Representational Reformulation in Hypothesis-Driven Recognition

Ben Rode
Robert C. Kahlert

Cycorp
7718 Wood Hollow Drive
Suite 250
Austin, Texas 78731-01601
http://www.cyc.com
512-342-4022
512-342-4035
ben@cyc.com
rck@cyc.com

Abstract

Formal work on hypothesis-driven models of recognition bring us face-to-face with the problem of ontological interoperability: the ontology used for generating models employed in hypothesis-matching must align with the schema of the databases being searched in order to control the computational cost of the model-building process.

We describe preliminary work toward a system for hypothesis generation and corroboration. The system uses a plurality of ontological representation approaches plus application-specific biases to transform descriptions of probable candidate scenarios into descriptions of observable and deductively provable candidate scenarios, based on the available data sources. The transformation occurs at system setup time. The application biases come from the domain of threat detection.

Introduction

This paper concerns a general approach to recognition using deductive models that casts recognition as a problem of compiling and decompiling information. Assuming that recognition is only possible with a model of what to recognize—a notion we defend below—then models are tied inextricably to the types of data available. For computationally expensive domains such as threat detection, this may indeed be the only way to remain computationally tractable. Unfortunately, at the moment of representing knowledge such as threat scenario templates or background information, it is often not clear what types of data will be available at deployment time and what formats the schemas will take. In addition, the arrival of new types of information could necessitate modifications to the existing knowledge.

The system we are currently designing attempts to tackle this problem in a top down fashion. Threat scenario templates and background knowledge is modeled at a very general level, i.e. applicable to a wide variety of data sources. In addition, the system possesses information about various representational choice points. However, this general description is not intended to be the model that the system uses to detect threats during deployment. Rather, domain specific assumptions and a description of the de-facto available data types are used to transform the general descriptions into knowledge representations specialized for the domain and the available data sources. We envision performing this optimization at system setup time, that is, once—which makes the significant cost of performing this translation acceptable.

The following example tries to explain this tri-partite division of the representational infrastructure. Consider a nefarious action such as a bombing of a public building. From the point of view of the terrorist, all information is available for any phase of the proceedings. This is the most detailed albeit privileged model. From the point of view of Homeland Security, only some of the proceedings are potentially externally observable, such as the purchase of the chemicals. Such knowledge is akin to the domain specific assumptions. Finally, there is a difference between the traces the Secret Service guards could obtain in theory and the sources they can actually scour for traces. These correspond to the de-facto data types available for scenario corroboration.

The remainder of this essay is structured as follows: We first give a problem sketch of some of the philosophical issues of using deductive models for recognition. We then introduce our design for a model for hypothesis generation and corroboration. Next, we discuss the principle components of our modeling approach, that is: ontological choices; application-domain specific constraints; and the knowledge reformulation infrastructure. We conclude with a discussion of the problem of "thinking outside the box" in threat scenario recognition and to what extent the system design we describe contributes to tackling this problem.

Recognition: A Problem Sketch

The field of artificial intelligence has been slow in developing a general theory of recognition. This puzzling state of affairs may be partly explained by philosophical "baggage" in modern thought. The decision of Descartes to postulate a sensorium-based, behavior-independent act of recognition that served as a decisive boundary mediating between afferent and efferent neuromotor action generated an unsolvable metaphysical problem of how a construction from wholly internalized sense data could be known to faithfully represent a mind-independent world. Virtually every philosopher of mind after Descartes followed him in this mistake until the early twentieth century. Not until the emergence of a genuine behaviorism, which defined recognition operationally, as consistently taking appropriate action in response to appropriate stimulus, was much of the mystery dissipated. However, progress based on this behavioral understanding was further hampered by the behaviorist excesses of the 1950s and 1960s, when extremists like B.F. Skinner steadfastly refused to make room for anything as abstract as background knowledge, hypotheses, or mental models for making sense of response to stimuli, even as it became increasingly plain that a great many of the stimuli under consideration were too complex to figure in any other kind of explanation.

Backing this reluctance was a legitimate worry: if talk of 'mental models' entailed interpreting neuronal configurations of the brain as being *about* states of the world, did this or did it not entail smuggling a form of Cartesian dualism back into the picture? Could intentionality be made scientifically respectable? From the 1960s on, the most promising approach was computationalism, which sought to understand the brain as implementing an interpreted computable function that could be abstracted from neuronal interaction.

Yet there remained a two-fold problem. First, there still was the question of what mandated the interpretation. Turing machine interpretations are cheap, such that practically any physical system can be interpreted as implementing any Turing-computable function given appropriate assignation of inputs and outputs to physical states (Assuming intelligence subsisted in such a function begged the question of why we are justified in reading it into a physical system like a brain and not, say, the physical system consisting of the molecules composing a coffee-table). Secondly, there were and are doubts about just how computational common sense intelligence really is. For purely deductive reasoning, such as is used to derive theorems of arithmetic from the Peano axioms, there is a good account. Yet empirical indicators suggest that whatever processes are involved in recognition are the very anti-thesis of formal reasoning in that they are not entirely deductive.

At the very least, prospective recognition systems are confronted by the inconvenient fact of fundamental ontological mismatch between pattern types and data types that cannot be bridged in one direction or the other by inferences that are limited to the material conditional. The suspicion is that this has to do with the need to be able to act in a shorter time frame than a deductive system, given its computational complexity, might allow.

A Model for Hypothesis-Driven Recognition

As mentioned before, this paper treats recognition as a problem in compiling and decompiling information. Our fundamental model begins with an agent that needs to take an action or a set of actions that pertain directly or indirectly to the agent's goals.[1] The agent's goal requirements effectively define a number of types that need to be responded to when they are instantiated in the agent's immediate environment. There is also a relational data set which for present purposes we may take to be a state of the world or a fragment of the world.[2]

We can now formulate the fundamental issue in recognition more precisely, as a problem that emerges in the attempt to model recognizing agents as Turing machine equivalents with inputs and outputs. What is generally found, when this is attempted, is that the mere fact that a certain combination of features is instantiated in data is almost never logically sufficient to support the conclusion that a relevant type is instantiated; by the same token, merely hypothesizing that there *is* an instance of such a type is usually insufficient for deductively inferring the existence of the data features.

Bridging the gap between what is relevant and what is detectable in a computationally effective way is the problem that confronts the recognizing agent, just as accounting for that bridging is the problem that confronts those of us engaged in computationally modeling the behavior of the system. In order to characterize what is happening, it is convenient to work with rules, which are used for characterizing the relevant types and which we may think of as constituting, for all intents and purposes, agent-specific versions of natural laws that compile relevant information about world regularities. We may refer to such a set of rules as a world-theory. The agent is modeled as using this to progress synthetically from invariances, tendencies, and propensities in the input data to existential judgments concerning the relevant types via models that decompile the information stored in the world theory into a useful, data-specific form. This happens via a combination of deductive and abductive inference, which starts from the agent's background knowledge and builds up a large body of statistically relevant prototypical models that in turn are matched against the available data by corroborative database queries.

[1] In a biological scenario, such a goal might be survival.

[2] The data is not characterized as being about the world or representing the world, but as a fragment *of* the world: the problem of recognition, as characterized here, is the problem of building artifacts that respond appropriately to the regular occurrence of structure. It is true that humans approaching a relational database do so with a linguistic understanding that interprets one body of structured data as being about some other body of structured data somewhere else, but this involves more than recognition, and this paper will not concern itself in any very deep way with how this is managed.

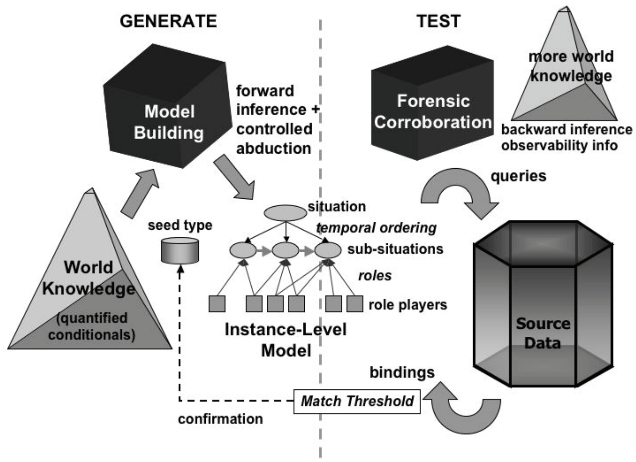

fig. 1. A model for hypothesis-driven recognition.

The challenge inherent in this framework is that of computational tractability. Deductive forward inference is computationally expensive in its own right, and abduction is both combinatorically explosive and hard to control. We would like to be able to appropriately channel model-building based on profiles of the available data, the available inferential functions, and the use case: the more focused the relationships and classes, the more computationally tractable is the reasoning.

However, in attempting to effect focus declaratively, we cannot ignore the question of the computational cost: the possibility at least exists that, for some nontrivial subset of use cases, this will prove more expensive than what could be accomplished by generate-and-test.

Just-in-Time Knowledge Representation

In the remainder of this paper, we describe the design of a system that is the outgrowth of previous work on level-two-three information fusion and on situation-based event extraction using CYC.

From the description of the recognition problem and the explication of our model for hypothesis-driven recognition, it should be clear that the key capability for such a system is the ability to exploit tightly optimized representations targeting the available data, while having the capability to re-target other data sources, if possible. We propose to achieve this through a more general representation and the automatic fitting of that representation to the deployment situation at hand.

Such an approach requires three key components of infrastructure:

A set of knowledge representation choices

Below the upper ontology strata, there are often multiple ways of representing the same information. Ontological engineers usually base their choices (beyond the implementation constraints of representation language and the inferencing mechanism) on the types of ground data available and the types of inferences the representation is to support efficiently. Knowledge of these representation choices and how one can be mapped into the other allows the system to be data-driven in its representation choice.

Application Specific Meta-Constraints

Just as the ontological engineers are guided by the availability of ground data in their representation choices, so are there meta-constraints that are specific to the application domain that the system can exploit when changing its representation.

Infrastructure for Performing a Representational Switch

Such infrastructure allows the system to analyze the categories of data types available in the data sources; to perform the meta-reasoning to select the next appropriate problem representation; and, to rewrite the existing domain knowledge and background representation into a form optimized for the input.

Once such an infrastructure is presented with the semantics for specific data sources [Masters 2002], it can then select appropriate representation choices for the data at hand, transform its domain and background knowledge in a (potentially lossy) fashion into a data-optimized or at least data-focused representation. Application-domain specific heuristics will either cash out as additional adjustments to this transformation process or as inference heuristics, both for deduction and abduction, for reasoning in the application domain.

Choices for Representation Change

Ontology and Methodology

In order to be able to think usefully about representation change in the context of pattern-to-data bridging, it is first necessary to know what kind of transformations are available. Not surprisingly, providing an exhaustive list of ontological representation choices akin to the patterns of the software engineering community [Gamma et al. 1994] is clearly outside of the scope of this paper. We will therefore focus on a few exemplars that have the property that they are relevant to the domains of level-two-three information fusion, threat scenario recognition and situation-based event extraction.

Most of the representations described below trade expressivity for complexity, which translates directly into important influences on computational tractability, such as overall size of the knowledge store, branching factors, etc.

Events vs. Action Predicates

One of the best-known knowledge representation choices is the one between scripted event types and action predicates, also known as Davidsonian and non-Davidsonian representation.

Davidsonian Representation

Describing events as instances of scripted event types is often referred to as Davidsonian Representation (Davidson 1963), where Davidson proposes to reify the temporal situation of an action in its own right, and then represent

additional event details as binary relations about that reified situation.
A statement such as "On the Ides of March of 43 BCE, Brutus stabbed Caesar with a dagger." is readily translatable into one assassination event in which Brutus is the aggressor, Caesar the victim, the date of the event the Ides of March of the year 43 BCE, and the murder weapon the dagger.
Such a representation is very flexible. It handles unequal information about events such as multiple assassins, multiple murder weapons, or even missing information (the technique is most effective when combined with a hierarchy of scripts [Schank & Abelson, 1977] and actor roles).
In addition, Davidsonian representations are attractive because all relevant temporal qualification can be attached to events as individuals in the form of starting dates and stopping dates; there is no necessity for context-bound temporal qualification of the sentences.

Action Predicates

The alternative to reifying events and situations as first-order objects is to attempt to handle them relationally via predicates that embody actions. Thus, for example, the fact that Brutus kills Caesar might be represented as a two-term relation between Brutus and Caesar. This approach eschews reifying an actual situation or event at all, thereby sidestepping the problem of event substructure. However, the very fact that the event has been dispensed with entails that temporal qualification must attach at the level of the ground formula, which is a far more expensive recourse. In addition, the relation-based approach features the possibility of information loss unless additional support vocabulary is also defined.
For example, a meeting in which five people participated *could* be modeled with 10 pairwise meeting links between every possible pairing of the 5 participants, with identical temporal and spatial qualifications applying to each link. However, if these qualifications are sufficiently coarse-grained (say, 'the week of September 10, 2007' and 'in New York City'), then the pairwise linking as represented would be perfectly consistent with Davidsonian models wherein more than one meeting event occurred, and indeed, with models in which all five of the participants never met together.

Options for Temporal Qualification

Closely related in practice to the question of whether or not to reify situations is the question of whether the ontology proposes to deal with individuals as temporally extended, four dimensional things with temporal parts whose attributes are not subject to change, or as three-dimensional objects whose attribute sets are subject to change with time, and which may even have incommensurate attribute sets at different times, or whether the ontology will attempt to accommodate both these sorts of thing.
The computational trade-off here is reminiscent of what attaches to the Davidsonian question: four dimensional entities as a rule greatly simplify the temporal qualification question because such qualifications as are needed attach to individuals, not propositions. However, from a representational standpoint the fact remains that the 4-D view seems genuinely counterintuitive with respect to some classes of objects, notably those we are accustomed to thinking of as playing roles in events: persons, artifacts, locations, and the like. Moreover, insisting upon either an exclusively 4-D or exclusively 3-D approach means that it will be impossible to distinguish between a 3-D object and its corresponding four-dimensional space-time 'trace', e.g., between Winston Churchill and Winston Churchill's lifetime.
Although metaphysicians like Sider have argued strongly for the ultimate incoherence of the 3-D view [Sider, 2001], we believe that CYC has achieved a representation which incorporates a provisional version of the 3-D model within a general 4-D approach with no ill effects, and in a fashion that allows us to accommodate ontologies predicated on both models.

Mereology and Sets

The 3-D/4-D debate is one example of an ontological dichotomy that repeats itself at many choice points in ontological engineering, namely, the choice between partonomy and set theory. The four dimensional approach is an inherently partonomic one: changes in attributes are handled by assigning temporal parts to a four dimensional object, and having the attributes hold, timelessly, for those parts—something that the three dimensional view handles in its turn by assigning the same object to different sets at different times.
An analogous choice presents itself in dealing with substances, some mereologists having argued that at least some substances are better represented as parts of a common whole, as opposed to elements of a common class.

Agentive Characterization

Representing behaviors requires identifying who it is that is taking actions and therefore exhibiting behavior. At the level of detail of individuals, it is usually appropriate to ascribe behaviors and actions to the individual. However, this use becomes increasingly more metaphorical as one talks about legal agents and voluntary associations, be they state or non-state actors, business or geo-spatial abstractions. The epitome of this mode of knowledge representation is the political historiography of the late 19th century (nowadays often derided as "Great Man" historiography), where Louis XIV is France and Bismarck is Germany.
Such an abstract level of knowledge representation is powerful in its simplicity, and appropriate for many domains. In diplomacy, for example, governments intend to appear as "in-dividual" in the strict sense of the word. Businesses work hard to project a corporate identity and identify persons who are permitted to make public statements on behalf of the corporate.

Yet, even powerful simplifications can be too simple. The separation of powers in Western democracies, for example, is intended to ensure that the overall political action of the state emerges from the inputs of the individual powers and their constituent parties. Equally, nefarious groups are not homogeneous or of one opinion either: 9/11 hijacker Ziad Jarrahi at one point threatened to withdraw from the scheme because of tensions amongst the plotters.

As a concrete implementation example, the Cyc ontology supports two ways of thinking about countries: One that treats the government of countries and their territories as interchangeable, and one that treats governments and territories of countries as distinct.

Data-Side Transformations

The foregoing discussion has focused primarily on transformations on the side of the ontology, in background knowledge; another important option available is the transformation of the available data into a more conducive format via inference. There is good reason to believe that data-side transformations could afford the most ready means of adding needed features in the short run.

To take a simple and somewhat artificial example, suppose that we are attempting to corroborate a hypothesis that references a relation, 'nephewOf' in a database wherein this relation is not populated, but where the relations 'siblingOf' and 'sonOf' are. Rather producing an equivalent, nephew-free hypothesis in the background knowledge, it might be more expedient to use Cyc inference to propagate the definition of 'nephewOf' into the data, finding, in effect, all of the implicit 'nephewOf' relations that exist there.

Of course, 'nephewOf' might not be a part of the database schema to begin with, in which case more radical surgery would be required. Here is must suffice to say we have thought about how to effect this; the exigencies involved in having the Cyc system access an external structured knowledge source for purposes of transformation are beyond the scope of this paper.

The caution with data transformation is that many of the modifications effected are likely to be irreversible, and, interestingly, sometimes even when the definitions supporting the transformation are bi-conditional. Thus, while data-side transformation can be a powerful technique for compiling information in ways that square with background knowledge, decompiling what one has compiled may not always be practicable. However, since our design treats such transformations as one-time actions at system deployment time, under our constraints, this issue is negligible.

Domain-Specific Constraints

Methodology and Infrastructure

A representation language that is expressive enough to address potential demands for ontological reformulation is a necessary, but not a sufficient condition for inter-operability. In itself, it does little to address the worry that the combinatorial explosion of the representation choices will effectively prevent the detection of any scenarios. Here is where context-specific meta-constraints enter the picture: Context-specific because they are closely tied to the ground-level data at hand as well as exploitative of simplifying assumptions that might be inappropriate in other use-cases, together with the essential concept of *data profiling*.

Ground level data sources come with a predefined set of semantic relations. Databases provide specific information about the entities represented and no other. Document extraction systems can recognize specific types of relations and entities. These combine to form a minimal set of representations that any proofs, no matter what their interior representation choices, will have to bottom out in. At the same time, the categories of entities and the kinds of information for which data sets and extraction systems are available in principle are highly limited compared to the expressiveness of formal representation languages.

These constraints imply that there are representations that the system could construct in principle that could never be used to find a proof given the provided data.

Consider a database of Mob-financed contract killings that contains no information about the weapon used for the hit. Any proof path that required such information to be available would simply not succeed in the described configuration. Consequently, any representation adjustments that introduced such a dependency can be avoided from the get-go, pruning the search through representation space.

Besides data observability, we also needs must distinguish discriminability as a relevant control for scenario generation: The extent to which a feature that is a candidate hypothesis serves to distinguish the type that is being sought. This is in part derived from data characteristics, but only in part, and should properly be distinguished from what is strictly observable.

Thus, to take the contract killing case again, it might be that an available database might consist in human intelligence reports of movements of suspicious persons, and it might be that such a database might include many records of meetings between individuals. Strict observability is not at issue here: the category 'meeting' is reflected in one fashion or another in the database schema, and there is no want of meeting records in the database. And indeed, at least one plausible scenario for 'murder-for-hire' includes a meeting feature: meetings are plausible ways of hiring a hitman.

However, considered as a discriminator for contract killings, meetings alone are not terribly useful: the conditional probability of a contract kill is not greatly increased by the observation of a meeting, or to put it another way, the vast majority of meetings observed in our putative database are likely to have nothing whatever to do with contract killings.

This gives us a prima facia reason for deprecating meetings in hypothesis generation relative to the contract killing case, unless of course there is a more compelling discriminator that includes meeting as a component – perhaps a meeting between two suspect individuals, followed, within a certain period of time, by a suspicious death of one of those individuals.

Probability plays an evident role in these considerations, and is itself a factor in hypothesis generation. Considerations of discriminability and observability aside, we may say that generally, it is a bad idea to postulate wildly improbable scenarios. However, this must be qualified with the observation that plausibility cannot be the sole criterion, especially in threat detection domains where there is a willful effort to mislead or obfuscate on the part of an intelligent adversary. Plausibility is closely correlated with what the analyst expects, hence the advantage rests with the threat force that is able to find a statistically improbable but still workable means for attaining its objective.

Plans as Constraints in Threat Detection

Capturing application specific domain constraints is primarily relevant for the design of the representation that controls representational switch. In threat analysis, for example, numerous constraints derive from the fact that the hypothesis-driven recognition targets plans, which have properties that the recognition system can exploit heuristically when searching for an appropriate implementation representation.

We identify three dimensions of proximity along which the search through representation change can be optimized:

1. Finding an impact-equivalent plan

Often there is a generalization of an attack plan that, if prevented, is sufficient to foil many more specific attacks. Consider Operation Bojinka, where the liquid explosives were to be smuggled onto the planes using bottles for contact lens cleaning fluid. It is doubtful whether much is gained for a scenario recognition system to iterate through its representation of bottled liquids to identify possible candidates for smuggling explosives onto planes inconspicuously; the realization that suspect actors are working toward liquid explosives should be sufficient for the system to alert its superiors.

2. Finding the minimally detectable plan

As mentioned above, not all phases of an attack are observable or likely to be reflected in the available data sources. Therefore, there is no point for the scenario recognition system to look for a plan that it cannot distinguish from another plan, because the kinds of data needed to differentiate them is not available. By restricting the target representation to those attack plans that can actually be corroborated, the scenario recognition system can avoid cute and clever plans that are, from its point of view, not provable.

3. Finding the simplest executable plan

During the planning for Operation Bojinka, Khalid Shaikh Mohammed had originally wanted to blow up eleven airplanes bound for the United States using liquid explosives. There are some indications that when he reworked Operation Bojinka into the 9/11 attacks, he also envisioned the use of eleven airplanes. Eventually, that number was whittled down to four. However, it might have been a sufficient contribution of a scenario recognition system to uncover just one of the kidnapping teams in order to prevent all 9/11 attacks. Therefore, going for the simplest executable plan first has advantages both for the representation space search and for the plausibility of the scenarios detected.

Transforming Knowledge Representations

Heuristics for scenario generation are well and good, but what they implicate, namely, a declarative and quantitative infrastructure for implementation, is still in the very early stages of being implemented at Cycorp. We assume that earlier work in knowledge exportation, which involves representation change of a different kind, will provide initial guidance [Reed & Lenat 2002, Cabral et. al. 2005].

Applying Domain-Specific Restrictions

The key insight about domain-specific restrictions is that knowledge about the lack of observability of an entity can be used to block looking for the entity. In a sense, an entity of the appropriate type is abduced once and from then on silently reused.

Consider the example of unobservable communication: Terrorists might communicate through a wide variety of signaling means that leave no trace in the data sources analysts typically have access to. Instead of wasting time abducing ever more complex yet unverifiable ways in which the communication might have taken place, a stub unobservable communication event can be abduced to complete the inference in the most cost-effective manner.

Doing so can cut down significantly on the complexity of abduction, which (due to its lack of focus) is even more explosive than deduction.

We expect the key benefits to come from script and scene simplification on the one hand side and from class-graph collapse on the other hand. These benefits translate directly into a reduction of the overall search space.

Exploiting the Deployment Context

With the installation of a threat scenario recognition system, the decision over which types of formulas can be bound against the databases is finally determined. Consequently, the knowledge transformation process can begin to trim down the ontology to the terms needed.

The basic insight is that distinctions that are not reflected in the instance level of the data sources are without a difference for the purposes of threat scenario recognition. For example, classes that have no members without the database (including no stand-in for an unobservable member; see the discussion above) will not have any members if the database has none. Such classes (and all of their subsumed classes) might as well be pruned from the abductive reasoning step. A similar argument applies to specializations of relations.

Once classes and relations have been reduced to the data-backed level of detail, all rules that require one of the pruned classes or relations can be analyzed as well. Rules that have either members of the pruned classes or relations in their antecedent can be pruned as well (so there is no rule that establishes the relation or the membership via the consequent).

Because of the web-like connectivity of knowledge, we envision a multiple-pass approach, repeatedly traversing suspected pruning candidates, until a pass without progress is made. This avoids the need for computing the dependency graph up front.

Conclusions

It is an old adage in computer science that the introduction of abstraction removes problems at the cost of efficiency, while the elimination of abstraction provides speed benefits at the cost of maintenance. Separating the appropriate description language from the implementation language is a technique dating back at least to Grace Hopper.

The design of the system described here exploits this insight by allowing a flexible description of nefarious schemes that is then in two passes whittled down to the actually observable and eventually to the actually provable. The focus this provides in terms of deduction and abduction and their concomitant computational complexities is well worth the one-time cost of tightening the knowledge representation to the situation at hand.

Ongoing Research Directions

One danger of abducing reusable, drill-down blocking-entities is that they could posit a link where full deductive inference would have proven the absence of a link. For example, if domain knowledge suggests that the transport of a liquid explosive in a small container will be unobservable, we still would not want to abduce properties for the unobservable event that are impossible (e.g. moving the explosive from the staging area to the target at super-sonic speeds). Thus, the abduced entities cannot be complete blanks, but must preserve the bounds of the possible.

Also, it is possible that the separation of the domain-specific and the deployment-specific knowledge transformation turns out to be unnecessary in practice. If the deployment-specific knowledge transformations prune the same things that the domain-specific transformations would have (which effectively form the set of all "suspected" deployment contexts), the effort involved of two pruning passes may not be justified.

Furthermore, while the top-down approach is a sensible first pass, eventually a feedback loop of some form or another from the data to the higher levels should provide superior performance. As distinctions appear in the data that are not supported by the representation, these should move upstream. It is worth noting that such improvements merely extend our approach. The benefits from the top-down flow of control are that representational opportunities identified at the very specific data level can be lifted to the appropriate level of generality and from thence flow down into different deployment contexts or even domains, as appropriate.

Finally, we confront issues when bridging from a model with a Davidsonian treatment of events to data that bottoms out in action predicates. Since the events have to be hypothesized, their equivalence criteria are difficult to determine, as described above in the case of the multi-person meetings in New York. Even assuming reasonably sound and precise inter-definition, there remain questions of how to handle pigeonholing. The strong temptation is to let some combination of Occam's razor and the model settle the matter, but that takes the risk of over-fitting the data. Leveraging social, contextual, and temporal confinement knowledge for purposes of event reconstruction remains an open research issue.

Thinking "Outside the Box"

As described in the introduction, the more topically focused the ontology and scenario-generation methodology used in threat recognition, the more computationally tractable is the process. We have described methodologies that we believe will be useful in achieving tractability.

However, there appears to be fundamental tension implied by our approach, one that is of particular relevance to threat detection. As intelligence community analysts well know, preempting hostile attacks requires what is loosely called "thinking outside of the box", that is, imagining novel ways in which the nefarious schemes of the non-state actors could be achieved. Thus, analysts, whether human or machine-implemented, are asked to both optimize their knowledge representation and to think as wide ranging as possible

We believe that this comparison misses a decisive difference. For the system that we have described here, thinking "outside of the box" is primarily a feature of the representation side. Indeed, we would want our system to be as "creative" as possible in coming up with ways of combining evidence, as long as they all bottom out in the types of evidence that are actually available in the databases the system is integrated with.

This restriction does not apply for intelligence analysts, who are asked to conceive of novel threats for the precise purpose of identifying new categories of data that should be gathered.

References

Bostwick, B., Hunter, D., Pioch, N. 2006. CADRE: A System for Reasoning over Very Large Datasets. *Capturing and Using Patterns for Evidence Detection: AAAI Fall Symposium Technical Report FS-06-02.* AAAI Press.

Cabral, J., Kahlert, R.C., Matuszek, C., Witbrock, M., Summers, B. August 2005. *Converting Semantic Meta-Knowledge into Inductive Bias.* In: *Proceedings of the 15th International Conference on Inductive Logic Programming, Bonn, Germany, August 2005*, Springer-Verlag.

Clarkson, A., Kaisler, S., Lenat, D., Oresky, C. 1991. Strategic Automatic Discovery System (STRADS). *Knowledge-Based Simulations.* Springer-Verlag.

Davenport, D. and Hill, R. 2006. Fast Abductive Reasoning over Ontologies. *Capturing and Using Patterns for Evidence Detection: AAAI Fall Symposium Technical Report FS-06-02.* AAAI Press.

Davidson, D, 1963. *Actions, Reasons, and Causes*. In: *The Journal of Philosophy*, Vol. 60, No. 23, 685-700. Nov. 7, 1963.

Hawkins, J. 2004. *On Intelligence.* Times Books.

Gamma, E., Helm, R., Johnsson, R., Vlissides, J. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Kahlert, R. and Rode, B. Tracking Quantity Fluctuations Using STT. *Capturing and Using Patterns for Evidence Detection: AAAI Fall Symposium Technical Report FS-06-02.* AAAI Press.

Lenat, D. 1976. A.M.: An Artifical Intelligence Approach to Discovery in Mathematics as Heuristic Search. Stanford Artificial Intelligence Laboratory Memo AIM-286.

Lenat, D. and Shepard, B. Cycorp Scenario Generation Final Report. Government Contract Number F306-00-0160 DARPA

Masters, J. July 2002. *Structured Knowledge Source Integration and its applications to information fusion*. In: *Proceedings of the Fifth International Conference on Information Fusion*, Annapolis, MD, July 2002.

Raghavan, S. 1991. KNOB: a mutation-based approach for generating stereotype behaviors in active decision support systems. *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences.* vol. iii, issue 8-11, pp. 417-429.

Reed, S. and Lenat, D.B. 2002. *Mapping Ontologies into Cyc*. In: *AAAI 2002 Conference Workshop on Ontologies For The Semantic Web*, Edmonton, Canada, July 2002.

Rode, B. 2006. Strategies for Hypothesis-Driven Recognition in Rule-Based Systems. *Capturing and Using Patterns for Evidence Detection: AAAI Fall Symposium Technical Report FS-06-02.* AAAI Press.

Sider, T. 2001. *Four Dimensionalism: An Ontology of Persistence and Time*. Clarendon Press.

Schank, R. and Abelson, R. 1977, *Scripts, Plans, Goals and Understanding: An Inquiry Into Human Knowledge Structures*, Yale University Press.