012.pdf

Kowalski, R., & Sergot, M. (1986). A Logic-Based Calculus of Events. *New Generation Computing*, *4*, 67–94.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: the Penn Treebank. *Journal Computational Linguistics*, *19* (2), 313-330.

McCarthy, J. (1969). Actions and other events in Situation Calculus. Computer Science Department Stanford University. Retrieved from http://www-formal.stanford.edu/jmc/

Menzel, C. (2013). *Completeness Theorem for Logic with a Single Type*. From: http://www.jfsowa.com/ikl/Menzel13.pdf

Niles, I., & Pease, A. (2001). Towards A Standard Upper Ontology. In *Proceedings of Formal Ontology in Information Systems* (FOIS 2001), October 17-19 Ogunquit, Maine, USA, 2-9. See also http://www.ontologyportal.org.

Palmer, M., Gildea, D., & Xue, N. (2010). Semantic Role Labeling. *Synthesis Lectures on Human Language Technologies*, *3* (1), 1–103. https://doi.org/10.2200/S00239ED1V01Y200912HLT006

Palmer, A., Kingsbury, P., & Gildea, D. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, *31* (1), 71–106.

Pednault, E. P. D. (1989). Exploring the Middle Ground between STRIPS and the Situation Calculus. In R. J. Brachman, H. Levesque, & R. Reiter (Eds.), *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89)* (pp. 324-332). San Mateo, California: Morgan Kaufmann Publishers, Inc.

Pustejovsky, J., & Moszkowicz, J. L. (2011). The Qualitative Spatial Dynamics of Motion in Language. *Spatial Cognition and Computation*, *11* (1), 15-44.

Reiter, R. (1991). The Frame Problem in the Situation Calculus: A Simple Solution (Sometimes) and a Completeness Result for Goal Regression. In V. Lifschitz (Ed.), *Artificial Intelligence and Mathematical Theory of Computation* (pp. 359-80). Academic Press Professional, Inc.

Schoening, J. R., Duff, D. K., Hines, D. A., Riser, K. M., Pham, T., Stolovy, G. H., Houser, J., Rudnicki, R., Ganger, R., James, A., and Nagler, E. (2015). PED Fusion via Enterprise Ontology. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VI* (*SPIE Proceedings 9464*). https://doi.org/doi: 10.1117/12.2182064

Schubert, L. K. (1990). Monotonic Solution of the Frame Problem in the Situation Calculus: an Efficient Method for Worlds with Fully Specified Actions. In H. E. Kyberg, R. P. Loui, & G. N. Carlson (Eds.), *Knowledge Representation and Defeasible Reasoning* (pp. 23–67). Boston, Mass: Kluwer Academic Press.

Segers, R., Vossen, P., Rospocher, M., Serafini, L., Laparra, E., & Rigau, G. (2015). ESO: A Frame based Ontology for Events and Implied Situations. In *Proceedings of MAPLEX2015*. Yamagata, Japan.

Shank, R. (1973). The Fourteen Primitive Actions and Their Inferences. *Stanford Artificial Intelligence Laboratory Report*, 73-344.

Smith, B., Kusnierczyk, W., Schober, D., & Ceusters, W. (2006). Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain. In *Proceedings of KR-MED* (Vol. 222, pp. 57–66). Baltimore, MD: CEUR.

Smith, B., Malyuta, T., Mandrick, W. S., Fu, C., Parent, K., & Patel, M. (2012). Horizontal Integration of Warfighter Intelligence Data. In *Semantic Technology in Intelligence, Defense and Security (STIDS)* (Vol. 996, pp. 112-119). CEUR.

Thomsen, E. (2002). *OLAP Solutions: Building Multidimensional Information Systems* (2nd edition). New York: Wiley.

US Air Force. (2016). Contingency and Crisis Execution: The Tasking Cycle. In *Air Force Doctrine Annex 3.0 Operations and Planning*. Retrieved from https://www.doctrine.af.mil/Portals/61/documents/Annex_3-0/3-0-D29-I-OPS-The-Tasking-Cycle.pdf

Van Hage, W. R., & Ceolin, D. (2013). Simple Event Model. In *Situation Awareness with Systems of Systems*. Springer Link.

Van Hage, W. R., Malaisé, V., Segers, R., Hollink, L., & Schreiber, G. (2011). Design and use of the Simple Event Model (SEM). *Journal of Web Semantics*, *9* (2), 128-136.

Whorf, B. L. (19564). *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf*. (J. B. Carroll, Ed.). The Technology Press of Massachusetts Institute of Technology and John Wiley and Sons, Inc.