The Open World: Challenges in Knowledge Representation on the Web

Harry Halpin
Institute for Communicating and Collaborative Systems
University of Edinburgh
2 Buccleuch Place
Edinburgh, United Kingdom
H.Halpin@ed.ac.uk

1. INTRODUCTION: NO FREE LUNCH

When paradigms or subject matters change, what were once great and insurmountable problems recede into the distance, easily solved or safely ignored. The reverse can happen, as shedding new light into some dark unexplored territory may cause a molehill to reveal itself to be mountain. The impact of the World Wide Web upon knowledge representation is no exception: Has it solved any of the problems of knowledge representation or has it made them more insoluble? While the Web is a remarkable feat of engineering - a brilliant hack - that has irrevocably changed our society by its sheer rate of adoption, the sudden resurrection of artificial intelligence it has caused, from Google to the Semantic Web, should give one pause. At first one is tempted to concur that there must be something special about the Web that has caused its remarkable growth, so that it has escaped or simply not encountered the long-standing problems about knowledge representation that have long-plagued artificial intelligence and the engineering extension to philosophy that calls itself artificial intelligence.” However, the Web is not magical, and an analysis of its architecture and that architecture’s impact upon knowledge representation is in order. So while the success of the Web has given an uplifting feeling to many researchers in knowledge representation, there have also been nagging doubts hiding in the back of our minds, for the shadows of previous problems re-emerge like unbidden shadows to haunt our endeavors in this new open world.

Although it may come as some surprise, we claim that insofar as the Web has remained primarily a system of distributed hypertext, it can avoid the hard problems that have plagued knowledge representation. However, these hard problems can not be avoided forever, for as vast amounts of Web-based information grows seemingly exponentially, evermore energies are needed to categorize and share this information. There are two methods, often seen as opposing, that broadly characterize the classification of information on the Web. The first movement would have applications scrape the information in implicit in data. The second has the authors make the information in their data explicit.

Discovering implicit information on the Web signals a return to empiricism, as only through gathering large amounts of real-world data can programs discover implicit information via statistics, as exemplified by search engines such as Google. To some extent this technique makes a vast amount of sense on the hypertext Web, where the primary medium of communication is human language. The Web itself can be conceived as a giant corpora of human language frequencies [Keller and Lapata, 2003]. Even with statistical natural language processing there are three large problems looming in the distance. The first is the immense amount of data that must be centralized in an index in order for machine-learning to succeed may eventually be dwarfed by the open-ended and ever-increasing nature of information on the Web. This could be dealt with by the use of decentralized indexing systems. However, another problem is the blind amount of trust users put in statistical systems, for any statistical system is dependent critically on the assumptions about distributions and parameters it makes, assumptions that are correct insofar as they effectively model the information itself. While these assumptions are currently hidden, as these systems proliferate, they will need some way of both exposing their assumptions and proving their trustworthiness. Lastly, while statistics can be wonderful at searching for topical web-pages using keywords, there are certain types of data that are not amendable to statistics, and no-one wants statistics telling them approximately how much money they have in the bank. The vast amount of structured knowledge on the Web automatically generated from large databases and then put onto the Web in human-readable web-pages is testament to this, yet trapping the data inside web-pages makes it all the more difficult for other programs to process.

The second movement for information management on the Web is to explicitly represent one’s knowledge using some sort of machine-usable format. This movement has led to a revival of the field of knowledge representation, historically falling under the domain of artificial intelligence. Even statistical search engines cannot escape knowledge representation, for in a broad sense all the information on the Web can be considered to be some form of external knowledge representation in order for it to be on the Web at all, and this ranges from the text and pictures in people’s web-pages to the information in databases accessible only over highly-secured Web services. While the implicit information in the hypertext Web has skirted around questions of knowledge representation so far, the much-heralded next generation of the Web, the Semantic Web attempts to increase the scope of the Web by putting explicit information on the Web, and so collides with issues of knowledge representation. As de-
fined by Berners-Lee “The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation” [Tim Berners-Lee and Lassila, 2001]. To anyone versed in the history of artificial intelligence, this ambitious sentence opens a veritable mindfield of problems. So, before we go farther will first grasp the history of knowledge representation so we may glimpse its future.

2. A BRIEF HISTORY OF CLASSICAL ARTIFICIAL INTELLIGENCE

Artificial Intelligence as a project was originally a strange stepchild of cybernetics, the study of systems between humans and machines inaugurated by Norbert Wiener [Wiener, 1948]. Cybernetics as a unifying framework began to fragment, and inspired by Turing yet another unifying and competing framework, artificial intelligence, whose followers began propagandizing for the removal the error-prone human from the picture altogether, to be replaced with digital computers of human-level intelligence [Turing, 1950]. Artificial intelligence occupies the curious position of being both an engineering project and a philosophical project, as should be apparent from the statement of John McCarthy at the 1956 Dartmouth Conference Proposal: “The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it” [McCarthy et al., 1955]. The crucial component “intelligence” itself is still - decades later - not defined in any commonly agreed upon manner. The proposal put forward by McCarthy gave a central role to a Kantian “common-sense” that unified the faculties of intelligence. McCarthy defined “common-sense” operationally as “a program has common sense if it automatically deduces for itself a sufficient wide class of immediate consequences of anything it is told and what it already knows” [McCarthy et al., 1955]. The artificial intelligence project was the summit of analytic philosophy. Every aspect of intelligence could be formalized symbolically independent of the physical substrate of the human body, and then when these formalisms were injected by programmers into a computer, that computer would be autonomously intelligent. This did not sound as far-fetched at the time. Chomsky had just recently dealt a death-blow to behaviorism by formalizing the syntax of the universal grammar, so it logically followed that a formalization of the “knowledge level” above the sheery syntactic level would follow [Newell and Simon, 1976, Chomsky, 1957]. The question was one of knowledge representation: What explicit and machinic language could be created to formalize human knowledge?

Given its heritage in analytic philosophy and mathematics, it should come as no surprise that the first knowledge representation language put forth was first-order predicate calculus, and the encoding of “common-sense” into logical propositions was put forward as the unifying project of AI. A number of extensions were proposed to first-order logic in order to make it better resemble human intelligence, such as “default logic.” Default logic is supposed to mimic how humans make assumptions in the face of limited knowledge: “If the bear is white, the bear is a polar bear,” an improper assumption if the bear is actually an albino. A number of researchers thought that a declarative formalism well-suited for mathematics was not well-suited for human intelligence, and they began exploring other options they termed more “procedural” [Winograd, 1972]. The main competitor to first-order logic, known as frames, considered knowledge to be composed of “frames” and “slots” that are filled in by various values [Minsky, 1975]. Another popular model is the semantic network, for one common behavior that humans have attempting to represent knowledge is to begin to start drawing circles and lines between them, as Figure 1 illustrates.

All of these knowledge representation languages were tied together by their use of hierarchy, for models at that time, as pioneered by Herbert Simon, laid the enabling factor of human complex behavior in society as hierarchy, as exemplified by large corporations like IBM [Simon, 1965]. This led to a nearly pathological obsession with taxonomy and other forms of “arborescent” thought by artificial intelligence. Early accounts of the mind like Minsky’s Society of the Mind used hierarchy and specialized components called “daemons” to explain the workings of the mind, and while the theory is rarely used, “daemons” survive today in the workings of the UNIX operating system [Minsky, 1985]. A plethora of representation schemes all flourished, attacking all sorts of problems, to such an extent that Herbert Simon wrote that “machines will be capable, within twenty years, of doing any work that a man can do” [Simon, 1965].

Much of the work in the classical era of artificial intelligence, from Logic Theorist [Newell and Simon, 1956] to SHRDLU [Winograd, 1972], simulated intelligence in a well-specified domain such as proving logical theorems or moving blocks. Although AI had done well in “tightly-constrained domains,” extending this ability had “not proved straightforward” [Winston, 1976]. The crux of the problem was that knowledge-representation schemes were incredibly ad-hoc and task-dependent, so that they could not be used outside the limited domain in which they were created. When researchers attempted to communicate about their knowledge representation scheme, since these schemes had no basis in any logical formalism, no-one really knew what the symbols in the language “meant” except the author. Even within a widespread knowledge representation form such as semantic networks, it was shown that a principal element such as a “link” was interpreted in at least three different
Knowledge representations were not obviously denoting the knowledge they supposedly represented. One answer was to return to the use of formal logic to give a well-defined semantics using a language such as first-order predicate logic, and it was shown many of the ad-hoc “procedural” knowledge systems could be re-expressed in first-order-logic [Hayes, 1977]. The clear next step was the formalization of as much common-sense knowledge as possible using rigorous standards of logic, in order to overcome small, domain-specific strategies [Hayes, 1986]. A large technical problem arose though, that of undecidability, the logical variant of the Turing-completeness from computational theory. A system is undecidable if, for any given inquiry, one cannot predict whether or not the system will produce the correct result, and first-order predicate calculus is decidable by the notion of taxonomic classification, informal semantic networks were transformed into formally defined description logics. This logical “family” branched into a combinatoric number of languages with different expressive powers, and the real quest at hand began to keep expressive languages that maintained decidability.

Other researchers attempted to break even further with logic by specifying knowledge representations based on numeric vectors of similarity¹ and almost any other conceivable extension, like the immensely flexible KRL (“Knowledge Representation Language”) (Bobrow and Winograd, 1977). KRL too failed due to undecidability, as even when faced with a simple cryptographic puzzle it would fail to produce an answer. The more expressive a knowledge representation language becomes, the less tractable (and so more likely to be undecidable) the language becomes, so for expressive languages to do anything with the language with any certainty becomes impossible. The death-knell of classical artificial intelligence was given by the Brachman-Smith survey [Brachman and Smith, 1991], a testament that the immense range and diversity of AI systems was a virtual Tower of Babel. While some remaining researchers maintained that all of necessary common-sense knowledge could be encoded shortly [Lenat and Feigenbaum, 1987], many other researchers left the field and the nascent “expert system" industry collapsed. To this day, Lenat is still encoding “common-sense" into Cyc, the largest remaining attempt to encode common-sense knowledge in some sort of knowledge representation [Lenat, 1990].

Given the failure of classical artificial intelligence, Brian Smith in his critique of Lenat’s Cyc program noted that the number of “common-sense" facts continually grew larger and so the entire project receded perpetually into the future, perhaps ever incomplete. An alternate paradigm was to realize useful knowledge was situated in a particular task at hand and embodied in the very body of the agent [Dreyfus, 1979]. On this basis, it was unlikely at best and philosophical chicanery at worse that any traditional knowledge representation language could capture all aspects of knowledge necessary for human-level intelligence [Smith, 1991]. This was echoed by Winograd, who believing that analytic philosophy was fundamentally flawed, believing instead that artificial intelligence had to re-founded on hermeneutic philosophy, although no-one know how one could do hermeneutic on a digital computer. Regardless, this was an intellectual move that has at this junction led artificial intelligence to be dominated by the notion of embodiment [Winograd and Flores, 1986]. Instead of formalizing common-sense, Smith instead asked what practical purposes this technology could be put to: “Forget intelligence completely, in other words; take the project as one of constructing the world’s largest hypertext system, with Cyc functioning as a radically improved (and active) counterpart for the Dewey decimal system. Such a system might facilitate what numerous projects are struggling to implement: reliable, content-based searching and indexing schemes for massive textual databases", a statement that prefigures the development of the Web [Smith, 1991].

3. THE WEB AS A UNIVERSAL INFORMATION SPACE

The World Wide Web itself is defined by Berners-Lee as a universal information space, a totaling system for all digital resources [Berners-Lee, 1996]. As when one considers Hegel, it is not the thinker but the system of thought that is totalizing. Berners-Lee is far from a totalizing thinker - for if anything he is an autodidact - yet the technological system Berners-Lee has produced is totalizing, encompassing evermore information.

The Web as a totaling system is not accidental. First, it is part of the genetic heritage of the Web: the World Wide Web is built on the Internet, yet another descendant of cybernetics. Unlike artificial intelligence, which hoped to replicate and so replace human intelligence, the Internet was originally began as a way to increase “man-computer partnership" by J.C.R. Licklider, acolyte of Wiener and primary advocate of the notion of interactive computing, where a computer should be response with a ever decreasing feedback cycle. Unlike artificial intelligence, this vision did not remove the human from the picture, instead searching on how to better use digital computers as communication devices: “Creative, interactive communication requires a plastic or moldable medium that can be modeled, a dynamic medium in which premises will flow into consequences, and above all a common medium that can be contributed to and experimented with by all. Such a medium is at hand: the programmed digital computer" [Licklider and Taylor, 1968]. As head of the Information Processing Technologies Office at ARPA Licklider and his successor Taylor were to implement a computer network for researchers to interactively share their resources across the nation, and after decades of development this became the Internet, the network of networks. While the various protocols allow some level of universality, as any network with a proper gateway and IP address can connect to the Internet and so share its resources, the connectivity was still application-dependent. The Internet’s protocol TCP/IP successfully solved the problem of interacting with other computers on a low-level, but as far as the actual software was concerned chaos reigned. Every application such as FTP (“File Transfer Protocol") or Telnet had its own protocol that was implemented above TCP/IP, as well as its own user-interface.

¹A sort of system that has been further revived by information retrieval.
The Web puts all resources on the Internet in the same universal space by using a scheme called URIs, or “Uniform Resource Identifiers.” One can easily recognize a URI by its now familiar syntax, as defined most recently in RFC 3986, such as http://www.example.org [Berners-Lee et al., 2005]. URIs are not limited to only hypertext documents delivered by the HyperText Transfer Protocol, but URIs also subsume other Internet schemes like File Transfer Protocol (ftp://myhost.org:6665), and further include mundane things such as e-mail (mailto:John.Doe@example.com) and phone numbers (tel:+1-816-555-1212). The general structure of a URI allows one person with a single domain to define virtually limitless number of URIs, since one can embed the hierarchical parts of the URI recursively.

The URI was originally called the “Universal Resource Identifier” in order to stress its universality. However, the rest of the IETF (The Internet Engineering Task Force, one of the governing bodies of the Internet) was lost in philosophical bickering over the possibility (and perceived arrogance) of using “universality” to describe the Web, and the entire working group made to standardize URIs was nearly shut down due to lack of progress. Finally, a compromise was struck: Berners-Lee and other settled on URIs, or “Uniform Resource Locators” [Berners-Lee et al., 1994], later generalized to URIs or “Uniform Resource Identifiers.” Tim Berners-Lee has stated that URIs “identify one thing” [Berners-Lee, 2003]. This thing is a resource.

Tim Berners-Lee once stated that the great thing about resources is that he went on for years without having to define it. The most current IETF RFC for URIs states that it does not “limit the scope of what might be a resource” but that a resource “is used in a general sense for whatever might be identified by a URI” such as “human beings, corporations, and bound books in a library” and even “abstract concepts” [Berners-Lee et al., 2005]. So, in this twist of hand, by radically underspecifying what resources are, resources can be anything at all that could use an universal name, and URI schemes can specify any protocol whatsoever that might need something with a name. The formerly chaotically striated space of the Internet was given as uniform and striated syntax that, ironically enough, makes it a smooth space of identification.

There is no “Web” yet in just a set of identifiers. What one needs is another concept, the link. A link is merely a connection between two resources, and allowing all resources to share the same universal space along with unconstrained links creates a web, where any resource may connect to any other. What is meant by “connection” is underspecified, but it generally demarcates some sort of relationship between the two resources. Some languages, such as HTML, have a flat space of relationships, while others such as XLink, allow the relationships to have types [DeRose et al., 2001]. In an universal information space, links are of utmost importance, as URIs serve as a single global identification system, so that “any party can share information with any other party,” [Jacobs and Walsh, 2004] This leads to the principle of global identifiers: “Global naming leads to global network effects”[Jacobs and Walsh, 2004]. If people can mention and use your data by writing the URI by linking to it and sharing it, the value of the web itself increases as everyone has the possibility of also now connecting to the new identifier.

While a resource can be literally anything, representations exist are what a resource returns to an agent when the resource is accessed over the Web. By “accesses” we mean that when using the URI through some sort of URI scheme-aware agent, such as the use of HTTP’s GET command to retrieve a web-page using a web browser, one gets a stream of bits on the other end. A representation is defined as “data that encodes information about resource state” [Jacobs and Walsh, 2004]. One resource can have multiple representations, especially due to content negotiation. Content negotiation allows computer hosting the resource to deliver differing representations depending on the request of the agent. One agent may request a version of a resource in HTML using the French language while another may want the same representation of the resource as only a picture, or as a collection of logical statements. This allows resources to adapt to differing languages and browsing devices without breaking the universality of URIs. Without content negotiation, for every state change or accessing of a resource using a new protocol, a new URI would have to be minted, so that the French and English versions of a website would have different names. Content negotiation could be particularly useful for the always-frustrating mobile phone web surfing, since it would allow one URI to deliver both a mobile compatible and more full-featured version of the same page. This picture from the W3C’s Architecture of the Web sheds some light on the connection between how the story of URIs, representations, and resources works on the hypertext web [Jacobs and Walsh, 2004]. A user-agent requests a URI it has gotten from somewhere for the “Weather in Oaxaca” and proceeds to attempt to access that resource over the Web. A representation is delivered with an attached content type that allows the agent to decipher the results. Figure 2 from the W3C [Jacobs and Walsh, 2004] gives an overview of the example of “getting the weather in Oaxaca” on the Web.

![Figure 2: The relationship between identifier, resource, and representation](image)

Yet the following sentence seems to confuse the situation: “Representations do not necessarily describe the resource, or portray a likeness of the resource, or represent the resource in other senses of the word ‘represent’”[Jacobs and Walsh, 2004]. A representation on the Web is any collection of bits with an Internet Media Type (also occasionally called “content types” or “MIME (Multipurpose Internet Mail Extensions) types”) [Bray, 2002], and these go by names such as text/html, application/xhtml+xml, or image/jpeg. This gives the authors of representations a way of specifying what the
preferred method of processing of their types are, and an authoritative list is given by the Internet governance body IANA (Internet Assigned Numbers Authority). So, a resource can have multiple representations on the Internet.

These principles of Web architecture are abstract, and can in principle apply to any information-sharing system. The first incarnation of the Web was a simple multi-user hypertext system. Hypertext is defined by Ted Nelson as a sort of document that can “branch or perform on request,” to be displayed in a web browser [Nelson, 1970]. Berners-Lee and Connolly created a rather compact hypertext language called the “hypertext markup language” (HTML), where markup is defined as “syntactically delimited characters added to the data of a document to represent its structure” [Berners-Lee and Connolly, 1995]. While there were many a previous hypertext systems such as Hypercard, there was only easily readable by humans, would create machine-readable metadata to allow machines to access their information. The formal framework for this metadata, called the Resource Description Framework (RDF). RDF is a simple language for creating assertions about propositions [Hayes, 2004]. The basic concept of RDF is that of the triple: any statement can be composed into a subject, a predicate, and an object as illustrated by Figure 3. At this level, the Semantic Web is nothing more than an abstraction of the Web with links, where the resources and their directed links (predicates holding between subjects and objects) can be given types. For example, the statement “The creator of the web-page is Henry Thompson” can be phrased as a triple in the following manner: www.inf.ed.ac.uk/hdc:creator “Henry Thompson.”

The framework given by RDF is reinventing the wheel of early research into semantic networks in artificial intelligence. The first incarnation of RDF was approached much in the slapdash manner of HTML, with a simple syntax given to describe these graphs[Lassila and Swick, 1999]. This was due to a confusion between the syntax and the semantics of the Web. HTML is defined as a syntax with its rendering into an expression for presentation to a human agent left informally defined, so there was no clear semantics per se. This was no longer enough if one wished for there to be clear “machine-readable” semantics for the data that were independent of any particular syntax. While “machine-readable” sounds at first nebulous, it has a behavioral definition: A program has a formal semantics if its operation can be expressed in a formal logic independent of its syntax that allows two separately implemented programs to deduces the same class of consequences of statements it produces, or perform the same class of actions. Pat Hayes, the same one who noticed the lack of formal semantics in original work in semantic networks, got involved with RDF and crafted a set of formal semantics for RDF [Hayes, 2004]. RDF was extended to that of a full ontology language, called the Web Ontology Language (OWL) [Welty et al., 2004]. While RDF can just make statements and very limited deduction, OWL allows one to classify any statement in a taxonomy of classes. OWL consists of three languages, of which OWL Lite and OWL DL are described by the description logic SHOIN(D) and SHIF(D) respectively [Horrocks et al., 2003]. This Web Ontology Language (OWL) is thus more expressive than RDF [Welty et al., 2004], and its heritage comes more from

4The final clause in this sentence is somewhat a slight of hand and should be a conjunction rather than a disjunction. The former clause assumes a logical manner based on deduction and the latter clause on functions and procedures, but as can be shown through the Curry-Howard isomorphism [Howard, 1980] in some cases these can be shown to be the same set of computations

4At http://www.iana.org/assignments/media-types/

3As he said is his keynote address to the World Wide Web Conference in 1994
the taxonomic and classifying ambitions of classical artificial intelligence than a simple abstraction of the Web as embodied in RDF. In this manner, the Semantic Web has at this point evolved into the full-scale problem of knowledge representation and ontology development, albeit with goals and tools that have been considerably modified since their inception during the origins of artificial intelligence. The question of the hour is: Are these differences between classical artificial intelligence and the Web enough to let it overcome the fundamental problems of knowledge representation?

5. DIFFERENCE OF SCALE

For the first time in human history, mammoth amounts of human-readable digital information are easily accessible. While it is unclear exactly how much data is on the Web, there are increasing demands for more intelligent ways of navigating and organizing it. This contrasts with the origins of artificial intelligence, where the amount of information in digital form was tiny. The sheer quantity of human-made ontologies and metadata available over the web is at the time of this writing still small (currently 5 million sources of Semantic Web data, containing over 363 million statements), but larger than the projects of classical artificial intelligence including Cyc (which has released its taxonomy in OWL). So, the Semantic Web’s potential for economies of scale is larger than that of classical artificial intelligence.

Before any celebrations commence, let us remember the size of the Semantic Web is currently exponentially dwarfed by size of the hypertext Web. Most of the data on the Web in uninterpretable by digital computers, so a rapidly growing body of work attempts to address the automatic extraction of Semantic Web data from text in web-pages [Dill et al., 2003]. As language is inherently ambiguous, just as in artificial intelligence, the conversion of natural language to Semantic Web knowledge representations is always going to be error-prone, since even if parsing technologies become perfect, human language makes many assumptions about the knowledge of the reader that are not made explicit in human text, making translation of human text from a source like Wikipedia into Semantic Web knowledge almost impossible [Suh et al., 2006].

The other strategy for deploying the Semantic Web is to map data from structured relational databases into RDF, and a number of programs already exist to do [Bizer and Seaborne, 2004]. One should be suspicious of any plot that hopes that all databases will simply export their databases to RDF and have them merge using the “Great Schema in the Sky,” as even what appears to be semantically simple databases have very complex assumptions. More realistic is the hope that within domains where there is a need for information to be integrated, RDF can and will be used as a common exchange format to map to and from already existing schemas [Halpin and Thompson, 2006]. As for natural language, it is likely only text that is already marked-up with explicit semantics, such as already can be done using microformats[Suda, 2006], will be some level of reliability converted to data for the Semantic. Although scale is one difference between the Semantic Web and artificial intelligence, it is still a difference a degree than of kind.

6. DECENTRALIZATION

The failure of artificial intelligence was in part forced by having different research teams attempt to formalize ever larger and larger domains by each hand-coding the domains in differing knowledge representation languages. Knowledge representations by a single core group of experts, whether an academic research group or a particular company, is centralized by nature. In a distinct contrast, the Semantic Web is hoping to propagate itself much as the hypertext Web was propagated, by allowing anyone to create knowledge representations and post them on the Web for others to find and use. In contrast with the centralized nature of classical artificial intelligence, the Semantic Web is counting on the decentralized creation of knowledge representations. Best practice on the Semantic Web does not have every person re-inventing the proverbial wheel, but re-using knowledge representations created by others as much as possible. In this manner, Semantic Web researchers correctly deduce that the creation of knowledge representation, even using a simple semantic network model like RDF, is not going to be done by absolutely everyone. Instead, Semantic Web advocates hope that industries and researchers will reach consensus on large-scale ontologies. In the spirit of MYCN [Shortliffe, 1976], the life sciences have been one of the first domains to begin standardizing ontologies such as the Gene Ontology, which can coexist and be used with similar ones such as OpenGalen.

Semantic Web knowledge representations can also be explicitly mapped to each other so knowledge representations that otherwise might remain mutually incommensurable can have human-created bridges. Automated creation of these mappings is still an active and difficult area of research, one that is unlikely to ever be fully-automated with any degree of reliability since every knowledge representation, much like human language, has its own assumptions and divisions of the world that are usually implicit and often mutually irreconcilable [Bouquet et al., 2003]. As regards linking simple knowledge representation, there is sign of success in heavily decentralized metadata creation in the Friend Of A Friend project, which uses the work of over a million users to create a huge, if simple, network of metadata that map people and their interests.

Classical artificial intelligence relied on an expert group of “wise old men” that would create the ultimate single knowledge representation - known sarcastically as “God Forthcoming” - that stuffs the essence of the world itself into bits and bytes. There are few advocates of the Semantic Web that believe that through sheer decentralization “God Forthcoming” will be revealed. The Semantic Web is more pragmatically hoping that people will bit by bit use knowledge representations that they find useful, and there will be domains of emerging consensus and domains of unending opposition, but that overall the network effect of the Web will lead the Semantic Web to be created.

7. FORMAL SEMANTICS

One problem with traditional artificial intelligence was the lack of an agreed upon foundation with well-understood properties. Usually ontologies were created by small research groups, with each group having its own form of knowledge representation and with its own “semantics,” although

---

5 http://www.geneontology.org/
6 http://www.opengalen.org
7 http://www.foaf-project.org
almost all representational schemes were found to be equivalent to first-order logic. OWL (and to some extent RDF) are based on logics are well-studied in both academic and industrial use, with OWL closely modeling itself on widely studied earlier projects such as CLASSIC and KL-ONE [Borgida et al., 1989, Brachman and Schmolze, 1989], while maintaining decidability. Decidability of utmost importance on the Web, for if one is literally querying the open-ended world of the Web, actually getting an answer back becomes incredibly important. Decidability is usually shown through proofs involving the formal semantics of the knowledge representation language. However, besides this technical advantage, is there anything else that makes formal semantics special?

Formal semantics avoid the problem of *derivative semantics*, where the meaning of any syntactic statement is ultimately grounded out in human interpretation, a classic philosophical argument against the possibility of machine semantics [Dretske, 1985]. That is not to say that the Semantic Web throws human interpretation to the wayside, but it is not dependent on human interpretation. The question of exactly how formal semantics accomplishes this goal is not well-understood by many, primarily as it uses the double-edged sword of placing one level of mathematical abstraction over the domain (content) and then proceeds to use this abstraction to implement the semantics in a program, and so make the mathematical abstraction physically effective and independent of the domain modelled.

Formal semantics accomplishes moving meaning from the realm of human interpretation by defining semantics in terms of a *models* defined using mathematics. The *interpretation function* maps from the syntax, a collection of physical marks, to the semantics, a mathematical model that stands in for the “real” world. In formal semantics, the interpretation function is a substitute for human interpretation, is illustrated in Figure 4 from Brian Cantwell Smith.

![Figure 4: Syntax, Models, and Semantics](image)

As one can see, since the real world is out of causal reach of most computers, the best they have is a mathematical model that can “stand-in” for the semantics of the real world, and is causally independent of them once it has been formalized. The creation of formal semantics can be done in a number of ways. One of the most common is *denotational semantics*, mapping syntactic structures to well-understood formalisms like set theory or first-order logics. However, the mapping can often be made much more compressed and implicit using *axiomatic semantics* that lay out a relatively small number axioms that in turn define a much richer model. Another possibility is to use *operational semantics* that maps the allowed (operations) in the syntactic system to properties that can be inductively proven. In formal semantics when one mentions the word “semantics” of a symbol like “cat” one is no longer allowed to think of a black cat that purrs and warms itself underneath the covers on long winter nights, and all the informal human meaning that entails, but one can only only with certainty declare the mathematical consequences that string of characters entails, such as a particular set or a particular proof.

One might question the value of this obsession with formality. Is it not just the logicist program of AI revived? There is a key difference that should not be elided. In the classical AI programme, one supposes that real intelligence of the same kind as embodied in humans can be defined using sets, proofs, and other mathematical devices. While it perhaps has not been proven conclusively dead, the prognosis for this outlook is glum, and few people are willing to bet the house on such as logicist view of human intelligence given the rise of the “intelligence-without-representation” paradigm as championed by Brooks and others [Brooks, 1991]. On the Semantic Web, these formal semantics is not making any claims about human intelligence. Formal semantics are primarily good engineering. When one is specifying a standard for some totalizing knowledge representation or other system, one does not want to specify the exact running code people should be installing, which is exactly what happened with software like Windows. Imagine whether or not the Web would have taken off had Berners-Lee forced everyone on the Web to use his particular web browser on a NeXT machine; obviously the Web would have failed as a universalizing technology. Instead, it succeeded precisely because Berners-Lee, Masinter, and others specified the Web at one level of abstraction above code: the technology was crated at the level of a specification that could in turn be implemented in many different programming languages and over many different environments. How does one ensure that in these diverse environments with differing languages, this piece of running code is actually implementing the specification? One can make test-cases that test each feature, judging the results of the particular code versus results the specification says it should have produced. In the open world of digital computing one can never completely specify exactly the environment what code is running in, and so the number of test cases is innumerable infinite. One could hope to rely on the clear prose of the specification. Yet human language is ambiguous to the its very bones, and even the most well-crafted of prose will be open to misinterpretation of the author’s intent, which while productive in literary theory will spell doom to getting programs to work in a uniform manner. One reason why increasingly specifications have to be made at the level of abstraction is because mathematics is the closest thing humans have to a precise lingua franca; formal semantics goes above the level of prose and can specify with certainty regardless of any situation the behavior of a program.

Formal semantics is also the perfect lingua franca between humans and machines. While it may be forever unclear if a computer can know what it feels like to be a bat or even
understand children’s stories, because of their heritage in mathematical logic computers are ideal machines for following formal semantics. Due to their huge memory and limitless patience, in the world of pure formalism computers far outmatch humans in unerringly deducing the results of a formal semantics. The key difference is that the “semantics” of humans (and supposedly artificial intelligence) is grounded in the perception of the world around them, while on the Semantic Web semantics are grounded in mathematical models with the mapping from useful domains to models done by humans. This is the true art of the knowledge engineer, that without prior commitment a human knowledge engineer must discover what formalism best has properties analogous to the domain being modelled and is well-suited to the task at hand [Gruber, 1995].

Besides just good engineering and taking advantage of the nature of digital computing, the semantics of formal semantics is not derivative precisely because once the formal semantics has descended from the world of mathematics and been exteriorized by programmers as running code inside the machines, the semantics of the machine operation is independent of the human. Imagine if all the humans were destroyed by some sort of neutron bomb and the computers kept running. The programs would continue to behave the same, exactly as specified in their formal semantics, without any human interpretation. Unlike any other program just hacked together from whatever code ended up working, the behavior of these programs is normative and rule-governed in a manner independent of human judgement, governed by the norms described by the formal semantics. So the semantics on the Semantic Web are not purely an epiphenomenon, insofar as these formal semantics are embodied in conformant and running programs. These real running programs can have real impact on the world, and so increasingly in domains like flying planes, companies are relying for safety’s sake on careful formal semantics. The life of very real people and very real aircraft may depend on exactly how some running code conforms to the abstract notion of meaning given by formal semantics, so to dispute the causal efficacy of formal semantics reveals a latent vulgar Cartesian divide between computers and the “real world.” The semantics on the Semantic Web has meaning, and it’s not just in our head - it’s in our machines, and the line between us and our machines is disappearing every day.

8. THE KNOWLEDGE REPRESENTATION PROBLEM

At this point in our analysis, despite the superficial similarities, the Semantic Web seems to be on far surer ground than that classical artificial intelligence by virtue of its scale, its decentralized nature, and its focus on a common formal semantics. The devil is in the details though, for formal semantics passes the buck of connecting the representation on the Web to the world back to the knowledge engineer. How does a person choose what knowledge to represent on the Web? As much as the Semantic Web effort has made careful and web-scale improvements over the foundations of knowledge representations used traditionally in AI, the Semantic Web is inheriting one of the more dangerous problems of artificial intelligence, one which needs to explicitly recognized otherwise the Semantic Web will return to the problems of AI “the first time as tragedy, the second as farce” [Marx, 1852].

This thorny issue facing the knowledge engineer can be called the Knowledge Representation Problem. If knowledge representations are fundamentally stand-in surrogates for facets of the world, then “how close is the surrogate to the real thing? What attributes of the original does it capture and make explicit, and which does it omit? Perfect fidelity is in general impossible, both in practice and in principle. It is impossible in principle because any thing other than the thing itself is necessarily different from the thing itself.” [Davis et al., 1993]. Imperfection in knowledge representation is not just an aesthetic problem, for it can lead to very real errors since “imperfect surrogates mean incorrect inferences are inevitable” [Davis et al., 1993].

Is there any a priori reason why we cannot have perfect knowledge representations? Knowledge representations are the exteriorization of our understanding of some portion of their world onto a digital substrate. In other words, the fuzzy and vague understanding inferior to “our head” (or our phenomenological experience) must be given digital flesh and so exteriorized into bits and bytes. The key in this is not exteriorization, but digital, for digitality by definition always forces one to abstract away from the infinite fuzziness of the analogue world, and so something, an ineluctable difference, is always lost in translation from interior understanding to exterior knowledge representation due to the digitality of the external representation. Therefore, one can never fully obtain “God Forthcoming” in a digital substrate.

The set of classical AI was that while irreconcilable differences exist in human relations on subjects like politics and religion, the vast majority of the world as humans experience it is shared and relatively non-controversial. This may seem to be an immense injustice to the various subject matters one may want to matter, it is a purposively mundane engineering decision. How is one supposed to model intelligence if one is constantly getting caught up in questions of semiotics, such as whether or not my conception of “cat” differs from yours? In the open world of the Web and its attendant diversity of cultures, the set of everyone having the same naïve ontology is increasingly hard to uphold.

The scale of the Semantic Web gives a special twist to the Knowledge Representation Problem that didn’t face the cloistered researchers of classical artifical intelligence. With a decentralized method of creating knowledge representations, it becomes increasingly difficult to guess what features of the world people might formalize into an ontology. This will lead to many ontologies that are about the same domain, yet it can be impossible to tell on the level of domain whether or not components of similar ontologies are equivalent. As iterated earlier before, there is no unambiguous human-understandable prose that can show two knowledge representations to be equivalent, so the task of mapping be-

9There are potentially interesting questions about the simulation of analog phenomena on a digital substratum, and while a priori there will still always be some loss, there may be limits to the ability to humanly perceive these losses.

10One could even defend this standpoint from an evolutionary point of view, saying that as we humans have in general evolved together to survive on the same world, and so by nature we will share the same common-sense ontology about the world. In some cases this may be even true, yet time and time again, we find that culture has outstripped biological evolution, and that our cultural understanding of things that we assume is ontologically very different.
tween many small ontologies manually is immense. One way to resolve the ontology mapping issue would be to use only a few well-specified large ontologies, yet one loses the ability to map one’s locally rich semantic space to a personalized knowledge representation. Even if one knowledge representation is perfectly suited to a particular domain, that domain is often itself in flux, so it is also hard to tell how “brittle” these knowledge representations are in the face of real-world use.

The problem is not only in the domain being modelled by the knowledge representation. Knowledge representation languages and their attendant formal semantics themselves shape the nature of the domains they represent, since languages force certain semantical commitments that do not exist in the domain [Gruber, 1995]. One example is the distinction between individuals and classes in ontology languages like OWL. This particular division of abstraction is both a benefit and a curse for the Semantic Web, for people tend to have different intuitions about how to use abstraction divisions between classes and individuals [Smith, 1996]. The “Dartmouth School of Art” can be thought of as a concrete instance of the class of all schools, or as a subclass which remains the same regardless of the moving of the physical building or the change of staff. It then becomes unclear what one is referring in statements such as “The Dartmouth School of Art is now specializing in sculpture” or “The Dartmouth School of Art has changed its address.” The OWL documentation mentions both that “in certain contexts something that is obviously a class can itself be considered an instance of something else” and “it is very easy to confuse the instance-of relationship with the subclass relationship” [Welty et al., 2004].

Should we give up all hope in knowledge representation, and so the Semantic Web? Or of course not. While fantasies like “God Forthcoming” are certainly not forthcoming, small scale knowledge representations that model certain parts of the world certainly are forthcoming. In contrast to the ontological claims made traditionally by knowledge representations in AI, knowledge representations should be judged not whether or not their model their domain in some context-free manner, but whether or not the knowledge representation accomplishes some useful task in a certain context. When working with knowledge representations, abstraction is both a blessing and a bane. On one hand it causes knowledge representations to always be less than their particular domain, but on the other hand these knowledge representations can emphasize the important characteristics of the domain in the context of a given problem. When one is trying to do something as simple as merging address books, a person’s political beliefs or eye color is usually not important, yet disambiguating names becomes of utmost importance and so is brought to the forefront of the knowledge representation. In this case, a knowledge engineer would surrender on the deep philosophical problem of defining what a person qua person means, but focus their attention on “what are the important characteristics of a person in an address book.” It is this utility-driven mode that the Semantic Web sees massive deployment of knowledge representations. This is not unrealistic, for as more and more of our information exists in digital form, it seems to make sense that the field of knowledge representation would return from the proverbial grave.

9. THE FRAME PROBLEM

The question of how to represent time in knowledge representations is another problem from artificial intelligence that haunts the Semantic Web. RDF attempts to avoid this problem by stating that it “does not provide any analysis of time-varying data” [Hayes, 2004]. Yet, it would seem that any statement, even on the Web, is not meant to last forever, especially as URIs and their contents have a tendency to change. Berners-Lee attempts to avoid this problem in a note “Cool URIs don’t Change,” [Berners-Lee, 1998] in which he notes that the change of a URI damages its ability to be universally linked and have statements made about it. Despite this principle being made fundamental in new Web standards, it at the current moment does not stand true about the Web and we have no reason to believe that it will soon in the future [Jacobs and Walsh, 2004]. There is already a need to make temporally-qualified statements using the Semantic Web.

The notorious Frame Problem is the problem of how to represent changes over time in representations [McCarthy and Hayes, 1969]. It was discovered in the context of trying to formalize all the common-sense facts about a given situation using a version of first-order predicate logic called the situation calculus that formalized common-sense notions about actions holding over time and counterfactual sentences by means of logical fluents. These fluents, like $F(\pi, s)$, state for some situation $t$ in the future of $s$, $\pi(t)$ holds. So, if one is in the situation where “a person has a telephone, he still has it after looking up a number in the telephone books” [McCarthy and Hayes, 1969]. The problem is then that “if we had a number of actions to be performed in a sequence, we would have quite a number of conditions to write down that certain actions do not change the value of certain fluents.” A frame can be introduced so that “a number of fluents are declared as attached to the frame, and the effect of an action is described by telling which fluents are changed, all others being presumed unchanged” [McCarthy and Hayes, 1969]. These frames would presumably map to certain “common-sense” situations, like “being in a restaurant” or “dialing a phone number.” We call the fluents and the axiomization of their changes in the face of possible actions the frame axioms in regards to a particular frame.

Frame axioms are difficult on two levels. First, as the knowledge representation is abstracting from the world, by its very nature knowledge representations can not possibly represent all the dependencies and possible effects of an action in the domain being modelled. Second, the very notion of frames discretely portion out limited situations in the world, and these situations themselves change over time and may not be finite. Any change in the domain over time itself cannot be specified completely without changing the frame itself, and to do so leads one into an infinite regress of frame axioms. So for every possible situation one must create new frame axioms, and furthermore these frame axioms are dependent on the part of the world modelled by the knowledge representation. In general this problem can be skirted by simply trying to best mimic human reasoning by making some a priori division of the world into frames and modelling what the knowledge engineer believes are only

---

11In other words, missing key components as to make them unusable.

12Especially when two different people have the same name!
The “relevant” results of an action. Artificial intelligence is trying to create finite human-level intelligence, not the mind of God. Yet while there are a number of technical solutions and improvements to the original situation calculus in classical AI, none of them can claim to absolutely escape the problem of the infinite regress of possible frames, but only to “work” under certain circumstances [Simon, 1957b].

The Frame Problem has been taken from its narrow, technical definition as given by McCarthy and Hayes and reworked (perhaps incorrectly) into a broader problem by philosophers. In this context, the “Frame Problem” be thought of not only as a problem of logically describing the effects of temporal actions in a particular logic, but also a problem of knowing the possible results of any action, and so the computational tractability of any action. In computational viewpoint, it a problem about how to deduce the consequences of any action without having to also explicitly search through all the elements in the knowledge representation are not effected by the action. This version of the Frame Problem, unlike the axiomization problem put forward earlier, is not only a problem in logic-based AI, but a problem for connectionist and other “non-representational” forms of AI [Haselager and Rappard, 1998]. The general issue can even by generalized to humans and is phrased by Fodor as the problem of isotropy, as a problem is “isomorphic” when there is no bound to what could possibly be relevant to the problem at hand [Fodor, 2000]. In the analog world outside knowledge representations, anything can be relevant to an action, and actions have unintended consequences. As stated by Fodor, the Frame Problem is “Hamlet’s Problem” of “when to stop thinking” [Fodor, 1987].

There are a number of escape routes from the problem. The first is called the closed world assumption, pioneered in a deductive theorem prover used to maneuver a robot around the world [Fikes and Nilsson, 1971]. Since the robot had a fairly limited amount of time to make a decision to maneuver about the world in, it made the following assumption: if you cannot prove P or ¬P to a collected of known facts, add ¬P to the database of known facts. This is also known as “negation-as-failure.” Finally, in addition the closed world assumption can be generalized to the common sense law of inertia, which means that one should assume that when something changes, nothing else changes. This closed-world assumption has been tremendously popular, and is the assumption present in nearly all database systems. Yet this closed world assumption is perfectly justified only if one has somehow solved the knowledge representation problem and has a complete and accurate representation of the world, which is a hard problem by itself. So, at best the closed world assumption is, as was used by Fikes and Nilsson, a way of getting things done with imperfect representations and having to make large assumptions about the world. This is likely as what humans do, as work on bounded rationality has shown again and again [Simon, 1957c]. A few other lines of flight exist from the closed world problem, such as using a dynamic systems that build temporality directly into their operation [Wheeler, 2005].

10. THE OPEN WORLD

One thesis put forward that the Semantic Web languages would operate exactly the same if the identifiers in them were not URIs at all, and therefore if the Web did not exist [Hayes, 2006]. If we accept that proposition, then the Semantic Web is no different from classical artificial intelligence, a project whose progress has at best been dismally slow and at worst been philosophically misguided. The key difference between the Semantic Web and classical artificial intelligence isn’t the use of knowledge representations at all, the key difference is the use of Universal Resource Identifiers (URIs). Upon surface consideration, it is not clear what this would change at all, since replacing one arbitrary symbol with another arbitrary symbol beginning with a URI scheme like “http:///” would seem make no difference. However, choosing one symbol over another is not arbitrary in the least. When using a URI in a knowledge representation, one is grounding the knowledge representation in the Web in a very physical and causally effective manner. One could in turn argue that in terms of the symbol-grounding problem [Harnad, 1990], this would be grounding one symbol in merely a never-ending cycle of endless symbols, each without any genuine semantics. This presumption, the “Syntactic Web” as Harnad put it, we believe assumes an arbitrary division between a world of meaning, dividing the world in two. One world is our world outside computers and the Web and another is the world of ceaseless and meaningless syntactic churning inside computers or on the Web [Harnad, 2002]. This division is incorrect and reminiscent of some sort of vulgar Cartesian divide between mind and body, with the operative terms instead being our human selves and our digital computers. As our computers entangle themselves more and more in our lives, we find our meaning is often invested in our digital computers.

What about URIs on the Web make them different from any other mere syntactic symbol? One key is that URIs do indeed have causally and physically effective properties. This is called the follow-your-nose principle: a URI is not just an identifier, but allows one to accessing more information about that identifier. When one has a URI, one has the possibility of accessing evermore bytes, and running programs, with all the real consequences these entail. So, when one accesses one URI, one may in turn find more URIs that one can follow recursively. This is important, for under the normal operating conditions of the Web, the use of URIs makes knowledge representations “self-describing.” When one receives a collection of symbols purporting to be a knowledge representation, one can follow the URIs to determine what language formalism it uses, what greater family of knowledge representations it may be part of, human-readable documentation, programs that one can run, and remember that every URI is fundamentally owned by some party - how to contact the community or author responsible for the knowledge representation.

The power of using URIs is precisely because they are universal symbols, so that these knowledge representations can be used by anyone. By virtue of using URIs in a knowledge representation, a desperate late-night hacker can find a pragmatic solution to some problem, such as moving data from his mobile phone to his address-book and opens up his solution to the entire world. URIs gives knowledge representations the means to be accessed from anywhere with web access, and made accessible by simply putting their knowledge representation on the Web. If the solution is genuinely useful, others will likely take up the knowledge representation. There is also a huge difference between universality in theory and universality in practice, and in a pragmatic sense the Web increasingly encompassing more and more
information constantly, capable of devouring other smaller information-sharing systems, and is actually deployed by the vast majority of computers on the planet. So, one actually has deployed web-browsers and well-test code to retrieve the information that the URI identifies. The use of URIs then also makes sense, since by naming every concept simply by a URI, the Semantic Web lets anyone express new concepts that they invent with minimal effort” [Tim Berners-Lee and Lassila, 2001]. As long as one owns one computer with a URI or access to one, one can then mint an infinite number of identifiers for the URI by merely appending yet another string hierarchically to the URI. Therefore, if one is to press upon what fundamental difference a URI makes to knowledge representation, it is that Semantic Web knowledge representations exist in a universal information space. The universal space of the Web has impact upon how knowledge representations actually do work on the Web. By “work” we mean how one chooses a formal semantics for the Semantic Web, and so what deductions and inferences should be made by any given knowledge representation. While traditional AI systems solved the Frame Problem by operating by a “closed-world” assumption, the Semantic Web operates by the reverse assumption. The Open World Assumption states that in a universal space there is always the possibility of new information. In other words, on the Web reasoning “needs to always take place in a potentially open-ended situation: there is always the possibility that new information might arise from some other source, so one is never justified in assuming that one has ‘all’ the facts about some topic” [Hayes, 2001].

This severely impacts the ability of reasoning systems. Traditionally, in classical artificial intelligence, the goal of reasoning has been non-monotonic reasoning, reasoning where the results of adding new knowledge may lead to previous inferences being withdrawn. One may believe in Euclidean geometry that two parallel lines never intersect. However, if one learns that this assumption is not true, then one may have to subtract theorems that rely on this assumption. In a more everyday example, the type of reasoning often used by humans is default reasoning. Default reasoning is also non-monotonic, as one can withdraw assumptions and have new ones in the face of new information. If I know that if you’re a soldier, you will obey the commands of your officer. Yet, if I later discover that you’re a soldier that’s mutinying, then I can no longer assume you’ll be obeying the commands of your officer, and so have to retract that deduction. Since in the open world you never know what new information might arise, the Semantic Web has so far used monotonic reasoning, where one should never have to withdraw a deduction in the face of new information.

The open world assumption has even more dramatic effects upon the types of logic that are suited for the Semantic Web, for the notion of false in classical logic can not be up-held in an open world. One of the principles of logic is the Law of the Excluded Middle, the Aristotelian “Tertium Non Datur,” that states that any given statement is either true or false. In propositional logic, for any given proposition $P$, either $P \lor \neg P$. If one discards the Law of the Excluded Middle, one must also discard proof by counter-example, and therefore much of classical mathematics. One option is a logic without the notion of truth. Developing mathematics without the Law of Excluded Middle leads one to intuitionistic mathematics as developed by Brouwer [Detlefsen, 1990], which in turn serves as the basis for constructivist logic and the analysis of recursion. Philosophically, in intuitionistic mathematics one does not even strictly have a notion of truth, but instead has a notion of proof where anything that is “true” only in relation to a proof of its existence. Similarly, one no longer has a notion of false, but only those facts for which one does not have proofs, and so one can imagine that any fact can still be true. This is a barely tapped tradition for the Semantic Web, and since the Semantic Web has until this point confined itself to the simple semantic networks of RDF where everything is true, and OWL where things are only classified, not proven true or false. As the Semantic Web advances into the realm of trust, and the need for more expressive logics and proof systems arises, intuitionistic mathematics may be found naturally suited to the Web.

11. THE PHILOSOPHY OF THE WEB

In contrast to the centralized, small-scale, semantically heterogeneous, closed-world of classical artificial intelligence, the Semantic Web is a decentralized open-world with the potential for massive scalability due to its reliance on a universal information space and a common set of standards for formal semantics. The future is not all bright, for the Semantic Web has not escaped the clutches of the Knowledge Representation Problem and the Frame Problem. Yet this is not the fault of the Semantic Web per se, these problems are endemic to knowledge representation as a whole, regardless of whether they are on the Web or not. Despite the technical differences, is it still fair to say that the Semantic Web is an ancestor of classical artificial intelligence? The answer is not technical but philosophical.

As researchers from classical artificial intelligence have now turned towards a philosophy of embodiment as a way of solving the Knowledge Representation and the Frame Problem. To them, the resemblance between classical AI and the Semantic Web will be a cause of concern, for the Semantic Web seems precisely the opposite of the movement towards embodiment in artificial intelligence. The Semantic Web focuses instead on knowledge representations, logic, universality: areas that the mainstream of “embodied” AI research has surrendered on. The action in embodied artificial intelligence is focused on understanding perceptual and motor-based coupling between artificial organisms and their environments [Brooks, 1991]. Embodied AI mechanisms tend to work by relying on the exact physical configuration of the body and exploiting usually unnoticeable features of the environment. This is precisely the hallmark of evolution: fast, cheap, and efficient [Clark, 1997]. These are features of how our intelligence works that are unamendable to any sort of introspection, and only reveal their secrets under the glare of empirical experimentation.

Is it not ironic that just as artificial intelligence turns towards embodiment, the our lives are increasingly disembodied? The coalescence of the terms of “embodiment” and “intelligence” is misguided, for it presumes an essentialist reading of the “body” as somehow the ultimate limit of our intelligence, as if for some reason intelligence stopped at the skin. Instead of the mind being trapped “inside our heads” in some Cartesian nightmare, the mind has now stretched to the very skin of the body itself. As the “Extended Mind” hypothesis points out [Clark and Chalmers, 1998], there is
little reason to believe that our cognitive functions are in fact limited to the boundaries of our skin. It is the ability to expand our cognitive abilities to objects outside our human bodies that forms the basis for **human augmentation**, a project that is both earlier than and strangely parallel to artificial intelligence [Engelbart, 1962]. According to its intellectual progenitor Douglas Engelbart, “by augmenting human intellect we mean increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems. Increased capability in this respect is taken to mean a mixture of the following: more-rapid comprehension, better comprehension, the possibility of gaining a useful degree of comprehension in a situation that previously was too complex, speedier solutions, better solutions, and the possibility of finding solutions to problems that before seemed insolvable” [Engelbart, 1962]. The intellectual child of Douglas Engelbart, the project was not an attempt to replicate human-level intelligence, but to instead to augment human intelligence through digital technology.

The evolution of digital technology is the logic of human’s overcoming their own biological heritage. Far from replacing or replicating human intelligence, digital technology fundamentally complements human intelligence. With rare exceptions, humans are in general not skilled at actions that computers can routinely perform like complex arithmetic, keyword searching over massive amounts of text, and visualizing large data-sets. Most importantly, computers can store information with perfect fidelity and instant recall. It is not accident that as human-created artifacts digital computers excel precisely the areas that humans do not. The philosophical project of classical artificial intelligence in retrospect seems strangely perverse, as it is an attempt to re-build human intelligence on a material base, the digital computer, that was created precisely to enable the opposite of human intelligence. Thus the focus on biological realism and the movement towards exploring artificial evolution in distinctively differing substratum, such as electro-chemical computing and artificial intelligence. The motto of “First the earwig, then man!” is in fact a suitable rallying cry [Kirsch, 1991]. From a sheerly pragmatic point of view, while there are uses for evolutionary robotics both in the philosophical project of understanding ourselves and more mundane uses in the the military, those more inclined to pursue human augmentation should have a certain sympathy for classical artificial intelligence. Walking robots are not that interesting, since we already walk!

The Semantic Web is not artificial intelligence redux, since the Semantic Web does not share the philosophical project of artificial intelligence, that of creating human-level intelligence. The Semantic Web is instead a turn-of-the-millennium movement away from artificial intelligence and back to the goals of Engelbart’s human augmentation project and “keeping humans in the loop.” This break with classical artificial intelligence and towards human augmentation bodes well, for while Engelbart’s personal implementation of his vision in the NLS (oXline System) floundered, as a guiding vision it has been remarkably successful. Human augmentation has played a crucial role in technologies as diverse as the Internet, where Engelbart’s NLS was the first Network Information Center, and personal computing, where at Xerox PARC many computer scientists - some former researchers for Engelbart - implemented much of what is today called “personal computing.”

The connection between the Web and Engelbart runs deep, as both were inspired by Vannevar Bush’s “As We May Think” idea of the Memex [Bush, 1945]. Furthermore, when Berners-Lee’s original academic paper describing the World Wide Web was rejected by the Hypertext 1994 conference, he met with Douglas Engelbart, and many of the conceptual underpinnings of the Web were directly appropriated from NLS. The term **resource** was originally in NLS, as was the “hash” mark used to denote “fragments” of a resource. However, the Web also corrected many of the problems that sunk the original NLS, such as constraining users to a strict hierarchical structure or an arcane user interface. Just as the World Wide Web has appropriated many of Engelbart’s ideas from NLS, the Semantic Web is harvesting ideas from the last few decades of artificial intelligence and re-implement them on the Web. The logic and reasoning systems developed by artificial intelligence have been removed from their philosophical project and now re-appropriated for a main competing philosophical project, human augmentation. This is evident in the words of Dan Connolly, the co-inventor of HTML with Berners-Lee, “the bane of my existence is doing things that I know the computer could do for me” [Connolly, 1998].

There are further reasons to be glad the Semantic Web is not artificial intelligence. At both the heart of both classical and embodied artificial intelligence is an essentialist reading of the human. Ignoring the assumption that intelligence can be created on a digital machine, artificial intelligence believes that whatever the great secret of intelligence is, it must be part of being human. Cognitive science that shows the true secret to intelligence is the off-loading of difficult tasks into the environment [Clark, 2004]. For all the focus on embodiment, the ghosts of Strawson’s material individuals or the sets of Frege that lie at the heart of analytic philosophy, still plague our understandings of intelligence [Frege, 1879, Strawon, 1959]. Intelligence does not exists in any isolated individual, be it idealized individuals of classical logic or the all-to-human individuals of embodied cognitive science. Intelligence is in the ebb and flow of interactions with the environment, an environment that is both shaped by the human and shapes the human back. This is exemplified not by the artificial intelligence project as a whole, but by the spirit of the human augmentation project as exemplified in general by digital computing and in particular by the Semantic Web.

Even the project of human augmentation privileges the human subject more than strictly necessary, for the lines at any given instant drawn between humans and their machines are more one of our own convenience that of any deep division in nature itself, as Heidegger’s own mediations on an axe showed [Heidegger, 1927]. Despite our instinct against it, when one deprivileges the human subject as an transcendental individual, one opens the door for new subjects, subjects composed of what were formerly thought of as individuals. This new subjects are collective subjects, and the act of overcoming the artificial divisions between formerly individual subjects and creating new subjects is the act of collectivity. When this collectivity is put into practice, not as an abstract or utopian notion, but in confrontation against the concrete and practical problems that threaten these new subjects, collective intelligence is born.
Through the lens of collective intelligence the Web makes much more sense as a “a common space in which we could all interact.”\textsuperscript{13} For collective intelligence is not a new phenomenon, but an immensely successful evolutionary survival strategy as old as language itself, which like knowledge representation is an exteriorization of thought. What digital computing brought to the play is the ability for our knowledge representations to be no longer strictly dependent on our interpretation, but having their own physically effective powers. The Web revolutionized these developments by putting our representations in a universal space, allowing the network effect of the Web to drive \textit{collective human augmentation}, which is augmenting our collective as opposed to individual intelligence. By definition, this must involve ever more augmentation and exteriorization of external representations, a variation of human augmentation only dimly perceived by Engelbart. The rapid growth of collective web applications that leverage the power of the Web as a universal information space, ranging from Wikipedia to social tagging, are just the beginning of the trajectory enabled by the Web. What is the role of the Semantic Web in this? For too long data has been hobbled by proprietary data formats and incompatible software, preventing the collective use of representations. Equally so, for too long knowledge representation has been hobbled by its attachment to the philosophical project of artificial intelligence. In contrast to the flash of digital “new” media, the Semantic Web operates in the basement unseen by human users, yet providing the substrate of ever more powerful and useful knowledge representations that will serve as the nuts and bolts of universal collective intelligence.

12. REFERENCES


Barnes-Lee, T. (2003). Message on www-tag@w3.org list. Message on www-tag@w3.org list.


Gruber, T. (1995). Toward principles for the design of


http://www.w3.org/TR/webarch/.


