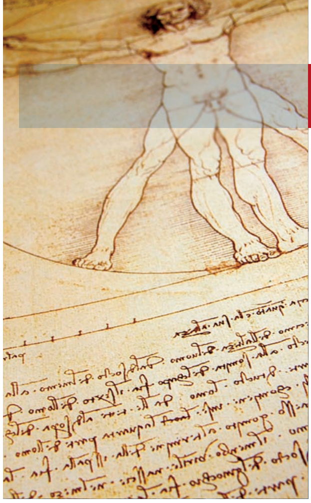

Editors: **Robert R. Hoffman, Jeffrey M. Bradshaw, and Kenneth M. Ford,**
Florida Institute for Human and Machine Cognition; rhoffman@ihmc.us

Ontology Creation as a Sensemaking Activity

Thomas C. Eskridge and Robert Hoffman, *Florida Institute for Human and Machine Cognition*

Sensemaking is a cyclical discovery process in which we use data to form an understanding, and this in turn influences what counts as data. Concurrently, we evaluate the plausibility, consistency, and coverage of our explanations. This helps us anticipate difficulties. We notice and try to explain apparent anomalies, or we decide we need to completely reframe our understanding.[1–3] Sensemaking is more than an analytical choice among competing explanations expressed in some fixed problem space. It is richer and more complicated than any linear description as "steps" or "stages." It is a process of deliberating over alternative plausible explanations while at the same time having our explanations guide the exploration of information. Furthermore, sensemaking is often a social activity that promotes the achievement and maintenance of common ground.[4]

Research has shown how diagramming can be a useful tool to support the sensemaking process.[5,6] Research has also shown how diagramming can be a useful tool for constructing, editing, and understanding ontologies.[7,8] The meaningful integration and comprehension of data as part of the sensemaking process can benefit from the development of ontologies that establish the underlying semantics of that data and then develop semantics for new concepts and scenarios. The emerging Semantic Web technologies address this challenge by developing an "open network" of formally described concepts, in which the publication of a formal ontology lets other users reuse concepts when describing new knowledge.

In this article we pursue the idea of diagramming to support simultaneous sensemaking and ontology development. We argue that tools can be developed to support a continuum of knowledge model development, from initial free-form diagrams that make it easy to capture an author's knowledge and intent but are only interpretable by humans, to propositionally coherent concept maps that structure the free-form diagrams into independently meaningful node-link-node triples, and finally to description-logic concept maps that combine human readability and understanding with logical formalism and a machine-understandable format. Although this idea is not new, the widespread availability and popularity of semantic technologies for a variety of tasks makes it especially pertinent.

Propositional Diagrams

The analysis of a topic by creating a propositional network can improve topic comprehension, relative to alternative representations such as text, hypertext, or bullet lists in presentation slides.[9–11] Research has shown how propositional diagrams can be used in individual and collaborative problem-solving.[12–14] One powerful tool for helping people make sense of situations is *concept mapping*. Concept maps are expressive diagrams used in the propositional representation of knowledge, spanning from educational contexts[15] to the critical thinking of intelligence analysts.[16–18]

Figure 1 shows an example of a concept map, representing an individual's attempt to integrate knowledge about the Bihari refugee situation in Pakistan.[19] This example is one from a set of 15 diagrams that represent an analysis of the Bihari situation; this one is about why the Bhiari do not qualify as refugees under international law. The icons beneath two of the nodes indicate hyperlinks to text pieces and URLs that present supporting evidence.

This concept map fulfills some requirements for a knowledge representation that will assist the sensemaking process. It expresses and organizes the concepts in a way that is meaningful to humans and presents specifics, in this case the conditions that must be met for a group to be considered refugees. It makes explicit the concepts that are involved

1541-1672/12/$31.00 © 2012 IEEE
Published by the IEEE Computer Society

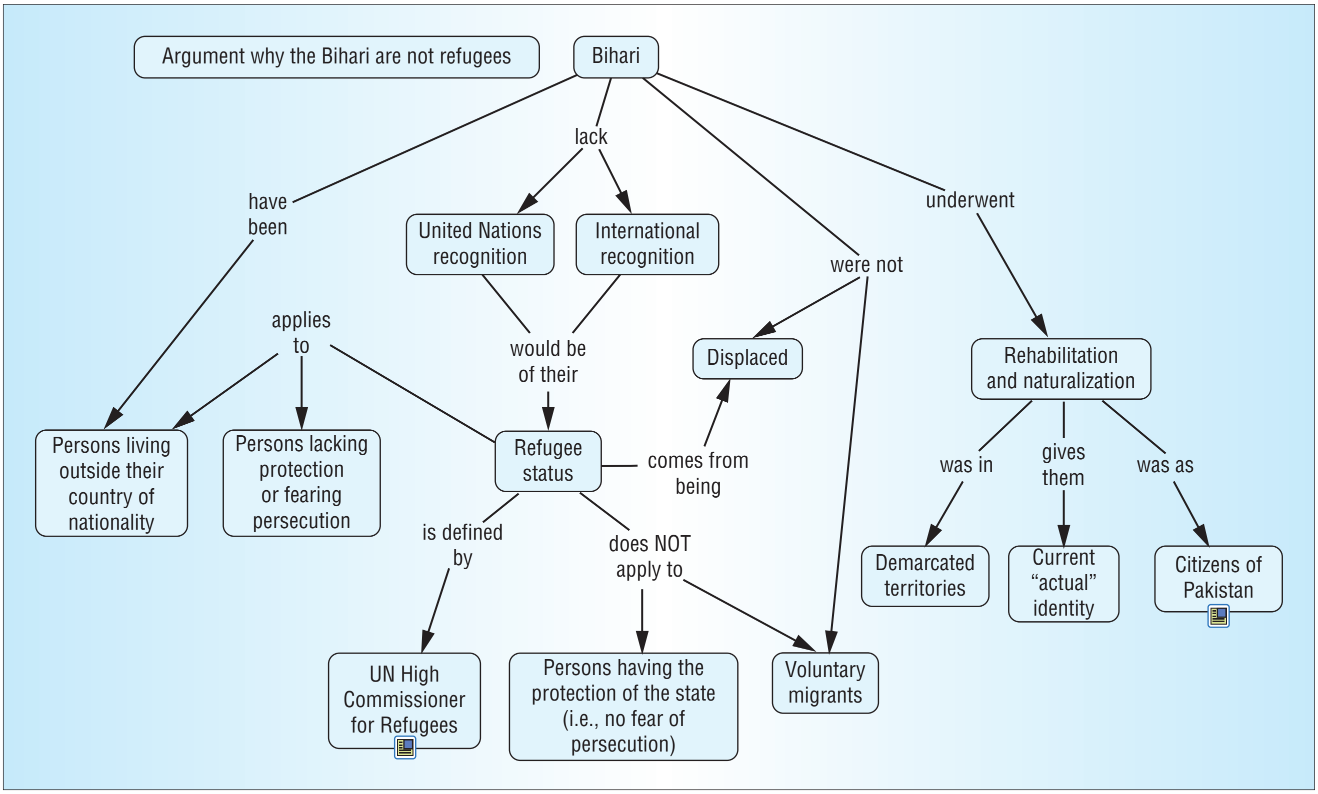

Figure 1. A concept map developed as part of an individual's knowledge modeling effort. The map integrates knowledge about the Bihari refugee situation in Pakistan.

in an answer to the main question posed by the concept map ("Why do the Bihari not qualify as refugees?"). Finally, it is argumentative in that it makes the contradicting points of view explicit (concerning the "actual" identity of the Bihari).

A method to make such meaningful diagrams computable—that is, amenable to the operations of logic—would be helpful. Doing so would make the information expressed in the diagram available not only for people to search, reference, and expand, but also for machines to operate on. Arguably, achieving this transformation without human input would require solving significant problems of inference and semantic alignment of concepts. Until recently, the closest that anyone has come to making concept maps computable (with or without human input) had been to make them propositionally coherent. In this article, we discuss a way in which meaningful diagrams, used as a part of the active effort to make sense of situations, might also feed into computational analysis, and vice versa. This would be diagramming as a method for active construction of a representation, an activity that helps the human-machine work system make sense of things.

Man Versus Machine

In his discussion of representation, Allen Newell defined "the knowledge level" as composed of agents that possess knowledge, have goals, and act rationally so as to achieve the stated goals.[20] He argued that logics were appropriate as approximate descriptions of knowledge, or appropriate as tools for the analysis of knowledge: "Given a representation—e.g., a semantic net, a data structure, a symbolic structure for some abstract problem space, a program, or whatever—to determine exactly what knowledge is in the representation and to characterize it requires logic." Newell said that natural language is not adequate for the analysis of knowledge because its complete structure and rules of inference have not been externalized. But his notion of "adequacy" refers only to analysis by the machine. Humans have a requirement too: to make sense of things precisely at the knowledge level, which would of course be reliant on a natural symbol system (in other words, natural language).

We do not contest Newell's insights concerning the definition of "levels" of representation and the relations of abstraction and reduction among those levels. We do wonder, however, what would happen if intelligence were considered with reference to the human-machine relationship as a work system. In this case, there must be some means of bridging the sensemaking gap between what computers and people understand through the use of so-called "mediating representations."[21–23]

What if a representation were to be both? In this article, we address the assumption that there must be a bifurcation between the artifacts used by people to express and share meanings versus the artifacts that machines require because they feed directly into computational procedures. These need not be separate; the twain *can* meet. The notion we present is a method for computation over representations that are also meaningful to humans who may not fully understand the machine. The goal is to support sensemaking by both humans and machines, in reference to what humans can do and what machines can do rather than by reference to what either of them cannot do.

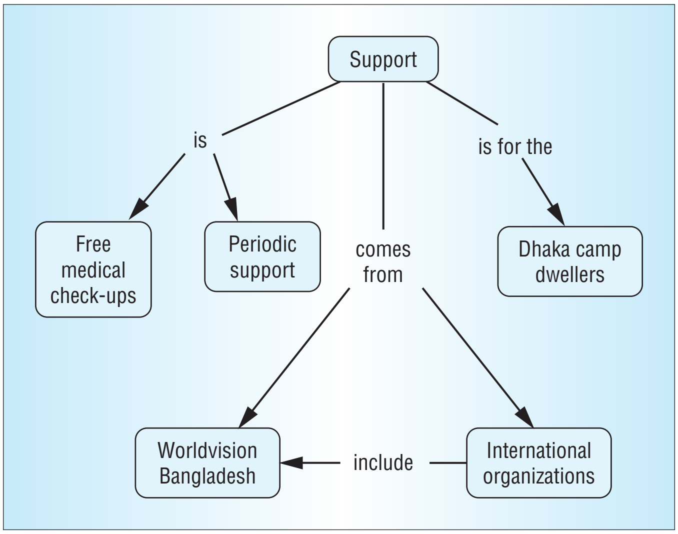

Figure 2. A concept map segment illustrating propositional coherence. Each node-link-node triple expresses a proposition that can be meaningfully read independently from other map elements.

Building a Continuum

We begin with the sentence, "Periodic support of free medical checkups for the Dhaka camp dwellers comes from international organizations such as Worldvision Bangladesh" (taken from the article "The Neglected Stateless Bihari Community in Bangladesh: Victims of Political and Diplomatic Onslaught" by Kazi Farzana).[19] The sentence is easy to comprehend and captures the thoughts or beliefs about the world held by the author. Figure 2 shows the propositional network for this sentence. Nodes represent individual concepts, and links represent relations. The propositions unpack some of the relations that are expressed through the natural syntax. The diagram is propositionally coherent because each node-relation-node triple can be read as a stand-alone proposition.

But this representation is still not as specific as it could be. Does the humanitarian aid extend to other medical checkups that may not be free? The modifier "free" would need to be unpacked from this representation, allowing "medical checkups" to be divided into "free" and "not free." Are there organizations involved in addition to Worldvision? And so forth.

Constraining Concept Map Semantics

Machines can manipulate logics in many ways, and complete inference mechanisms are available that can draw conclusions and check for internal consistency over a set of propositions. However, these logics usually have a more complex syntax than the node-link-node triples of propositionally coherent networks (as in Figure 2) and can require significant training to construct and interpret properly.

Work during the last decade has led to the idea of isolating useful patterns of quantification that can be identified with operators on concepts, and these patterns have been formalized in what are called *description logics*.[24] The basic idea of description logic is to isolate these logical patterns and redefine them as operators on a logical vocabulary—operators such as *union*, *intersection*, and *complement*. These operators can then be used to form new relations. We can use them to make our diagram more specific and accurate. Consider our previous problem of unpacking "free medical checkups." We can expand this by noting that it is the intersection of the concepts of "medical checkups" and "free services." By linking it also to the concept "periodic support" we can fully capture the author's intent. Modifying the Figure 2 diagram to reflect this thinking, we arrive at a diagram shown in Figure 3.

The original motivation for description logics was to make deduction and consistency checking more computationally tractable. For our purposes, the description-logic style it relates naturally to the concept map notation. Since we can unpack the definitions, this constitutes an extension to the notion of propositional coherence. We can also use the operator patterns directly as templates to modify existing concept map structures or provide the skeleton for a new structure.

Restrictions are another type of operator on concepts. Restrictions provide a general way to define new categories (or classes) in terms of their relationship to other meanings. Consider how the concept "Dhaka camp dwellers" might be formally defined. Members of the set `Dhaka camp dwellers` would be those people that reside in the Dhaka camp. This is a simple value restriction, `residesIn`, on the property of `people`. This property needs a formal definition, including specifying its domain (including "people") and range (some locations). Fortunately, we can consult other ontologies and reuse their concept and property definitions. Indeed, in order to be fully effective, description logic diagrams must have many conceptual connections among their components. Publicly available ontologies form a "voluntary syndication" of knowledge composition: a distributed network of linked mini-ontologies integrated into a connected web by their reuse of concepts.

In a concept map constrained by the semantics of description logic, the node-link-node triples express relationships describing (and, in effect, defining) categories. Such diagrams retain propositional coherence in the sense that all of the triples in a diagram are still meaningful when read in isolation. Figure 3 paraphrases the sentence in a way that seems slightly roundabout relative to the original natural language. However, the diagram still makes sense, with only minimal explanation, to the human interpreter.

Having incorporated description logic semantics as a means to structure relations among concepts, we still have not completely leveraged the power of computational reasoning. We need to take the diagram a full step further. We must add the capability to create new classes, using description logic operator templates so that they may be exported, understood, and reasoned about as full ontological content.

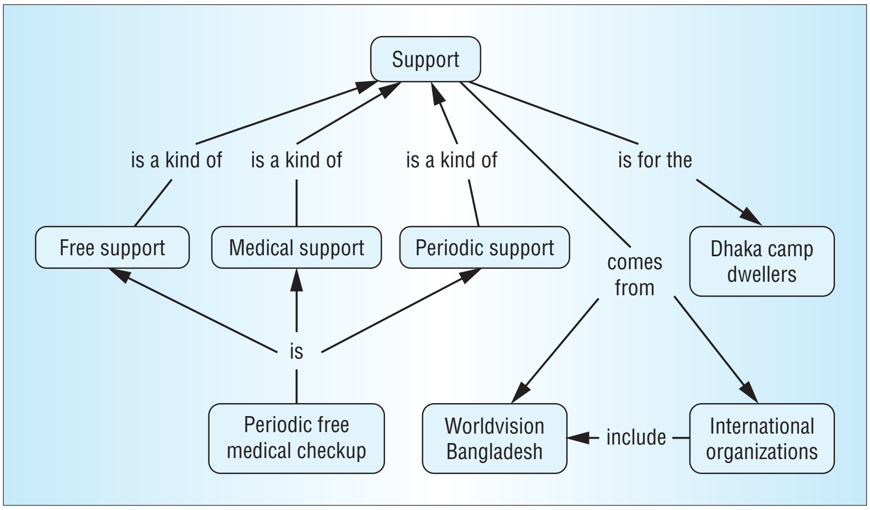

Figure 3. A step on the way to description logic. The propositions decompose the concept of "medical checkups."

The OWL Formalism

The CmapTools Ontology Editor (COE) is a version of CmapTools that uses a description logic style to encode meanings. It was originally intended for the World Wide Web Consortium–recommended Web Ontology Language (OWL) formalism of the Semantic Web.[25,26] OWL is a fully formal notation with exactly specified semantics, designed independently of concept maps expressly for inference engines to be able to process.

Figure 4 shows a description logic diagram generated using the COE tool. It can be exported into OWL for use by other Semantic Web technologies or other ontology authors, and it can be considered to be an alternative syntax for roughly 50 lines of fairly tedious XML. The diagrammatic representation attempts to show what OWL developers are most likely to want to see, and hide those aspects that often make such diagrams unintelligible—while still allowing people to drill down into details as needed.

Space does not allow a detailed description of all the COE conventions that are used in Figure 4.[27] Some general aspects of the design are important, however. Properties are always links, and classes and individuals are nodes. We use color—very selectively—to indicate the property type: black for restrictions, red for unions or intersections, and blue for subclasses. The shape of the node is also important: rectangles represent individuals, or specific members of classes, which are represented by rounded-corner rectangles. The text in the diagram often has a prefix associated with it, separated by a colon. This is called a *qualified name*, or "qname" for short. For example, `po1:Organizations` instructs reasoners to access the ontology available at the URL specified by the nickname `po1` and obtain its definition of the concept `Organizations`. If a concept or property does not have a qname associated with it, it is defined in the ontology under construction. We indicate properties of a property (that it is transitive, such as `ancestorOf`, or symmetric, such as `brother`) by additional tags on the link label, and indicate OWL restrictions on a property by attaching predefined phrases (such as `must be exactly`, `can be`, `exactly one`, `at least 3`) to the property name as part of the link label. This link/node conceptual discipline is notably absent from the XML syntax for OWL, which treats all entities similarly.

Rendering OWL into concept map notation may be regarded as a significant marker in the territory that lies toward the *formal* end of a continuum of human-machine functionality. The step has elimnated ambiguities

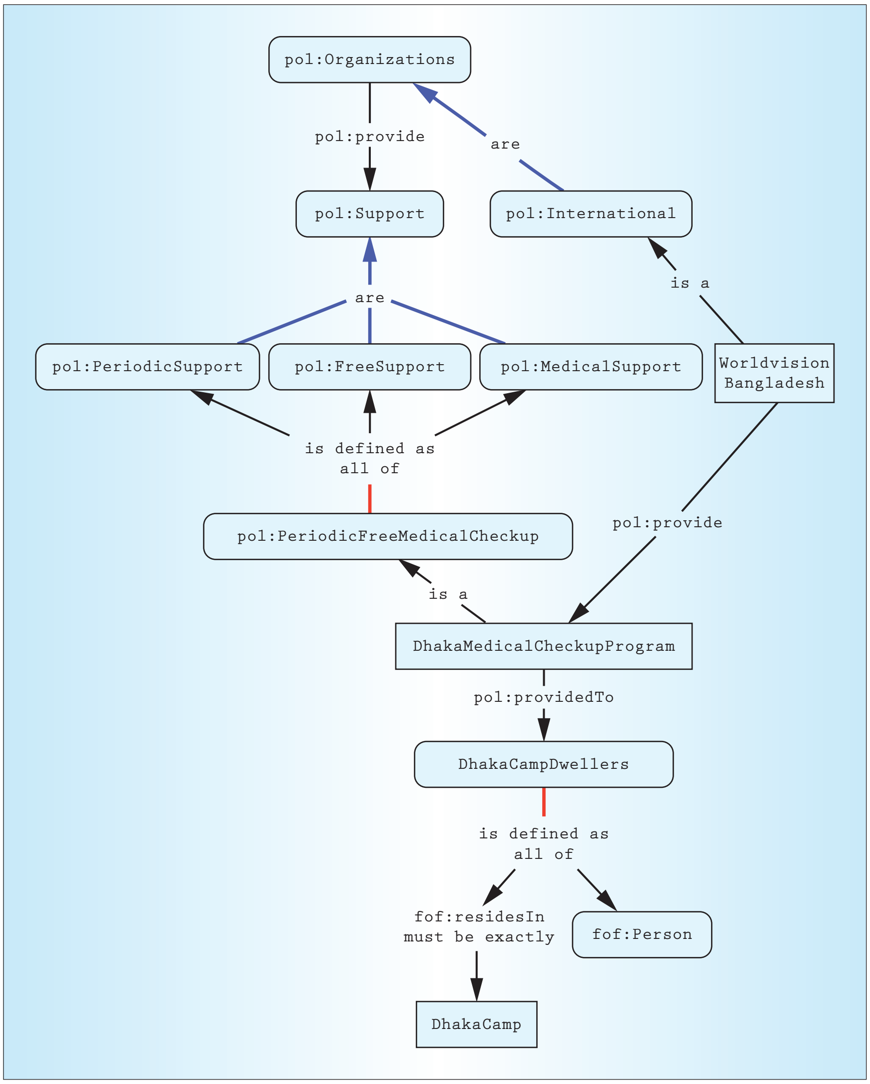

Figure 4. A fully specified description-logic concept map. It refines the concept map from Figure 3 to reference published ontologies for existing concepts and to provide formal definitions for "DhakaCampDwellers" and "PeriodicFreeMedicalCheckups."

that make computing over unconstrained concept maps difficult, though not without losing some nuance. There still remains something of a gap between the complexity of OWL-formalizable concept maps and the naturalness of the concept maps that are constructed by humans when they are free to express themselves intuitively. Moreover, OWL is by no means the final word in formalized content; it is already being extended to other forms of content such as if-then rules, logic programming notations, and Petri nets describing processes.

Some OWL constructions do not transcribe into anything remotely resembling natural language and so are rendered in COE using graphical and labeling conventions. As a side effect, we have found that importing OWL ontologies into COE often makes their essential nature quickly apparent. For example, it is easy to distinguish ontologies that are largely taxonomic, since COE displays the subclass links with a distinctive blue color and connects them together into a tree or graph. Ontologies that are less concerned with classes and more concerned with properties have many dotted links, arising from COE's display of domain and range information. COE's visual layout also draws attention to "missing" information, which is regrettably common in OWL formalizations, such as unspecified class names or missing domain and range information.

The CmapTools software allows for search, rapid navigation, and image zooming through large ontology maps using text-based matching on concept names. We have found that the conceptualization of an ontology as something displayed on a surface and the associated metaphors of "moving" and "looking more closely" are powerful user aids in organizing and comprehending large bodies of information.

Revisiting our example, we can migrate entities designated in the description logic for reasoning by a computer, making many of the complex terms used in the argument explicit and therefore more understandable to humans. To do this, we pull the class definitions of various entities from available ontologies or from ones we create ourselves to meet the specific needs of the map. Figure 5 shows a section of a larger ontology that contains some needed concepts along with some concepts that were defined specifically for this example.

By pulling information from other ontologies, we gain the benefits of reusing existing terminology as well as the ability to have our resulting ontology be interpretable by a wider variety of applications. In Figure 5, we provide a term that classifies a person that is a resident of one country who has nationality in a different country as an `expatriate`. Because the components of this term are drawn from the dbPedia ontology, other applications that use this ontology can import our term and use it directly.

Applying the terms found in other ontologies and those we constructed

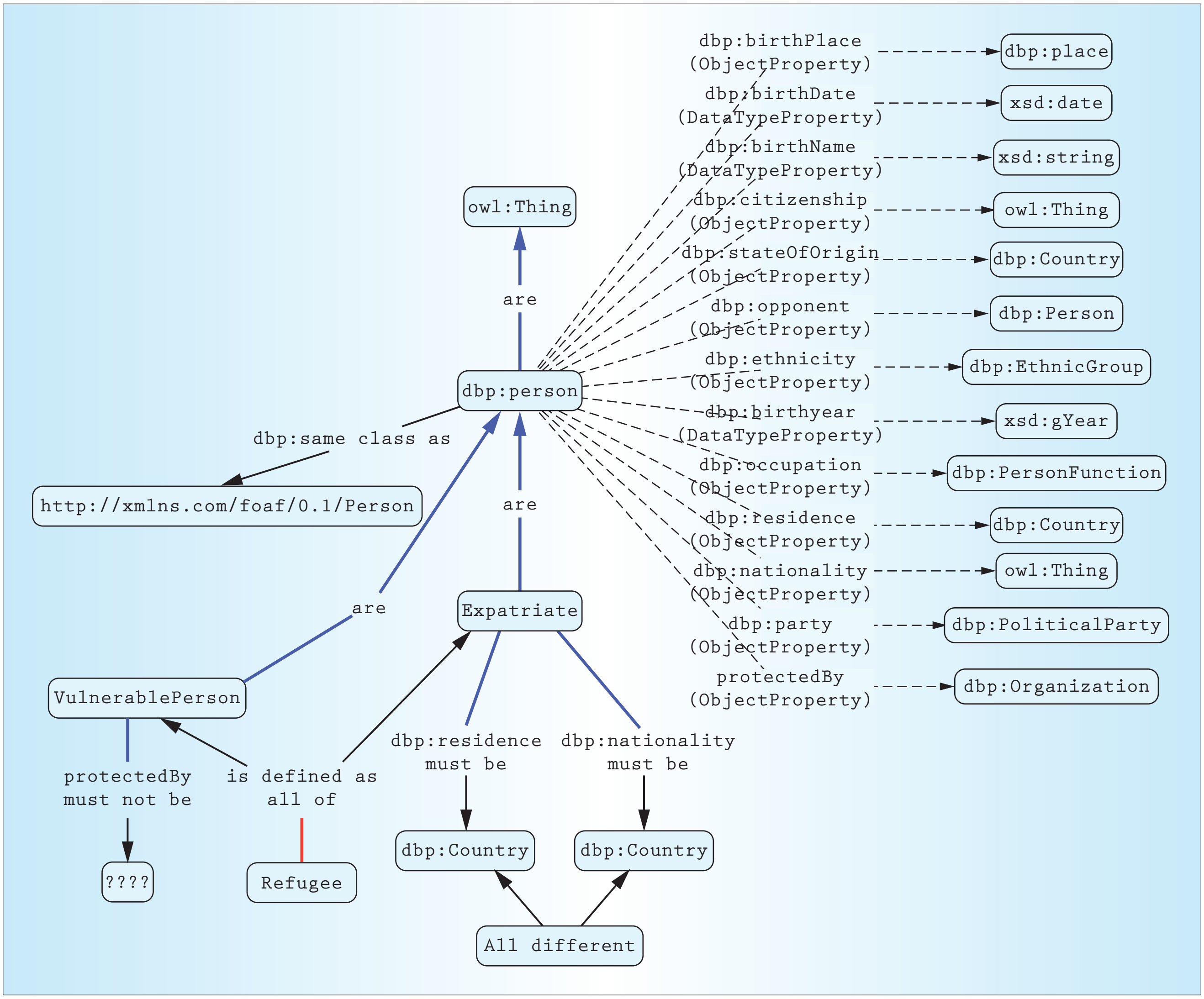

Figure 5. Developing new concepts by composing existing concept information. The concept of "Expatriate" is defined using a previously defined concept of Person and its supporting properties.

for the specific purpose of explaining the Bihari situation, we can convert our diagram shown in Figure 4 to the ontology shown in Figure 6. This diagram converts the intuitive description-logic concept-map presentation of the refugee situation into a logical form that can be reasoned over by other applications. In Figure 6, the square-corner rectangular nodes are OWL individuals, the black rounded-corner nodes boxes are OWL concepts, and the elements shown in green are ones that have not yet been converted into OWL. The ability to retain information that is not in logical form is an important benefit of the approach. This information serves the purpose of comments for the ontology. These comments may say something about the provenance of the concept (such as the concept `per:Refugee` – is defined by → "UN High Commissioner for Refugees"), or areas where the ontology is incomplete and requires further development (such as the concept "current actual identity").

Opportunistic Computation

We have described what might be regarded as a continuum. Knowledge expressed via natural language relies on syntax and the conventions of natural language communication. A concept map can make tacit information explicit in the form of propositions. The propositionally coherent concept map goes one step closer to the forms of logic. Each and every node-link-node triple can be read as a proposition. Going a step further, description-logic concept maps express meanings using the structures of description logic so that machines can draw all legal or necessary conclusions from content, and yet the concept maps are also meaningful to humans.

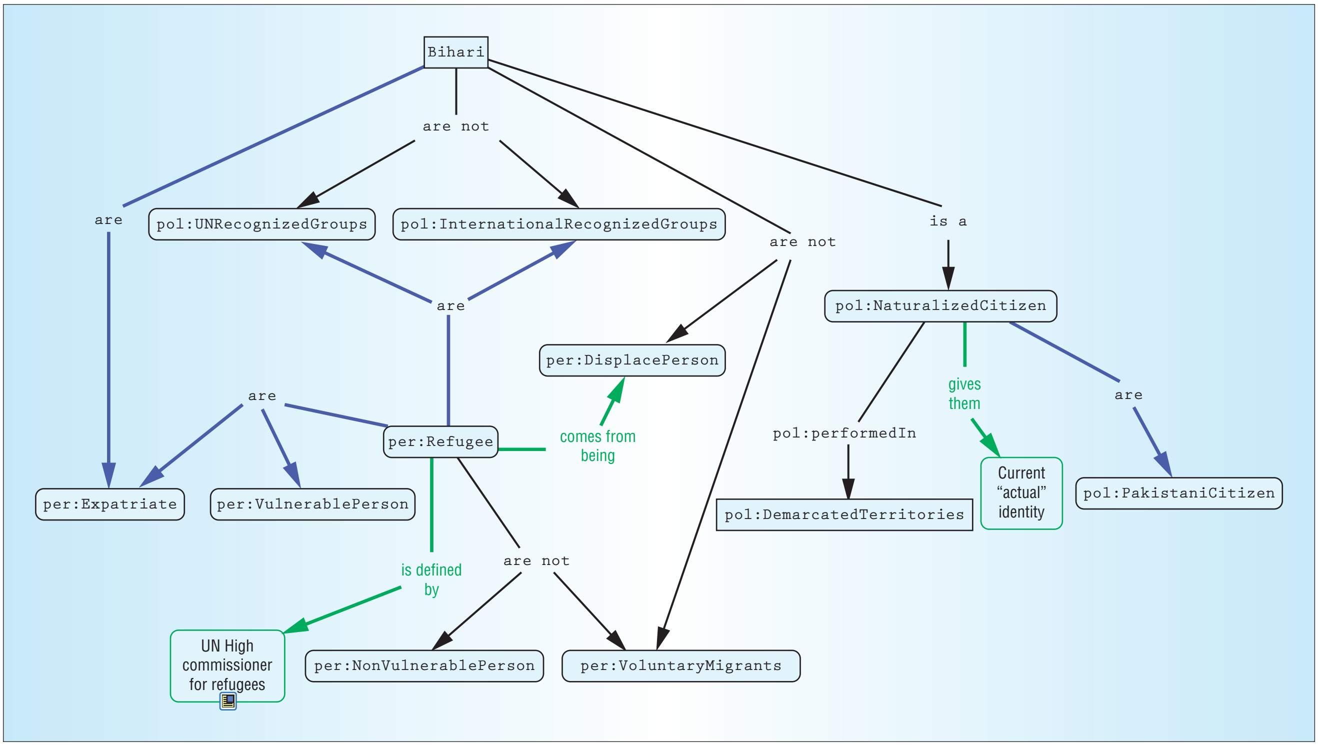

Figure 6. The results of "ontologizing" the description-logic concept map in Figure 1. The concept map includes both machine-readable ontology language for machine inference and descriptive connections that assist human readability.

In COE-OWL diagrams, the machine reads the parts of the diagram that it can, and performs allowable logical operations. These might include the identification of contradictions or the drawing of inferences. The machine simply ignores the parts of the diagram it cannot read, saying in effect, "OK, I can't do anything with that." Conversely, humans read those parts of the diagram that make sense, whereas they can ignore the parts that make little or no sense (depending on their knowledge of the formalisms and conventions), knowing that the two forms—the natural and the logical—are linked in a way that is itself meaningful and functional. This is what we mean by "opportunistic computation."

The human who is doing the work might fall anywhere on the learning curve with regard to their ability to understand the formalisms. The diagrams might help them advance in their fluency, while at the same time be sufficiently interpretable to support their own process of sensemaking. COE allows non-expert OWL users to interact with formal ontologies and begin to understand what the ontology represents and how it may be used.

The process for doing this is very similar to the process of developing concept maps in general. First, the user adds a number of concepts that are key for the domain under consideration. Then these concepts are organized into class-subclass hierarchies. The class definitions are then expanded to include restrictions on property values, and unions and intersections with other classes. As part of this, the user may need to define new properties and add individual instances of classes.

Throughout this process, users can add additional concepts and relations and use them to note areas into which the ontology might eventually be expanded, examples of how to use the ontology, and explanations of the design choices made in concept definition. To the human reading the diagram, these differences may be minor. COE can examine the map and extract all legal OWL definitions and use these definitions to populate lists associated with concepts (such as their super- and subclasses and their defined instances) as well as to help link to concepts in other ontologies.

When the ontology is ready to be shared, the concept map is exported into OWL and published. Comments in the OWL definitions store the graphical layout of all nodes and the text of nodes and relations that are not legal OWL constructs. This preserves the author-added information during translation to OWL and subsequent use. When read back into COE, the concept map will look exactly like it did before the export process. Supporting round-trip editing in this way allows the non-OWL information contained in the COE maps to maintain its value in communicating the explanations and intent of the ontology author.

Apart from the particular implementation that we have described here, a potential benefit to this general approach is that the representations would not only feed into computational activities but at the same time would allow users to understand, at least to some extent, what the software might be doing. A cardinal principle of human-centered computing is that machines must be understandable as well as usable and useful. This is typically expressed as the need for machines to be "transparent," but we believe that what is needed is for machine operations to be *apparent*, especially to individuals who are not software engineers. Thus, the approach we outline here would support three activities: (1) the human's sensemaking of the domain under study, (2) the machine's analysis of the domain, and (3) the human's sensemaking of what the machine is doing.

References

1. G. Klein et al., "Macrocognition," *IEEE Intelligent Systems*, vol. 18, no. 3, 2003, pp. 81–85.
2. G. Klein, B. Moon, and R.R. Hoffman, "Making Sense of Sensemaking 2: A Macrocognitive Model, *IEEE Intelligent Systems*, vol. 21, no. 6, 2006, pp. 88–92.
3. G. Klein, B. Moon, and R.R. Hoffman, "Making Sense of Sensemaking 1: A Macrocognitive Model, *IEEE Intelligent Systems*, vol. 21, no. 4, 2006, pp. 22–26.
4. K.E. Weick, *Sensemaking In Organizations*, Sage Publications, 1995.
5. P.M. Sanderson, A.M. McNeese, and B.S. Zaff, "Handling Complex Real-World Data with Two Cognitive Engineering Tools: COGENT and MacSHARPA," *Behavior Research Methods, Instruments and Computers*, vol. 26, no. 2, 1994, pp. 117–124.
6. J.C. Nesbit and O.O. Adesope, "Learning with Concept and Knowledge Maps: A Meta Analysis," *Review of Educational Research*, vol. 76, no. 3, 2006, pp. 413–448.
7. M.R. Khondoker and P. Mueller, "Comparing Ontology Development Tools Based on an Online Survey," *Proc. World Congress on Eng.*, Int'l Assoc. Engineers, 2010, pp. 188–192.
8. A. Zouaq and R. Nkambou, "Building Domain Ontologies from Text for Educational Purposes," *IEEE Trans. Learning Technologies*, vol. 1, no. 1, 2008, pp. 49–62.
9. S. Hauser, M. Nuckles, and A. Renkl, "Supporting Concept Mapping for Learning from Texts," *Proc. 7th Int'l Conf. Learning Sciences* (ICLS 06),International Society for the Learning Sciences, 2006, pp. 243–249.
10. I.M. Kinchin and L.B. Catob, "Using Concept Mapping Principles in PowerPoint," *European J. Dental Education*, vol. 11, no. 4, 2007, pp. 194–199.
11. C.D. Simone, R.F. Schmid, and L.A. McEwan, "Supporting the Learning Process with Collaborative Concept Mapping Using Computer-Based Tools and Processes," *Educational Research and Evaluation,* vol. 7, nos. 2–3, 2001, pp. 263–283.
12. C. Barroso and R. Crespillo, "Concept Maps: Tools for Understanding Complex Problems," *Proc. 3rd Int'l Conf. Concept Mapping* (CMC 08), Tallinn University, 2008, pp. 160–164.
13. T. Keller and S.-O. Tergan, "Concept Maps Used as a 'Knowledge and Information Awareness' Tool for Supporting Collaborative Problem Solving in Distributed Groups," *Proc. 2nd Int'l Conf. Concept Mapping* (CMC 06), University of Costa Rica, 2006, pp. 125–132.
14. I.S.M. Echeverría, S.A. García, and F.G. García, "Constructing Knowledge Models: Cooperative Autonomous Learning Using Concept Maps and V Diagrams," *Proc. 3rd Int'l Conf. Concept Mapping* (CMC 08), Tallinn University, 2008, pp. 113–116.
15. A. Cañas, *A Summary of Literature Pertaining to the Use of Concept Mapping Techniques and Technologies for Education and Performance Support,* tech. report, Institute for Human and Machine Cognition, 2003.
16. D.T. Moore, *Sensemaking: A Structure for an Intelligence Revolution*, Nat'l Defense Intelligence College Press, 2011.
17. D.T. Moore and R.R. Hoffman, "Sensemaking: A Transformative Paradigm," *Am. Intelligence J.*, vol. 29, no. 1, 2011, pp. 26–36.
18. R.J. Heuer and R.H. Pherson, *Structured Analytic Techniques*, CQ Press, 2011.
19. K.F. Farzana, "The Neglected Stateless Bihari Community in Bangladesh: Victims of Political and Diplomatic Onslaught," *J. Humanities & Social Sciences*, vol. 2, no. 1, 2008, pp. 1–19.
20. A. Newell, "The Knowledge Level," *AI Magazine*, vol. 2, no. 2, 1981, pp. 1–33.
21. K.M. Ford et al., "Knowledge Acquisition as a Constructive Modeling Activity," *Knowledge Acquisition as Modeling*, K.M. Ford and J.M. Bradshaw, eds., John Wiley, 1993, pp. 9–32.
22. B.R. Gaines, M.L.G. Shaw, and J.B. Woodward, "Modeling as a Framework for Knowledge Acquisition Methodologies and Tools," *Knowledge Acquisition as Modeling*, K.M. Ford and J.M. Bradshaw, eds., John Wiley, 1993, pp. 155–168.
23. N.E. Johnson, "Mediating Representations in Knowledge Elicitation," *Knowledge Elicitation: Principles, Techniques, and Applications*, D. Diaper, ed., Springer, 1989, pp. 177–194.
24. D. Nardi and R.J. Brachman, "An Introduction to Description Logics," *The Description Logic Handbook*, F. Baader et al., eds., Cambridge Univ. Press 2003, pp. 1–40.
25. T. Eskridge et al., "Formalizing the Informal: A Confluence of Concept Mapping and the Semantic Web," *Proc. 2nd Int'l Conf. Concept Mapping* (CMC 06), University of Costa Rica, 2006, pp. 247–254.
26. P. Hayes et al., "Collaborative Knowledge Capture in Ontologies," *Proc. 3rd Int'l Conf. Knowledge Capture*, ACM Press, 2005, pp. 99–106.
27. P. Hayes, *COE Manual*; www.ihmc.us/groups/coe/wiki/welcome/attachments/d2a1b/COEmanual06.pdf.