The Open World Assumption: Elephant in the Room

by Mike Bergman - Monday, December 21, 2009

http://www.mkbergman.com/852/the-open-world-assumption-elephant-in-the-room/

OWA Enables Incremental, Low-risk Wins for the Semantic Enterprise

In speaking of the semantic Web, it is not infrequent that the open world assumption (OWA) gets mentioned. What this post argues is that this somewhat obscure concept may hold within it the key as to why there have been decades of too-frequent failures in the enterprise in business intelligence, data warehousing, data integration and federation, and knowledge management.

This is a fairly bold assertion. In order to support it, we first need to look to the logic and mindset assumptions associated with traditional relational data management and the semantic Web. We then need to look to the nature of knowledge itself and its relation to data federation. It is in this intersection that the key of decades of faulty premises may reside.

The main argument is that the closed world assumption (CWA) and its prevalent mindset in traditional database systems have hindered the ability of enterprises and the vendors that support them to adopt incremental, low-risk means to knowledge systems and management. CWA, in turn, has led to over-engineered schema, too-complicated architectures and massive specification efforts that have led to high deployment costs, blown schedules and brittleness.

The good news is that abandoning these failed practices and embracing the open world approach can be done immediately based on existing assets. Simply shifting from the closed world to open world premise can, I argue, improve the odds for enterprise IT success in these areas.

It is time to meet the elephant in the room.

Scope and Some Root Causes of Enterprise IT Failures

It is, of course, a bit of editorial hyperbole to label most enterprise initiatives in business intelligence and knowledge management as being failures over the past few decades. And, insofar as failures have occurred, I also do not believe they are the result of vendor greed or cynicism, or IT management mistakes or incompetence. Rather, I believe the fault resides in the attempt to pound a square peg (relational model) into a round hole (knowledge representation).

The scope of these failures is not known. We have seen anecdotal claims of trillions of dollars in annual loses due to IT project failures worldwide; failure rates for major IT projects in the 65% to 80% ranges; and analysis of waste and failures in individual firms that are fairly eye-popping [1]. The real point of this post is not to try to quantify these problems. However, in my many years within IT it has been a common perception and concern that many -- if not most -- large-scale information technology deployments have disappointed in one way or another.

These disappointments range from cost overruns, to late delivery, to unmet objectives, or to low user acceptance. Many initiatives are simply cancelled before any such metrics can be documented. Whatever the absolute quantification, I think most experienced IT managers and executives would agree that these failures and disappointments have been all too commonplace.

> ***"Business Intelligence projects are famous for low success rates, high costs and time overruns. The economics of BI are visibly broken, and have been for years. Yet BI remains the #1 technology priority according to Gartner."***[2]

Why might this be?

I truly believe the reasons for these disappointments do not reside in bad faith or incompetence. The potential importance of IT knowledge projects to improve competitive position, lower costs, or aid innovation for new markets is understood by all. Dilbert aside, I find it simply incomprehensible that disappointments or failures are rooted in these causes.

Rather, I suspect the root cause resides in the success of the relational model in the enterprise.

As transaction systems and for modeling narrowly bound and structured domains (such as products, inventory or customer lists), the relational model and its proven and optimized RDBMs and SQL query language have been resounding successes. It is natural to take a successful approach and try to extend it to other areas.

However, beginning with data warehouses in the 1980s, business intelligence (BI) systems in the 1990s, and the general issue of most enterprise information being bound up in documents for decades, the application of the relational model to these areas has been disappointing.

The reasons for this do not reside in areas such as storage or hardware; these areas have seen remarkable improvements over the decades. Rather, the problem resides in the nature of the relational model itself, and its lack of suitability to knowledge-based problems.

Technical Aspects of OWA, Broadly Defined

I have noted the importance of the open world assumption to the semantic enterprise in many of my more recent posts [3,4]. But I, like many others, often refer to the open world assumption with facile summaries such as it means that a lack of information does not imply the missing information to be false. Yet to fully understand the implications of OWA and many of its associated assumptions, it is necessary to delve deeper.

I am using here a shorthand that poses the closed world assumption (CWA) *vs.* the open world assumption (OWA). Actually, the data models behind these approaches (Datalog or non-monotonic logic in the case of CWA; monotonic in the case of OWA [5]; OWA is also firmly grounded in description logics [4]) tend be coupled with a few other assumptions. I use the shorthand of relational approach *vs.* (open) semantic Web approach to contrast these two models.

There are instances where the relational model can embrace the open world assumption (for example, the null in SQL) and there are instances where semantic Web approaches can be closed world (as with frame logic or Prolog or other special considerations; see conclusion). But, as generally applied and as generally understood, this contrast between typical relational practice and the semantic Web (based on RDF and OWL) tends to hold.

From a theoretical standpoint, I have found the treatment of Patel-Schneider and Horrocks [6] to be most useful in comparing these approaches. However, the *Description Logics Handbook* and some other varied sources are also helpful [7,5]. Much of the technical aspects summarized in the table below are from these sources; I refer you to these sources for more informed technical discussions:

Relational Approach	(Open) Semantic Web Approach
Closed World Assumption (CWA) That which is not known to be true is presumed to be false; it needs to be explicitly stated as true. *Negation as failure* (NAF) is a related assumption, since it assumes as false every predicate that cannot be proven to be true. Under CWA, any statement not known to be true is false. Everything is prohibited until it is permitted.	**Open World Assumption (OWA)** The lack of a given assertion or fact being available does not imply whether that possible assertion is true or false: it simply is not known. In other words, lack of knowledge does not imply falsity. Everything is permitted until it is prohibited.
Unique Name Assumption (UNA)	**Duplicate Labels Allowed**

The unique name assumption (UNA) is premised that different names always refer to different entities in the world.	OWL allows different synonym labels to be used for the same object; same names may refer to different objects. Identity assertions must be explicitly stated.
Complete Information The data system at hand is assumed to be complete. (Missing information is often handled via the null statement in SQL, but that has been controversial and contentious in its own right.) This is also known as the *domain-closure assumption*.	**Incomplete Information** A central tenet of OWA is that information is incomplete. A corollary is that the attributes of specific objects or instances may also be incomplete or partially known.
Single Schema (one world) A single schema is necessary to define the scope and interpretation of the world (domain at hand).	**Many World Interpretations** Schema and data instance assertions are kept separate. Multiple interpretations (worlds) for the same data are possible.
Integrity Constraints Integrity constraints prevent “incorrect” values from being asserted in the relational model. It is useful for validation/parsing/data input and is related to the single model that contains only the facts asserted. Strict cardinality is used for checking validation.	**Logical Axioms (restrictions)** Logical axioms provide restrictions through property domains and ranges. Everything can be true unless proven otherwise, and multiple possible models can satisfy the axioms. This provides more powerful inferencing, though can also be unintuitive at times. Cardinality and range restrictions exhibit different behavior for objects (inferred) or datatypes.
Non-monotonic Logic The set of conclusions warranted on the basis of a given knowledge base does not increase (in fact, it likely shrinks) with the size of the knowledge base [5].	**Monotonic Logic** The hypotheses of any derived fact may be freely extended with additional assumptions. Additional assertions tend to reduce the inferences or entailments that can be applied. A new piece of knowledge cannot reduce what is known [5]. New knowledge can arise through inference.
Fixed and Brittle Changing the schema requires re-architecting the database; not inherently extensible.	**Reusable and Extensible** Designed from the ground up to reuse existing ontologies (axioms) and to be extensible. Database design and management can be more agile, with schema evolving incrementally.
Flat Structure; Strong Typing Information organized into flat tables; linkages and connections between tables based on foreign keys or joins. Strong data typing orientation.	**Graph Structure; Open Typing** Inherent graph structure, supporting of linkage and connectivity analysis. Datatypes are inherently loose, though axioms can add strong types. Datatypes treated in the same way as classes, and datatype values are treated in the same way as individual identifiers (*i.e.*, a data value is treated as referring to an object).

Querying and Tooling	**Querying and Tooling**
SQL and query optimizers well developed. Tooling well developed. Disjunction not supported; negation must be accommodated through approaches such as NAF. Sums and counts are easier due to unique name premise. Answer closure (one answer passable to a next calculation) is easier than OWA. Most tools are not suitable for any arbitrary schema.	SPARQL and emerging rule languages used for querying; performance at scale and with broad distribution a concern. Queries require contextual information for proper set selection. Negation and disjunction are allowed and are powerful constructs. Tools generally less developed. Exciting opportunities for *ontology-driven applications* working against a small set of generic tools.

In well-characterized or self-contained domains (seats on a plane, books in a library, customers of a company, products sold via distribution channels), the traditional relational model works well. A closed-world assumption is performant for transaction operations with easier data validation. The number of negative facts about a given domain is typically much greater than the number of the positive ones. So, in many bounded applications, the number of negative facts is so large that their explicit representation can become practically impossible [7]. In such cases, it is simpler and shorter to state known "true" statements than to enumerate all "false" conditions.

However, the relational model is a paradigm where the information must be complete and it must be described by a single schema. Traditional databases require an agreement on a schema, which must be made before data can be stored and queried. The relational model assumes that the only objects and relationships that exist in the domain are those that are explicitly represented in the database, and that names uniquely identify objects in this domain. The result of these assumptions is that there is a *single* (canonical) model for relational systems where objects and relationships are in a one-to-one correspondence with the data in the database [6].

This makes CWA and its related assumptions a very poor choice when attempting to combine information from multiple sources, to deal with uncertainty or incompleteness in the world, or to try to integrate internal, proprietary information with external data.

The process of describing an open, semantic Web "world" can proceed incrementally, sequentially asserting new statements or conditions. The schema in the open semantic Web -- the *ontology* -- consists of sets of statements (called axioms) that describe characteristics that must be satisfied by the ontology designer's idea of “reasonable” states of the world. Formally, such statements correspond to logical sentences, and an ontology corresponds to a logical theory [6].

Irregularity and incompleteness are toxic to relational model design. In the open semantic Web, data that is structured differently can still be stored together via RDF triple statements (*subject* - *predicate* - *object*). For example, OWA allows suppliers without cities and names to be stored along alongside suppliers with that information. Information can be combined about similar objects or individuals even though they have different or non-overlapping attributes. Duplicate checking now occurs based on the logic of the system and not unique name evaluations. Data validation in OWA systems can both become more complicated (via testing against restriction statements) or partially easier (via inference).

It is interesting to note that the theoretical underpinnings of CWA by Reiter [8] began to be understood about the same time (1978) that data federation and knowledge representation (KR) activities also began to come to the fore. CWA and later work on (for example) default reasoning [5] appeared to have informed early work in description logics and its alternative OWA approach. This heavily influenced the development of the semantic Web languages RDF and OWL. However, the early path toward KM work based on the relational model also appears to have been set in this timeframe.

We are still reaping the whirlwind from this unfortunate early choice of the relational model for KR, KM and BI purposes. Moreover, though there is quite a bit of theoretical and logical discussion of the alternative OWA and CWA data models, there are surprisingly few discussions of what the implications of these models are to the enterprise. (That is, the elephant in the room.) The next two sections tackle this gap.

The Knowledge Management Argument for OWA

The above should make clear that the relational model and CWA are appropriate for defined and bounded systems. However, many of the new knowledge economy challenges are anything but defined and bounded. These applications all reside in the broad category of knowledge management (KM), and include such applications as data federation, data warehousing, enterprise information integration, business intelligence, competitive intelligence, knowledge representation, and so forth.

Let's looks at the characteristics of such knowledge systems and why they are more appropriately modeled through the open world assumption (OWA) rather than the relational model and CWA:

- ***Knowledge is never complete*** -- gaining and using knowledge is a process, and is never complete. A completeness assumption around knowledge is by definition inappropriate
- ***Knowledge is found in structured, semi-structured and unstructured forms*** -- structured databases represent only a portion of structured information in the enterprise (spreadsheets and other non-relational datastores provide the remainder). Further, general estimates are that 80% of information available to enterprises reside in documents, with a growing importance to metadata, Web pages, markup documents and other semi-structured sources. A proper data model for knowledge representation should be equally applicable to these various information forms; the open semantic language of RDF is specifically designed for this purpose
- ***Knowledge can be found anywhere*** -- the open world assumption does not imply open information only. However, it is also just as true that relevant information about customers, products, competitors, the environment or virtually any knowledge-based topic can also not be gained via internal information alone. The emergence of the Internet and the universal availability and access to mountains of public and shared information demands its thoughtful incorporation into KM systems. This requirement, in turn, demands OWA data models
- ***Knowledge structure evolves with the incorporation of more information*** -- our ability to describe and understand the world or our problems at hand requires inspection, description and definition. Birdwatchers, botanists and experts in all domains know well how inspection and study of specific domains leads to more discerning understanding and "seeing" of that domain. Before learning, everything is just a shade of green or a herb, shrub or tree to the incipient botanist; eventually, she learns how to discern entire families and individual plant species, all accompanied by a rich domain language. This truth of how increased knowledge leads to more structure and more vocabulary needs to be explicitly reflected in our KM systems

- ***Knowledge is contextual*** -- the importance or meaning of given information changes by perspective and context. Further, exactly the same information may be used differently or given different importance depending on circumstance. Still further, what is important to describe (the "attributes") about certain information also varies by context and perspective. Large knowledge management initiatives that attempt to use the relational model and single perspectives or schema to capture this information are doomed in one of two ways: either they fail to capture the relevant perspectives of some users; or they take forever and massive dollars and effort to embrace all relevant stakeholders' contexts
- ***Knowledge should be coherent*** -- coherence is the state of having internal logical consistency. A library of books organized by the Dewey Decimal Classification *v.* the Library of Congress Classification *v.* the Colon classification system (or others) is not inherently correct or wrong, but it is important that whatever system is used be applied consistently. Because of the power of OWA logics in inferencing and entailments, whatever "world" is chosen for a given knowledge representation should be coherent. Fantasies such as Avatar and the Lord of the Rings trilogy, even though not real, can be made believable and compelling by virtue of their coherence
- ***Knowledge is about connections*** -- the epistemological nature of knowledge can be argued endlessly, but I submit much of what distinguishes knowledge from information is that knowledge makes the connections between disparate pieces of relevant information. As these relationships accrete, the knowledge base grows. Again, RDF and the open world approach are essentially connective in nature. New connections and relationships tend to break brittle relational models, and
- ***Knowledge is about its users defining its structure and use*** -- since knowledge is a state of understanding by practitioners and experts in a given domain, it is also important that those very same users be active in its gathering, organization (structure) and use. Data models that allow more direct involvement and authoring and modification by users -- as is inherently the case with RDF and OWA approaches -- bring the knowledge process closer to hand. Besides this ability to manipulate the model directly, there are also the immediacy advantages of incremental changes, tests and tweaks of the OWA model. The schema consensus and delays from single-world views inherent to CWA remove this immediacy, and often result in delays of months or years before knowledge structures can actually be used and tested [9].

To be sure, there are many circumstances where large stores of instance data and their analysis are necessary for knowledge purposes. In these cases, hybrid CWA-OWA systems (see conclusion) may make sense.

But, as these points emphasize, the general assembly and organization of knowledge is open world in nature. Trying to fit KM and related applications into the straightjacket of the relational model is folly. The relational model and CWA for KM is the elephant in the room. Three decades of failures and disappointments affirm this fact.

The Business Argument for OWA

Besides the native match of knowledge systems with OWA, there are sound business arguments for embracing the (open) semantic enterprise as well. These arguments can be summarized as lower risk, lower cost, faster deployment, and more agile responsiveness. What is there not to love?

It should now be clear that it is possible to start small in testing the transition to a semantic enterprise. These efforts can be done incrementally and with a focus on early, high-value applications and domains.

Open world does not necessarily mean open data and it does not mean open source. Open world is simply a way to think about the information we have and how we act on it. OWA technologies are neutral to the question of open or public sources. The techniques can equivalently be applied to internal, closed, proprietary data and structures. Moreover, the technologies can themselves be used as a basis for bringing external information into the enterprise. An open world assumption merely asserts that we never have all necessary information and lacking that information does not itself lead to any conclusions.

Further, we need not abandon past practices. There is much that can be done to leverage existing assets. Indeed, those prior investments are often the requisite starting basis to inform semantic initiatives. However, in leveraging those assets, it is important that the enterprise begin to embrace and understand the open world assumption.

We also see that RDF and OWL, while important behind the scenes as a canonical data model and languages for organizing this information, need not be exposed as such to most users. Most instance data can be expressed as is with the data languages of choice such as XML, JSON or whatever. We are merely using the techniques of the (open) semantic Web as the data model to organize our information assets at hand. These assets need not themselves be represented in the native RDF or OWL languages.

Thus, open world frameworks provide some incredibly important benefits for knowledge management applications in the enterprise:

- Domains can be analyzed and inspected incrementally
- Schema can be incomplete and developed and refined incrementally
- The data and the structures within these open world frameworks can be used and expressed in a piecemeal or incomplete manner
- We can readily combine data with partial characterizations with other data having complete characterizations
- Systems built with open world frameworks are flexible and robust; as new information or structure is gained, it can be incorporated without negating the information already resident, and
- Open world systems can readily bridge or embrace closed world subsystems.

One might argue, as we believe, that the biggest impediment to the semantic enterprise is the mind shift necessary to start thinking about and accepting the open world premise. Again, this perspective is not applicable to all problems and domains. But, where it is, much can be left in place and leveraged with semantic technologies, so long as the enterprise begins to look at these existing assets through a different open-world lens.

In most real world circumstances, there is much we don't know and we interact in complex and external environments. Knowledge management inherently occupies this space. Ultimately, data interoperability implies a global context. Open world is the proper logic premise for these circumstances. Via the OWA framework, we can readily change and grow our conceptual understanding and coverage of the world, including incorporation of external ontologies and data. Since this can easily co-exist with underlying closed-world data, the semantic enterprise can readily bridge both worlds.

So, we can now define the ***open semantic enterprise*** as one that embraces OWA for its knowledge management applications and engages in rapid and low-risk testing of incremental learning. The open world assumption is the proper framework to reverse decades of failure and disappointment for knowledge projects in the enterprise.

Some Open Questions about OWA

In our own discussions about ABox - TBox splits [10], we have, in essence, supported a hybrid OWA-CWA argument for the enterprise. It is beyond the scope of this current piece to describe these approaches in detail, but some of the options include local CWA, the addition of rule languages and constraints to basic OWA, use of the new OWL 2, TopQuadrant's SPIN notation, and others [11]. I will address some of these in a later post.

There are also questions about performance and scalability with open semantic technologies. Here, too, progress is rapid, with billion triple thresholds rapidly falling with daily reports of better performance [12]. Fortunately, the incremental approach that we advocate herein dovetails well with these rapid developments. There should be no arguing the benefits of a successful incremental project in a smaller domain, perhaps repeated across multiple domains, in comparison to large, costly initiatives that never produce (even though their underlying technologies are performant).

There are also architecture issues inherent in these OWA designs. In one of our next posts, we return to the topic of Web-oriented architecture and its role in support of these OWA knowledge management initiatives.

In the end, there is no substitute for doing and learning. KM based on OWA for the open semantic enterprise can be started today, in a focused manner with tangible benefits and outcomes, at low cost and risk. Let's push the elephant out of the room and let the learning and doing begin.

[1] For example, see Roger Sessions, 2009. Cost of IT Failure, September 28, 2009. This analysis suggests failure rates of 65% with a total estimated worldwide cost of $6.2 trillion in 2009. Commenters have raised questions as to what constitutes failure and have questioned some of the analysis assumptions. Nonetheless, even with over-estimates, the scale of the numbers is alarming; see Jorge Dominguez, 2009. The CHAOS Report 2009 on IT Project Failure, June 16, 2009, which indicates combined failure and challenge rates for IT projects have ranged from 65% to 84% over the period 1994 to 2009; see Dan Galorath, 2008. Software Project Failure Costs Billions; Better Estimation & Planning Can Help, June 7, 2008. In this report, Galorath compares and combines many of the available IT failure studies and summarizes that 3 of 5 IT projects do not do what they were supposed to for the expected costs, with 49% showing budget overruns, 47% showing higher than expected maintenance costs, and 41% failing to deliver expected business value; the anecdotal failure rate for years for IT projects has been claimed as 80%, with business intelligence and data warehousing particularly failure-prone areas; in 2001, a study by Mark N. Frolick and Keith Lindsey, Critical Factors for Data Warehouse Failures, for the Data Warehousing Institute noted conventional wisdom says the failure rate of data warehousing projects is 70 to 80 percent, with a then-recent study in the insurance industry found a 90-percent failure rate. This report is useful for combining many historical studies.

[2] According to this article, by Antone Gonsalves, *Poor Use Of Data Integration Tools Can Waste $500,000 Annually: Gartner* (April 27, 2009), which reports on a recent Gartner Report, large global 2000 companies, using several data integration tools with overlapping features, can reduce costs by more than $500,000 annually by eliminating redundant software and leveraging a shared services model. In a further report by Roman Stanek,

Business Intelligence Projects are Famous for Low Success Rates, High Costs and Time Overruns (April 25, 2009), Gartner is talking about a dirty little secret in the world of data integration, the fact that the data integration technology in place is based on generations of data integration technology being layered in the enterprise over the years. Thus, technology that was purchased to solve data integration problems, and reduce costs, is actually making the data integration problem more complex and no longer cost efficient.

[3] Here are some of my earlier postings dealing in some degree with OWA: Ontology-driven Applications Using Adaptive Ontologies, November 23, 2009; Fresh Perspectives on the Semantic Enterprise, September 28, 2009; Confronting Misconceptions with Adaptive Ontologies, August 17, 2009; Advantages and Myths of RDF, April 8, 2009; Making Linked Data Reasonable using Description Logics, Part 2, February 15, 2009, which specifically relates OWA to the ABox and TBox [4]; and, The Role of UMBEL: Stuck in the Middle with You . . ., May 11, 2008.

[4] We use the reference to "ABox" and “TBox” in accordance with our working definition for description logics:

"Description logics and their semantics traditionally split *concepts* and their relationships from the different treatment of *instances* and their attributes and roles, expressed as fact assertions. The concept split is known as the TBox (for *terminological* knowledge, the basis for *T* in *TBox*) and represents the schema or taxonomy of the domain at hand. The TBox is the structural and intensional component of conceptual relationships. The second split of instances is known as the ABox (for *assertions*, the basis for *A* in *ABox*) and describes the attributes of instances (and individuals), the roles between instances, and other assertions about instances regarding their class membership with the TBox concepts."

[5] **A *model theory*** is a formal semantic theory which relates expressions to interpretations. A "model" refers to a given logical "interpretation" or "world". (See, for example, the discussion of interpretation in Patrick Hayes, ed., 2004. RDF Semantics - W3C Recommendation, 10 February 2004.) The logic or inference system of classical model theory is ***monotonic***. That is, it has the behavior that if S entails E then (S + T) entails E. In other words, adding information to some prior conditions or assertions cannot invalidate a valid entailment. The basic intuition of model-theoretic semantics is that asserting a statement makes a claim about the world: it is another way of saying that the world is, in fact, so arranged as to be an interpretation which makes the statement true. An assertion amounts to stating a constraint on the possible ways the world might be. In comparison, a ***non-monotonic*** logic system may include *default reasoning*, where one assumes a 'normal' general truth unless it is contradicted by more particular information (birds normally fly, but penguins don't fly); *negation-by-failure*, commonly assumed in logic programming systems, where one concludes, from a failure to prove a proposition, that the proposition is false; and *implicit closed-world assumptions*, often assumed in database applications, where one concludes from a lack of information about an entity in some corpus that the information is false (*e.g.*, that if someone is not listed in an employee database, that he or she is not an employee.) See further, Non-monotonic Logic from the Stanford Encyclopedia of Philosophy.

[6] Peter F. Patel-Schneider and Ian Horrocks, 2006. Position Paper: A Comparison of Two Modelling Paradigms in the Semantic Web," in *WWW2006*, May 22–-26, 2006, Edinburgh, UK. See http://www.comlab.ox.ac.uk/people/ian.horrocks/Publications/download/2006/PaHo06a.pdf.

[7] Other resources include: Franz Baader, Diego Calvanese, Deborah McGuiness, Daniele Nardi, and Peter Patel-Schneider, eds., 2003. *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press, 2003. Online access to much of the book is available at http://www.inf.unibz.it/~franconi/dl/course/ ; see esp. Chapters 1, 2, 4 and 16 relate to this topic; Jos de Bruijn, Axel Polleres, Ruben Lara and Dieter Fensel,

2005. OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning for the Semantic Web, in *Proceedings of the Ninth World Wide Web Conference*, Japan, May 2005. This paper argues against the use of description logics for the semantic Web; Andrew Newman, 2007. A Relational View of the Semantic Web, March 14, 2007; Hai Wang, 2006. Frames and OWL Side by Side, presented at the 9th International Protégé Conference, July 23-26, 2006, Stanford, CA; Nick Drummond and Rob Shearer, 2006. The Open World Assumption, Powerpoint presentation at *The Chris Date Seminar: The Closed World of Databases Meets the Open World of the Semantic Web*, e-Science Institute, Edinburgh, Scotland, 12 Ocotober 2006; Yulia Levin, 2008. Closed World Reasoning, presentation at *Non-classical Logics and Applications Seminar - Winter 2008*, Tel Aviv University; and Pat Hayes, 2001. "Why must the web be monotonic?", email thread at http://lists.w3.org/Archives/Public/www-rdf-logic/2001Jul/0067.html.

[8] Raymond Reiter, 1978. “On Closed World Data Bases”, in *Logic and Data Bases*, H. Gallaire and J. Minker, eds., New York: Plenum Press, 55-76; see also, Raymond Reiter, 1980. "A Logic for Default Reasoning," *Artificial Intelligence*, 13:81-132.

[9] See this Google search on ontology-driven applications.

[10] See this Google search on ABox-TBox articles.

[11] See, as examples: J. Heflin and H. Munoz-Avila, 2002. LCW-Based Agent Planning for the Semantic Web, in *AAAI '02 Workshop on Ontologies and the Semantic Web*, AAAI Press, pp. 63–70. See http://www.cse.lehigh.edu/~heflin/pubs/lcw-aaai02.pdf (one of the first local CWA suggestions in specific regard to the semantic Web); K. Golden, O. Etzioni and D. Weld, D. 1994. Omnipresence Without Omniscience: Efficient Sensor Managment for Planning, in *Proceedings of AAAI-94* (one of the first to propose LCWA in general); Evren Sirin, Michael Smith and Evan Wallace, 2008. Integrity constraints: Opening, Closing Worlds — On Integrity Constraints, presented at *OWL: Experiences and Directions (OWLED 2008), Fifth International Workshop*, Karlsruhe, Germany, October 26-27, 2008; Timothy L. Hinrichs, Jui-Yi Kao and Michael R. Genesereth, 2009. Inconsistency-tolerant Reasoning with Classical Logic and Large Databases, in *Proceedings of the Eighth Symposium on Abstraction, Reformulation, and Approximation (SARA2009)*, July 2009; S. Gómez, C. Chesñevar and G. Simari 2008. An Argumentative Approach to Reasoning with Inconsistent Ontologies, in *Proceedings of the KR Workshop on Knowledge Representation and Ontologies* (KROW 2008), Conferences in Research and Practice in Information Technology, Vol. 90, pp. 11-20. Eds. T.Meyer, M. Orgun. Australian Computer Society, Sidney, Australia, July 2008. Holger Knoblauch, The Object-Oriented Semantic Web with SPIN, Sunday, January 18, 2009, that discusses the SPIN (SPARQL Inferencing Notation) Modeling Vocabulary, which is a light-weight collection of RDF properties and classes to support the use of SPARQL to specify rules and logical constraints.

[12] For example, the BigOWLIM can perform reasoning against 12 billion explicit statements and loads about 12,000 statements per second on a standard server; see http://www.ontotext.com/owlim/benchmarking/lubm.html; also, see Orri Erling's blog regarding performance of the Virtuoso RDF triple store (http://www.openlinksw.com/weblog/oerling/). In any case, these performance benchmarks continue to rise steadily and indicate the performance of RDF as an ontology integration layer.