Making Linked Data Reasonable using Description Logics, Part 1

by Mike Bergman - Wednesday, February 11, 2009

http://www.mkbergman.com/474/making-linked-data-reasonable-using-description-logics-part-1/

Can a 'Separation of Concerns' Lead to Better Slicing of the Pie?

It is clear that Fred Giasson and I have been spending considerable time on description logics of late. While we could perhaps claim that we like hard-to-read and -understand stuff, our reasons for this interest have been quite pragmatic: How to apply linked data principles to real-world commercial and organizational environments? Indeed, what should those principles even be?

Our first intuition, reaching back nearly two years now, was that linked data needed a context for bringing related datasets together. This belief led us to construct the UMBEL subject concept ontology, a basic reference roadmap for helping to point to information related (or "about") similar subjects. UMBEL as a set of subject concepts has proved useful as a reference roadmap; and the approach to construct UMBEL and its resulting vocabulary (heavily based in SKOS) has also proved helpful to construct specific domain-level ontologies.

By their nature, these ontologies have been ***conceptual*** and ***structural***. They define relationships, but are instance-poor. They focus on ways to describe various lenses -- world views -- into the domains for which we have been engaged.

But superstructures are meant to be built upon and fleshed out. For that, real instance data is required.

The Formative 'Named Entities'

Thus, we have more recently shifted from ***concepts*** and ***structure*** to focus on how to represent the ***actual things*** that populate that structure; that is, a domain's actual objects or instances. We appreciate that different audiences and proponents will use terminology such as *instance*, or *object*, or *entity*, or *individual* or even *foo* to describe such things, but for us (Peirceian logic aside) we simply wanted a way to describe the referents to a specific real-world thing [1].

We initially chose the term '*named entities*' to describe these actual objects. This naming arose from the work of Sekine and his 200 named entity types [2]. Typical named entities are specific (individual) people, organizations, events, artifacts ('Mona Lisa'), places, products, or whatever. For example, here is our first published definition describing '*named entities*':

Named entities are the real things or instances in the world that are themselves natural and notable class members of subject concepts. Named entities are the instances of the *subject concepts* in the standard definition of the term. Each named entity is mapped to a governing subject concept for ontology purposes.

Actually, '*named entities*', in even that sense, do not all have proper names with capitalization. Some accepted '*named entities*' are also written in lower case, with examples such as rocks ('gneiss') or common animals or plants ('daisy') or chemicals ('ozone') or minerals ('mica') or drugs ('aspirin') or foods ('sushi') or whatever.

But, hmmm. While 'daisy' might be an instance of the common flowers, is that the same as a specific daisy flower? especially when I can see literally thousands of daisy flowers at present in my back yard?

This epistemological question of thing *v* instance *v* individual can really mess you up! Furthermore, from the standpoint of describing these things on the Web, are we talking about the real thing, a symbol of some sort (Peirce again!) for that thing, or a multitude of similar descriptive terms (flower, bloom, daisy, florescence, bellis, chrysanthenum) for that thing?

Whether a thing is an instance or an individual or even a class depends on ***context***. A plant taxonomy could represent its terminal nodes as specific species or subspecies of daisy. But, in a flower show, the specific thing being referred to could actually be a unique individual, the *Pretty Miss Daisy* blue-ribbon winner.

Description logics with its TBox and ABox splits [3] actually helps considerably to unravel these potentially confounding distinctions. The ABox covers the description of instances with their asserted attributes or characteristics. Thus, we can have an ABox description of the daisy instance that refers to daisies in general or daisies as a species, or we can have an ABox description of an individual daisy with specific proper name in a flower show.

This instance idea is really a very clean one. As long as we focus on the idea of an instance and its attributes, we can put off for the moment (or defer to another layer, that is the reasoning TBox) what kind of instance this is.

After another segue, we'll return to this instance concept in a moment.

Happy Birthday!: DBpedia and the Beginning of Linked Data

DBpedia, the structured and linked data extraction of "facts" from Wikipedia, was first released about two years ago. Happy 2nd Birthday! I first wrote about DBpedia shortly thereafter claiming, I think somewhat accurately, the birth of the structured Web. We now know that phenomenon and the many additional datasets that nucleate around DBpedia as linked data.

When first explained, DBpedia used examples of so-called Wikipedia infoboxes for the cities of Leipzig or Innsbruck to describe the source of its structured data. (Subsequently Berlin has also been commonly referenced, all understandable given DBpedia's two principal founders of SÃ¶ren Auer at the UniversitÃ¤t Leipzig and Chris Bizer at Freie UniversitÃ¤t Berlin; of course, many others have joined and meaningfully supported the project since.) Infoboxes are Wikipedia templates that provide standardized, structured information across related articles of a similar type.

I have copied a similar infobox -- in this case for the same *city* type from Wikipedia for my home town of

Iowa City -- to show one of these structured data templates (shown to right). In using it I am, of course being a bit parochial, but it is also interesting to see the growth of structured data (attributes) that such templates now contain compared to what was available at the time of DBpedia's first release.

This infobox is a perfect example of an ABox. The instance it describes is the 'City of Iowa City'. Each of the items that follow show an attribute or data characteristic of some form with its associated value as a key-value pair. Sometimes those values refer to other instances, some of which are individuals, such as the county or name of the mayor.

In ABox terminology, these values are ***asserted*** for each attribute. Because this is Wikipedia, which has a reputation for accuracy and authority, we tend to believe and accept these assertions. But, we also know that sometimes these values are not correct, even for Wikipedia. We also know that instance records can come from many, many different sources, perhaps most not with the accuracy or authority of Wikipedia.

It is these types of instance records (for many other types of things than *city*, of course!) that are now being published as linked data. Today more than 50 general public datasets and perhaps another 50 from the sciences (especially biology) have been published. The total assertions across all datasets now exceeds millions, and the RDF statements that capture all of the relationships between these instances, attributes and concepts that describe them exceed one billion, as the recent Billion Triples challenge attests.

What is nice about this ABox structure is that they are relatively simple -- instances characterized by attributes -- with the "facts" so expressed understood to be assertions and not necessarily verified truth or accuracy. No matter what the source, there is no guarantee that all assertions will be complete and accurate. (Though, as has proved to be the case for DBpedia because of its Wikipedia heritage, some of the sources can be comfortably asserted to be authoritative.)

Assertions about many of the attributes are relatively straightforward such as, in the Iowa City instance, zip codes or time zones or population. (Still, the estimates used could also be out of date or the estimation methods could be argued.) However, other assertions, more based on interpretation or personal opinion, such as subject matter or political or religious affiliation or bias, can be quite controversial.

Another potential source of error is the linked data assertion that one instance is the `owl:sameAs` a different instance in a different dataset. Erroneous 'same as' assertions can arise quite simply and not require malice or stupidity. For example, for me, I actually live in Coralville, Iowa, not Iowa City. But, Coralville completely abuts Iowa City, shares a school district, and my wife works in Iowa City. I more often than not claim Iowa City as my location, though my actual mailing address is Coralville. How does one reasonably say that the identity of Michael Bergman of Coralville is the same as the Michael Bergman of Iowa City?

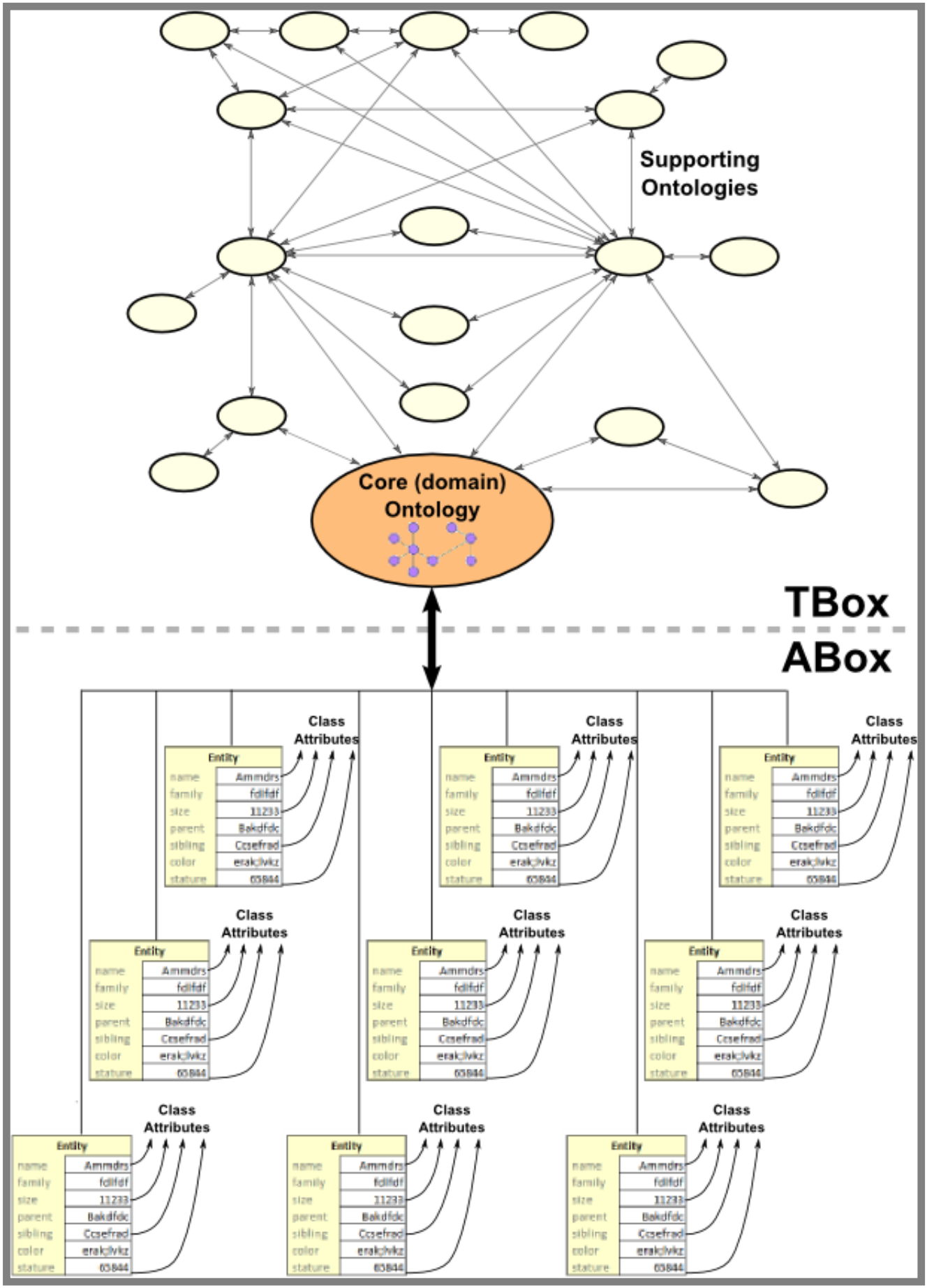
Cutting the ABox Slice Out of the Pie

So, what can the perspective of the ABox and description logics tell us about these issues?:

- First, instance records on their own can only contain assertions; instance records can not alone be a basis to decide the reasonableness of those assertions
- Second, if we stay focused on the idea of an instance record, we can wiggle off the hook about

whether we are talking about classes or groups or individuals. The instance is merely the thing at hand, with appropriate attributes or not based on the nature of the instance
- Third, the role -- or "burden" if you will -- of the instance record is merely to convey attribute assertions about a single instance. The ABox can be streamlined with comparatively little structure and comparatively little semantics
- Fourth, some attribute assertions are more straightforward and more easily tested, other attribute assertions are more problematic. That consideration should not limit the scope of any assertions that can be made in an instance record, just that certain attribute types may be harder to test or accept
- And, fifth, and most importantly, these considerations strongly suggest a clean break between data characterizations and structures to describe instances (the ABox) from how instances relate to one another or whether the attributes asserted for a given instance are reasonable or not (both being the work of the TBox).

This is the rationale from an earlier posting from me called Back to the Future with Description Logics that clearly separates the TBox and ABox functions:


Supporting Ontologies
Core (domain) Ontology
TBox
ABox
Class Attributes
Entity
name Ammdrs
family fdlfdf
size 11233
parent Bakdfdc
sibling Ccsefrad
color erakjlvkz
stature 65844

Now, it is true that the ABox and TBox distinctions are conceptual, and in practice not often actual, with no mandate or requirement based in description logics that they remain separate. However, for reasons of tractability and communication and computational performance at scale, there may be justification for keeping these constructs separate [4].

In the diagram, note that each ABox instance has the simple appearance of an instance record. Also note that the attributes that describe or characterize those instances should also be included and described with relationships modeled at the TBox level. The TBox is the proper place to describe all of the attribute relationships.

So, for Structured Dynamics, we have made a clean split in these roles and data structures in those client architectures over which we have design control. Ontologies populate the TBox level. Instance records assembled into instance dictionaries populate the ABox level, with various instance types governed by their own lightweight schema and vocabularies. This simple functional split leads to cleaner architectures and easier decisions about what belongs in which box or another for a given circumstance. It is also more performant, but more on that in a later part.

Summary of Benefits for Keeping the Pie Slices Separate

Of course, this is not how the linked data and semantic Web is currently architected or conceptualized. This smearing of roles and work responsibilities leads, we think, to many communication issues and slower uptake. As our own thinking gets clearer on these issues, we see there are some key benefits arising from keeping distinct the TBox (ontologies) and ABox (instances) pieces of the semWeb pie.

Benefit 1: Keeps World View Separate from Facts and Assertions

Wikipedia is a good case in point where conjoining facts with a world view does not work well. One part that does work well are the "facts" in the specific Wikipedia pages that describe things. They are the ABox structure of the Wikipedia knowledge base. Another useful aspect of Wikipedia, kind of at the interstices of the ABox and TBox, are its *see also* and *disambiguation* pages. These, too, have proved to be very useful for gathering synonyms for a specific instance or for disambiguating two similarly named instances.

But at the conceptual level of how the world is organized -- what are the relations between instances and how those instances are categorized -- Wikipedia has arguably been unsatisfactory. Why that might be is a discussion for another time.

One perhaps could make an inverse observation about the Cyc knowledge base where a quite coherent world view of concepts exists (and, actually, many world views through Cyc's very useful microtheories construct), but is often hard to discern and discover because of the admixing of instances, the coverage of which is also quite lumpy. Some domains have many instances, others are quite sparse.

Trying assiduously to keep bodies of facts and assertions (ABox) separate from how to interpret that world (TBox) brings distinct benefits. The facts base (ABox) is more easily tested for consistency. Different world views (TBox) can be more easily applied and compared against these fact bases. Testing and accepting different aspects of different sources is made easier if the ABox and TBox are not

conjoined.

Benefit 2: Keeps Terminology Simpler

When the different purposes and roles and resulting work that might be applied to ABox and TBox are conjoined, our ability to describe things gets murky. We sometimes call mere controlled vocabularies "ontologies", for example, which only acts to dilute the concept. We have facts and assertions and relations and hierarchies and stuff ranging from the minutiae to the abstract and sublime being lumped and described with the same terminology. Because we can not clarify and describe to ourselves roles and responsibilities for this stuff, no wonder we can't communicate well with the broader public.

I believe if the semantic Web community could stand back and try again to apply the rigor of description logics to its enterprise, now that we are gaining some real exposure and success with linked data, we could begin to clean up this emerging mess we are creating for ourselves.

Here are some starting suggestions. Let's call the combination of ABox and TBox a knowledge base, not an ontology. Let's reserve the term ontology for the terminological relationships and concepts at the TBox level. And let's focus on ABox instances as requiring only simple vocabularies to describe the assertions of attributes (what we might call schema consistent with RDFS and relational database schema). We thus could see a set of pieces similar to:

Knowledge Base = Ontology + [Disambiguation] + [Identity Relatedness] + Instance Schema

Note I suggest a couple of interesting work items at the interface between the TBox and ABox: disambiguating instances and determining the identity relatedness (for example, 'same as') between instances. This is work that should be kept apart from the ABox, but may or may not be best handled in the TBox (and, in any case, is generally separate work from the conceptual structure of the TBox).

This separation of concerns, or something akin to it, would result in a much cleaner -- and, therefore, simpler -- terminology for communicating with the interested public.

Benefit 3: Enables Simpler Instance Schema

Prima facie, an instance schema that merely needs to capture attribute assertions for an instance will be much simpler than current practice. In turn, that should lead to more patterned schema with easier and quicker extension to new domains and vocabularies. And, that, in turn, will aid ABox consistency checking.

Benefit 4: Easier Conversion of non-RDF Structured Data

Without the need for ontology mapping at time of conversion, existing RDFizers could be more readily applied to convert other structured data forms to simple RDF schema.

Benefit 5: Enables Better Substitutability, Modularity

Splitting the pie as suggested is merely the application of separation of concerns, which I believe all would largely acknowledge as leading to better substitutability and modularity. Besides swapping alternative world views to test their implications against common ABox datasets (the Benefit #1 case), we would also likely see quicker improvements in methods and algorithms for ABox consistency checking.

Benefit 6: Enables Better Dataset Descriptions

There has been growing interest and effort behind finding methods and vocabularies for describing datasets. The Sindice effort has led to the creation of suggested sitemap standard for crawling purposes; UMBEL has suggested standard vocabularies for describing what datasets are "about", and voiD has been working to standardize how to characterize the nature of a dataset.

Insofar as the ABox and TBox are more cleanly separated, the decisions and tradeoffs for accomplishing these tasks should enable better dataset descriptions.

Benefit 7: Minimize Tensions Between OWL and RDF Proponents

The discourse between the OWL and RDF communities can often be strained and at cross purposes. Many data publishers in the OWL community are from the sciences, where reasoning and decidability is imperative [5]. Many in the linked data community are trying to get as much data exposed and published as possible. Kendall Clark recently blogged about these 'tribes' to which I also commented.

Like any world view, there is nothing inherently wrong with being more comfortable or wanting to live in one world as opposed to another. But ultimately, the assertions made by most linked data at the ABox level needs to be tested for reasonableness, and structure and an organizational view of the world (TBox) is not terribly helpful without instance data.

I wonder, in fact, whether it might be best for linked data publishers to eschew OWL altogether. Different RDF predicates could be adopted to claim sameAs-type assertions, for example, and ABox vocabularies and schema could be greatly simplified and patterned for easier development and templating. No matter how we cut it, all of this published data and its properties are only assertions until they can be tested for reasonableness, so why not accept that and make linked data generation faster and easier?

Everyone knows that data for data's sake -- linked or not -- has to be tested for reasonableness before it can be relied upon for real work. Simple RDF schema for structured search purposes can work alone just fine: simply look at the error rates with current search engines. But, beyond search and non-critical linked browsing, reasoning is necessary.

The reasoning community has known for some time that all of these linked data assertions will have to be tested anyway. So, why not accept roles? Make linked data easier for search and browsing and publishing, and keep silly entailment assertions out of the mix. Then, allocate the reasoning work to coherent ontologies that know their world view and how to test for it. Instance records and ABoxes are not decidable on their own, so why pretend otherwise?

---

[1] Including imaginary ones, such as fantasy or mythical things like Gandalf.

[2] In a *named entity*, the word *named* applies to entities that have a "rigid designators" as defined by Kripke for the referent. For instance, the automotive company created by Henry Ford in 1903 is referred to as Ford or Ford Motor Company. Rigid designators include proper names as well as certain natural kind of terms like biological species and substances. BBN categories proposed in 2002 consists of 29 types and 64 subtypes; Sekine's extended hierarchy also proposed in 2002 is made up of 200 subtypes. We use Sekine (http://nlp.cs.nyu.edu/ene/version6_1_0eng.html) as our guide. For example, Sekine's top 15 named entity classes are: Name_Other, Person, Organization, Location, Facility, Product, Event, Natural_Object, Title, Unit, Vocation, Disease, God, Id_Number and Color; the remaining types are subsumed under these. See further http://en.wikipedia.org/wiki/Named_entity_recognition. Generally, *named entities* are the instances of classes.

[3] As I earlier wrote in Thinking 'Inside the Box' with Description Logics (now updating 'instances' for 'individuals'):

"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."

[4] From the Wikipedia article on Description Logics (2/9/09):

So why was the distinction [between TBox/ABox] introduced? The primary reason is that the separation can be useful when describing and formulating decision-procedures for various DLs. For example, a reasoner might process the TBox and ABox separately, in part because certain key inference problems are tied to one but not the other one ('classification' is related to the TBox, 'instance checking' to the ABox). Another example is that the complexity of the TBox can greatly affect the performance of a given decision-procedure for a certain DL, independently of the ABox. Thus, it is useful to have a way to talk about that specific part of the knowledge base.

The secondary reason is that the distinction can make sense from the knowledge base modeler's perspective. It is plausible to distinguish between our conception of terms/concepts in the world (class axioms in the TBox) and particular manifestations of those terms/concepts (instance assertions in the ABox.)

[5] See, most recently, Alan Ruttenberg's comment on the W3C's semantic-web mailing list at http://lists.w3.org/Archives/Public/semantic-web/2009Feb/0081.html.
