Representation as a Fluent: An AI Challenge for the Next Half Century *

Alan Bundy Fiona McNeill

School of Informatics, University of Edinburgh,
{A.Bundy, F.J.McNeill}@ed.ac.uk

Abstract

We argue that artificial intelligence systems must be able to manipulate their own internal representations automatically in order to deal with an infinitely complex and ever changing world, and to scale up to rich and complex applications. Such manipulation must go beyond the ability to change beliefs and learn new concepts in terms of old; it must be able to change the underlying syntax and semantics of the ontology. Initial progress is being made and is now urgent, due to the demands created by autonomous multi-agent systems. Understanding and automating this ability must be a major focus of artificial intelligence for the next 50 years.

1 Introduction

In the 50 years since the 1956 Dartmouth AI conference, we have become increasingly aware of the key role of representation in all areas of artificial intelligence. In the original proposal for the conference[1], for instance, John McCarthy says:

> *"The emphasis here is on clarifying the environmental model, and representing it as a mathematical structure."*

In his seminal Machine Intelligence 3 paper [Amarel, 1968], Saul Amarel demonstrated the sensitivity of search space size to problem representation by exhibiting an increasingly efficient series of representations for the Missionaries and Cannibals problem. Since then, huge amounts of AI energy has been expended on hand-crafting representations, or *ontologies* [Uschold & Gruninger, 1996], to hit a sweet spot combining adequate expressivity with inferential efficiency.

Despite these prodigious efforts, a few minutes studying any particular representation rapidly reveals deficiencies in expressivity or efficiency or both. The inevitability of expressivity deficiencies have long been recognised. For instance, the *qualification problem* refers to the practical impossibility of specifying all the preconditions of an action required to guarantee successful application. The *ramification problem* refers to the practical impossibility of specifying all the effects of an action.

There have been valiant, but ultimately misguided, attempts to provide a general-purpose, common-sense knowledge-base, for instance, the CYC Project[2] or SUMO[3] (Suggested Upper Merged Ontology). But even Cycorp has recognised the need to customise its general purpose knowledge base to each application. And the developers of SUMO recognise that their ultimate goal may be unattainable.

*The research reported in this paper was supported by EPSRC grant GR/S01771, and EPSRC studentship and the EU Open Knowledge project. We are grateful to Chris Walton for feedback on an earlier draft.

[1] `http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html`

[2] `http://www.cyc.com`

[3] `http://ontology.teknowledge.com/`

> *"The question, then, is how do we handle cases like these, given that our goal is to construct a single, consistent, and comprehensive ontology. It will be unfortunate if we cannot reach this goal, but perhaps it is unattainable." [Niles & Pease, 2001][p6]*

We will argue that attempting to build a general-purpose representation is chasing rainbows. The world is infinitely complex, so there is no end to the qualifications, ramifications and richness of detail that one *could* incorporate, and that you *might need* to incorporate for a particular application. This is not just a question of adding some additional facts or rules; it may also be necessary to modify the underlying representational language: its syntax, its representational power or even its semantics or logic. For a narrow application, it is often sufficient to hand-craft a representation that hits the desired sweet spot. But this will not be sufficient for the deeper and wider ranging applications that are the ultimate goal of artificial intelligence, e.g., autonomous agents able to solve multiple and evolving goals in a complex and messy environment in collaboration with some other agents and in conflict with others. For these, more ambitious, applications, the representation must be a *fluent*, i.e., it must evolve under machine control. This proposal goes beyond conventional machine learning or belief revision, because these both deal with content changes within a fixed representation. The representation itself needs to be manipulated automatically.

We believe that automatic representation development, evolution and repair must be a major goal of artificial intelligence research over the next 50 years.

2 An Example: Motherhood

To illustrate our argument, consider the standard family tree ontology used to illustrate many logic-based formalisms. Let $Mother(x)$ represent the mother of x, where $Mother$ is a function from children to their mothers. We can then define $Maternal\_Grandmother(x) ::= Mother(Mother(x))$.

However, motherhood has become much more complicated than this in our modern world. We have long had stepmothers and adopting mothers, but medical fertilization techniques have recently provided: biological mothers, who provide the eggs, but don't carry their baby to term; surrogate mothers who host the baby in their womb, but who do not provide the egg; even mothers who provide one part of the egg and other mothers who provide the other half. Who knows what tomorrow will bring in this rapidly changing field.

To deal with these constant changes in the world, we are constantly having to update our representations of that world. Firstly, $Mother$ can no longer be a function, since $Mother(x)$ is no longer guaranteed to return a unique result[4]. We must replace the function with a relation $Mother(x, y)$ where y is the mother of x. Now we could replace this single $Mother$ predicate with several new predicates: $Natural\_Mother$, $Step\_Mother$, $Adopting\_Mother$, $Biological\_Mother$, $Surrogate\_Mother$, etc. Alternatively, we could recognise the similarity between these relationships by replacing the binary $Mother$ predicate with a ternary one, in which the third argument specifies the type of motherhood involved, e.g., $Mother(x, y, Natural)$, $Mother(x, y, Step)$, $Mother(x, y, Adopting)$, etc. This third argument would have to picked up and manipulated by any predicate defined in terms of $Mother$, for instance:

$$Maternal\_Grandmother(x, z, Type(\tau_1, \tau_2)) ::= \exists y.\ Mother(x, y, \tau_1) \wedge Mother(y, z, \tau_2)$$

The term $Type(\tau_1, \tau_2)$ tells a complex story about the real family relationship. Modern families are like this: "she's the step-mother of his biological mother".

If a reasoning system is restricted to a fixed ontology, this will rapidly get out of date. Reasoning systems need to be able to develop, evolve and repair their underlying representations as well as reason with them. The world changes too fast and too radically to rely on humans to patch the representations. Some changes happen on a daily or even shorter timescale.

[4] Perhaps, soon, the very existence of a result will be no longer be guaranteed either.

3 Representational Repair in Multi-Agent Planning

As an illustration of the kind of research programme we have in mind, We will describe the recent PhD project of the second author [McNeill, 2005, Bundy *et al*, 2006]. This addresses the problem of ontology alignment in a multi-agent planning environment: a problem which it is essential to solve in order to realise the vision of the Semantic Web [Berners-Lee *et al*, 2001].

The environment of McNeill's project is that some agents offer services and others require these services. Each agent represents these services with STRIPS-like planning action rules with preconditions and effects written in a restricted version of KIF, a first-order logical language. Planning systems of this kind form the basis for many reasoning systems for multi-agent systems, for instance, in BDI-based systems, in which the intentions and desires provide the goals for a planning system and the beliefs provide the ontology over which planning is performed. We assume that there has been an attempt at standardisation of ontologies, e.g., with all the agents' authors downloading a common ontology from a central server. However, it is inevitable with any sufficiently large agent community that there will be small differences between the ontologies, for instance, caused by downloading different versions of the ontology or by local customisation to meet specific user requirements.

The purpose of McNeill's Ontology Refinement System (ORS) is to identify and repair these ontological mismatches at run time. Moreover, this must be done without full access to the other agents' ontologies. We cannot assume such full access because: (a) agents are unlikely to have been built with the functionality to provide access to their underlying ontology; (b) in any case, some details of an agent's ontology are likely to be confidential, e.g., a commercial secret; and (c) some aspects of the ontology may be generated dynamically in response to requests, e.g., RSS feeds.

Note that this project differs from previous approaches to ontology mapping, merging or aligning [Kalfoglou & Schorlemmer, 2003] in several ways.

- The ontologies are used for planning, so have to be more expressive than the concept ontologies (isa hierarchies) that are usually mapped.
- We do not assume complete access to the mismatched ontologies.
- Ontology repair is conducted entirely at run-time.
- We assume that the mismatches between the ontologies are relatively minor in extent.

The ORS ontology repair operations consist mostly of syntactic manipulations of the underlying KIF representation. For instance, (a) the number or order of the arguments of a predicate may be changed; (b) a predicate or constant may be divided into two or more, or two or more may be merged into one. The only belief revision operations available to ORS are to add or remove a precondition of an action rule.

The diagnosis and repair of a faulty ontology is guided by a decision tree. The nodes of this tree have questions such as "did the other agent ask a question we were not expecting to be asked?". Depending on the answer, the diagnosis process can ask a question further down the decision tree, enter the *Shapiro Algorithm* or suggest a repair. The Shapiro Algorithm tries to determine the truth or falsity of the agent's beliefs by a restricted dialogue with the other agent [Shapiro, 1983].

ORS has been evaluated on successive versions of third party ontologies from the KIF and planning communities (including SUMO). It was able to deal successfully with just over a third of the individual mismatches between these ontologies. Many of the mismatches were out of scope of the project, for instance: arbitrary predicate name changes, comment or formatting changes, changes that could not be represented in our restricted version of KIF. Given the novelty of our approach, we regard these results as very encouraging.

Our current research is directed at removing the many simplifying assumptions of the initial project and applying it to a practical architecture for open, automated communication between multiple agents. Part of this work is to combine our approach with conventional ontology mapping abilities, e.g., to address name differences between predicates and constants with the aid of WordNet[5], which identifies synonyms, hyponyms and hypernyms.

4 Conclusion

We have argued that artificial intelligence systems must be able to manipulate their own internal representations automatically in order to deal with an infinitely complex and ever changing world, and to scale up to rich and complex applications. Such manipulation must go beyond the ability to change beliefs and learn new concepts in terms of old; it must be able to change the underlying syntax and semantics of the ontology. Initial progress is being made and is now urgent, due to the demands created by autonomous multi-agent systems [Hendler, 1999]. Understanding and automating this ability must be a major focus of artificial intelligence for the next 50 years.

References

[Amarel, 1968] Amarel, S. (1968). On representations of problems of reasoning about actions. In Michie, D., (ed.), *Machine Intelligence 3*, pages 131–171. Edinburgh University Press.

[Berners-Lee *et al*, 2001] Berners-Lee, Tim, Hendler, James and Lassila, Ora. (May 2001). The Semantic Web. *Scientific American*, 284(5):34–43.

[Bundy *et al*, 2006] Bundy, A., McNeill, F. and Walton, C. (2006). On repairing reasoning reversals via representational refinements. In *Proceedings of the 19th International FLAIRS Conference.*

[Hendler, 1999] Hendler, J. (March 1999). Is There an Intelligent Agent in Your Future? *Nature - Web Matters*, (3).

[Kalfoglou & Schorlemmer, 2003] Kalfoglou, Y. and Schorlemmer, M. (January 2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31.

[McNeill, 2005] McNeill, Fiona. (2005). *Dynamic Ontology Refinement.* Unpublished Ph.D. thesis, School of Informatics, University of Edinburgh.

[Niles & Pease, 2001] Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9. ACM Press, New York, NY, USA.

[Shapiro, 1983] Shapiro, E.Y. (1983). *Algorithmic Program Debugging.* MIT Press.

[Uschold & Gruninger, 1996] Uschold, M. and Gruninger, M. (June 1996). Ontologies: Principles, Methods, and Applications. *Knowledge Engineering Review*, 11(2):93–155.

[5] `http://wordnet.princeton.edu/`