Important Issues, Considerations, and Opportunities for Accounting Professionals in Creating the Digital Financial Report *(Knowledge Engineering 101 for Business Professionals)*

by

Charles Hoffman, CPA (charleshoffman@olywa.net) and
Timothy Randle (timothy@clientserversllc.com)

March 1, 2015 (DRAFT)

ABSTRACT:

The specific pieces necessary to enable the digital general purpose financial report will not spring into existence by accident. The pieces must be created with high intension, specific conscious effort, well-thought-out direction, making specific conscious choices, and skillful execution by professionals with resolve to achieve the goal.

To achieve this goal will take the effort of accounting professionals, financial analysts, software engineers, knowledge engineers, and others who come from different worlds but who must effectively communicate to decide on what the right goal is and collaborate to achieve that goal.

Prudence dictates that using financial information from a digital general purpose financial report not be a guessing game. It is only through conscious effort that the specific control mechanisms can be put in place to realize this intent. The system that works safely, reliably, predictably, repeatedly, effectively, and efficiently is the desired goal.

XBRL-based digital financial reports created by public companies and filed with the US Securities and Exchange Commission offer empirical evidence as to what specific machine-readable information is necessary and opportunities to test digital financial reports to determine if they are working as needed.

If all aspects of digital financial reporting can be brought into equilibrium; similar to how CAD/CAM changed the blueprint and product lifecycle; financial reporting can be brought into the 21st century and can be a valuable tool used by millions and millions globally who participate in the financial reporting supply chain.

This document helps accounting professionals understand their role in creating a machine-readable, digital, general purpose financial statement.

The general purpose financial report is getting a face lift, being updated for the 21st century. It is hard to say exactly when this process began. In the early 1900's financial disclosures became more standardized. In the 1970's efforts began to create a set of international financial reporting standards. In the last part of the 20th century the XBRL technical specification was created, establishing a global standard technical syntax usable for business and financial reporting. In the early 21st century the US Securities and Exchange Commission funded the creation of the US GAAP XBRL Taxonomy and mandated that public companies report to the SEC using the XBRL technical syntax.

But public companies who report to the SEC amount to only about 10,000 entities who are regulated by the SEC. There are still approximately:

- 90,000 state and local governmental entities in the US
- 360,000 not-for-profit entities in the US
- 28,000,000 private entities in the US
- Similar numbers of state and local governmental entities, not-for-profits, and private entities in other parts of the world

All these entities could benefit from the digital financial report.

Digital financial reporting has the opportunity to do for the financial report and the financial reporting supply chain what CAD/CAM did for not only the blueprint, but for the entire product design and manufacturing life cycle[1]. The following is a brief explanation of CAD, commuter aided design[2]:

> CAD software is used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing. CAD output is often in the form of electronic files for print, machining, or other manufacturing operations.

CAD/CAM software not only enabled electronic blueprints and other product design and engineering information to be exchanged; it enabled a revolutionary change in work practices. The role of the draftsman, the designer, and the engineer merged. It would be a challenge to even build many of the complex products of today without CAD/CAM software. Electronic blueprints enabled the exchange of information directly from the designer to the numerically controlled machine which builds the products.

---

[1] *A Brief Overview of the History of CAD*, 2008 David E. Weisberg, http://www.cadhistory.net/02%20Brief%20Overview.pdf

[2] Computer-aided Design, Wikipedia, http://en.wikipedia.org/wiki/Computer-aided_design

But to make digital financial reports usable, digital financial reports need to work. Defining "work" can be subjective. What "work" means must be decided by the participants of the financial reporting supply chain, the ultimate creators and users of such financial reports. Other aspects of defining work are less subjective or even completely objective and even mechanical[3].

There are many choices involved in this process. The first choice is whether creating the machine-readable digital financial report is a desirable goal in the first place. Next, issues related to how a digital financial report might work need to be resolved.

Initial use of digital financial reports, mandated public company reporting to the SEC, has not gone perfectly and it can be hard to quantify the relative success or failure of that implementation of digital financial reports. But by all accounts, the following seems to be true:

- Software used to create XBRL-based digital financial reports is hard for public companies to use, there is no way private companies would tolerate that level of usability
- Data quality is not good enough to allow safe and reliable use of XBRL-based financial information which is reported.

This document is not about evangelizing XBRL-based digital financial reporting. Rather, this document explains general aspects of knowledge engineering that should be considered by accounting professionals in order to make digital financial reporting work effectively however one might define "work effectively". If creating and using a digital general purpose financial report is not simple, cost effective, and effective in creating and exchanging financial information of a financial report, then digital financial reporting can never be adopted by the masses.

This document summarizes important general issues, outlines important considerations, and points out the opportunities which could be provided by digital financial reporting if it works as deemed appropriate by the financial reporting supply chain. The essence can be summarized in one concise reality:

> The only way a meaningful exchange of information can occur is the *prior existence* of agreed upon technical syntax rules, business domain semantics rules, and process/workflow rules.

[3] See *Understanding the Basic Mechanics of a Digital Financial Report*, section Understanding the notion of slot or opening, page 9, http://www.xbrlsite.com/2015/Library/UnderstandingTheMechanicsOfDigitalFinancialReport.pdf

Goal

As Stephen R. Covey pointed out in is seminal work *Seven Habits of Highly Effective People*[4], "Begin with the End in Mind." We begin with the end.

Prudence dictates that using financial information from a digital financial report not be a guessing game. It is only through conscious effort that the specific control mechanisms can be put in place to realize this intent.

The goal is a system that works safely, reliably, predictably, repeatedly, effectively, and efficiently.

Information technology professionals creating software must be able to create software which yields the same result when it would seem obvious to a business professional using software that the result, such as a query of basic information from a financial report, should be exactly the same even if different software applications are used.

Conscious and skillful execution using this approach can create digital financial reporting which is simple and elegant; and yet a sophisticated and powerful tool.

Power of agreement

It is only through conscious collaboration, cooperation and coordination by the participants of the financial reporting supply chain that XBRL-based digital financial reporting will work safely, reliably, predictably, repeatedly, effectively, and efficiently. That is the goal. This goal will not be achieved by accident.

Consider the definitions of arbitrary and standard:

- **Arbitrary**: based on random choice or personal whim, rather than any reason or system; depending on individual discretion (as of a judge) and not fixed by law
- **Standard**: used or accepted as normal; something established by authority, custom, or general consent as a model or example

[4] *Seven Habits of Highly Effective People, Habit 2*, http://www.amazon.com/The-Habits-Highly-Effective-People/dp/1455892823

Is the purpose for each individual participant in the financial reporting supply chain to dig their heels into the ground and insist that their arbitrary reality is the only reality? Or is the purpose to consciously create a coordinated, shared, commonly accepted, standard, useful view of reality *to achieve a specific purpose*: so that reality does appear to be objective and stable enough yet nuanced enough to be useful so that information can be used safely, reliably, predictably, repeatedly by both human and automated machine-based processes. The desired system state is one of balance or equilibrium; of consistency.

Agreement is what creates the possibility of enabling machines to perform certain tasks. Business professionals are practical people. If business professionals wanted to have endless theoretical debates they would have become theologians, academics or philosophers. The goal is not to persist the debate; the goal is to agree in order to achieve a specific purpose.

Basic Mechanics of a digital financial report

The basic mechanics of a digital financial report are consistent[5]. XBRL-based public company financial reports filed with the U.S. Securities and Exchange Commission are empirical evidence of this consistency. This consistency is caused by clarity as to these fundamental mechanics articulated by the technical specifications which describe how XBRL works. These mechanics are not open to interpretation.

And yet while 99.9%[6] of these relations are consistent, professional accountants, software vendors, and others do interpret these fundamental mechanics slightly differently.

At the highest level the financial information which is reported is likewise consistent. Overall consistency of basic relations such as "Assets = Liabilities and Equity" (the accounting equation) is about 98%[7]. Consistency of that specific relation, the accounting equation, is 99.7%. By consistency we mean that every financial report universally follows a specific rule. This does not mean that every report follows exactly the same rules. For example, not every economic entity

---

[5] *Understanding the Basic Mechanics of a Digital Financial Report*, http://www.xbrlsite.com/2015/Library/UnderstandingTheMechanicsOfDigitalFinancialReport.pdf

[6] See *Arriving at Digital Financial Reporting All Stars: Summary Information*, page 4, http://www.xbrlsite.com/2014/Library/AnalysisSummary_ArrivingAtDigitalFinancialReportingAllStars.pdf

[7] See *Summary Information about Conformance with Fundamental Accounting Concept Relations*, http://www.xbrlsite.com/2014/Library/SummaryInformationAboutConformanceWithFundamentalAccountingConceptRelations.pdf

provides a classified balance sheet; some provide an unclassified balance sheet. But every entity provides either a classified or unclassified balance sheet. Classified and unclassified balance sheets have different rules. Liquidity basis statements of financial position are simply another class of statement.

And so it is the mechanical aspects of a financial report provides the frame of the report and are completely objective and not requiring judgment. What requires judgment is deciding *what* should go into the financial report; what gets disclosed.

Differentiating objective mechanical aspects from subjective aspects which require professional judgment

Digital financial reports contain thousands and sometimes many thousands of individual pieces or structures. These structures, commonly formatted in machine-readable form using XBRL, are used to represent the information contained in the digital financial report. There are two distinct aspects of these pieces or structures that are important to recognize and be conscious of:

- **objective aspects** which are mechanical and do not require judgment and therefore can be managed using automated machine-based processes.
- **subjective aspects** which require the professional judgment of a skilled accountant, therefore they must be managed by humans.

These objective mechanical aspects are distinct from the subjective aspects which require professional judgment. The mechanical aspects relate to the things and relations between the things in a digital financial report. These mechanical aspects are governed by logic, common sense, and the rules of math. These mechanical aspects are what make up the structure or substrate of a financial report. Everything else fits into this frame or skeleton. This is much like the keystones of a building.

Representing the financial report problem domain in machine-readable form

A **problem domain**, such as the domain of financial reports, can be broken down into distinct, identifiable elements. This can happen on two different levels: first, on the level of individuals, when we break down this specific financial report into these specific elements *unique* to that specific financial report. And second, on the level of classes, when we distinguish classes of elements common to all financial reports and therefore *universal* to all financial reports. Another term for problem domain is area of concern.

Historically, information technology professionals and knowledge engineers have used different terminologies and schemes for describing these identifiable elements of a problem domain (concept maps, UML, entity relationship diagrams, and now what is commonly referred to as the "Semantic Web"). Different schemes often use different terms to refer to exactly the same thing or they use the same term to refer to different things. This adds to the confusion of how to best represent real world problem domains in machine-readable form and get the results a business professional expects and desires.

Therefore, we created one common set of terms based on global standard and current state-of-the-art technology. That standard is OWL 2 DL[8] and *SROIQ* Description Logic which have different technical syntaxes but equivalent semantics.

XBRL should remain consistent with this W3C global standard.

Machine-readable representations, Taxonomy/Ontology 101

Different terms are used to describe a machine-readable representation including taxonomy, ontology, and vocabulary[9]. Although it might seem scary, we will standardize on the term ontology and state that an XBRL taxonomy is, and ought to be, an ontology.

An ontology is a salient collection of the classes and subclasses of a problem domain or area of concern. An ontology should fit the needs of the some specific community, such as a supply chain. Ontologies describe or explain how the collection of things within the problem domain can be represented.

Ontologies have no concern with computational efficiency of a software application.

An ontology should be tractable rather than intractable.

[8] See OWL 2 Web Ontology Language Primer (Second Edition), http://www.w3.org/TR/owl2-primer/#OWL_2_DL_and_OWL_2_Full and the OWL 2 Overview, Semantics section, http://www.w3.org/TR/owl2-overview/#Semantics

[9] Interestingly, the W3C page http://www.w3.org/standards/semanticweb/ontology (notice ontology at the end of the URL) uses the term "Vocabularies".

We have distilled the key terminology down to its essence, focusing on terms important to business professionals, information technology professionals, and knowledge engineering professionals who need to communicate in order to articulate information about a problem domain in machine-readable terms. We use the Semantic Web language OWL to capture what we see as the most important elements in the domain of financial reports. OWL is a state-of-the art global standard approach to describing a problem domain.

The following are the key high-level definitions of terms:

- **Thing**: A thing is something that exists in the real world, in the problem domain, in the area of concern. A thing is just a class that all classes and individuals of the problem domain must belong to. All classes are subclasses of thing. Every individual must be of some class. Every class is a thing. Therefore since all classes are subclasses of thing; then all individuals are likewise ultimately a thing. "Nothing" is the opposite of thing.
- **Individual**: An individual is some specific item that exists in reality. For example, a specific person such as *Bill Gates III*, a specific report such as *Fiscal year 2014 financial statement*, a specific economic entity such as *Microsoft Corp*. An individual exists only once.
- **Class**: A class is a set of individuals that have one or more distinguishing features in common. For example *person* is the class consisting of all persons of which *Bill Gates III* is a member. Each problem domain can be captured in terms of a family of classes, together with a set of relations. The most important relation is the subclass relation (also called *is-a*) which organizes the classes in a taxonomic tree. Other key types of relations are *whole-part* and *has-part*.
- **Property**: A property is a trait, quality, feature, attribute of an individual, for example the property of *being male* of a person, of *being filed* of a report, and so on.
- **Relations between individuals:** one individual can be related to another individual, as when *Bill* is *brother-of Dave*, *Bill* is *owner-of the building at 1835 73rd Ave NE, Medina*, and so on.
- **Relations between classes**: when every member of a certain class stands in a certain relation to some member of another class, then the relation is universal and we can formulate this as a relation between classes. So for example because every brother *is identical to* some male person, we can

assert this as a *relation* between the classes *brother* and *male person* to the effect that *brother is-a male person* – in other words the class *brother* is included as a subclass in the class *male person*. If every *financial report* has some *statement* as part, then we can assert *financial report has-part statement*. Relations between classes are universal and apply to every member of that class.

The result of the above rules is a system which always has a single root class at the very top called 'Thing' and a single leaf class at the bottom called 'Nothing'. Thing is the universal class to which all other classes are subclasses must ultimately belong. All individuals are ultimately members of the universal class. Nothing is an empty class which has no members at all. And so, every such system has Thing at the top, Nothing at the bottom, and business problem domain classes in the middle.

This is a crucial distinction because that resulting organization allows for a conclusion to be reached as to the consistency of some human-readable or machine-readable representation of the problem domain with the description of the problem domain provided by the system. Basically, this system organization is finite rather than infinite.

Having a finite system organization is crucial because if a conclusion cannot be reached as to the consistency of some representation with the description then the system is infinite. Infinite systems are unsafe. Unsafe means that unexpected results, ambiguous results, complexity results which can lead to a machine entering an infinite loop from which it cannot escape could possibly occur.

The fact that the system can be completely described, to the extent of the expressive power of the language of the statements/axioms, by a given set of statements/axioms is provable using formal logic. As such, the finite system a useful tool: it is safe, predictable, reliable, results are repeatable, and no unexpected complexity-caused blowups will occur.

Representing Reality

The ontology uses lower-level terms which fit into the higher-level terms we just described.

The central function of an ontology is to represent reality of the problem domain comprehensively and accurately. The quality of an ontology is a function of the comprehensiveness and accuracy of the representation of things and relations between things which make up the problem domain. An ontology is a machine-readable “window” into reality.

There are two approaches to viewing “reality”.

One approach is to believe that reality (the world) exists objectively in-and-of itself; reality is independent of any one person. Therefore, reality is knowable; the world exists and its properties are there to be discovered. This view implies that reality is objective and knowable and therefore constraints can exist as to what can be said about reality. In other words, ontologies which provide representations of the world could get things wrong. Therefore, an ontology is right insofar as it accurately reflects the way the world is.

A second approach is to believe that there is no one reality, that every individual perceives the world and that individual perception is reality. This view implies that reality is subjective. This view does not imply that reality is not knowable because there are so many realities that it is impossible to agree on one reality. Rather, it implies that there are “reality camps” or groups of individuals with common beliefs about reality. Therefore, an ontology can represent one “reality camp”. Which implies that an ontology can be created for each camp. Therefore, the second approach becomes equivalent to the first approach.

The following terms help one understand the difference between an important nuance and an unimportant negligible difference.

- **Nuance**: a subtle difference in or shade of meaning, expression, or sound; a subtle distinction or variation
- **Subtle**: so delicate or precise as to be difficult to analyze or describe; hard to notice or see; not obvious
- **Negligible**: so small or unimportant as to be not worth considering; insignificant; so small or unimportant or of so little consequence as to warrant little or no attention

Business professionals can best differentiate important nuances from unimportant negligible differences. They do not do it perfectly and the only real way to make sure things are right is testing and experimentation at times.

Ontologies are about getting the salient aspects of a problem domain right. One needs to take a pragmatic view of the world because it is impossible to describe every single aspect of the world. Ontologies only need to represent the important things. An ontology is therefore more like a “wireframe” or a “substrate”.

Central to the idea of representing the things in reality is the notion of fidelity. Fidelity means to be a faithful representation or expression of reality relevant to the domain experts who explain the problem domain. Fidelity is the correspondence between or quality of the ontology's representation of the problem domain and the real world.

One final set of terms is important to make clear:

- **Policy**: a course or principle of action adopted or proposed by a government, party, business, or individual; definite course or method of action selected from among alternatives or options and in light of given conditions to guide and determine present and future decisions or choices
- **Requirement**: a thing that is needed or wanted; something that is needed or that must be done
- **Choice**: an act of selecting or making a decision when faced with two or more possibilities or options; the act of choosing; the act of picking or deciding between two or more possibilities or options
- **Option**: a thing that is or may be chosen; the opportunity or ability to choose something or to choose between two or more things

The reason these terms are important is because if options exist and therefore a choice exists, but then a policy is established that no longer allows certain options; then an option can be turned into a requirement.

Difference between "simple" and "simplistic"

Anyone can create something that is sophisticated and complex. It is much harder to create something that is sophisticated and simple. Simple is not the same thing as simplistic. "Simple" is not about doing simple things. Simple is the ultimate sophistication. Simple is elegant.

Simplicity is "dumbing down" a problem to make the problem easier to solve. That is not what simple is about. Simple is about beating down complexity in order to make something simple and elegant; to make sophisticated things simple to use rather than complex to use.

Challenges of representing a problem domain

The creation of an ontology is an engineering process, the specialty of knowledge engineers.

An ontology is created about some problem domain, the specialty of domain experts. Financial reporting is a problem domain and professional accountants are experts in that problem domain.

The creation of the machines, the software applications, which leverage the machine-readable ontology is likewise an engineering process, the specialty of software engineers.

It is important to define the term engineering. Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of something." Building a bridge and engineering a bridge are different things.

Software engineering and knowledge engineering lives in their own little worlds, silos[10]. Professional accountants live in a completely different world.

And so to summarize this situation succinctly: software engineers generally don't understand knowledge engineering; knowledge engineers generally don't understand software engineering; neither software engineers nor knowledge engineers generally understand financial reporting; and professional accountants generally have no idea what knowledge engineering is and are only a little more adept at communicating with software engineers.

Yet, enabled functionality, when properly implemented in software, could provide professional accountants with an ability to automate certain specific mundane tasks.

Add to that differences in the interests of participants in the financial reporting supply chain. Professional accountants don't all have the same fundamental interests. Some professional accountants create financial reports. Other professional accountants, financial analysts, analyze the information reported within the financial reports. Other professional accountants work for the FASB and have to create the financial reporting standards necessary for economic entities to report and satisfy the information needs of financial analysts and other users of such information.

Each of these subgroups of professional accountants has a different take on reality because they have different fundamental interests.

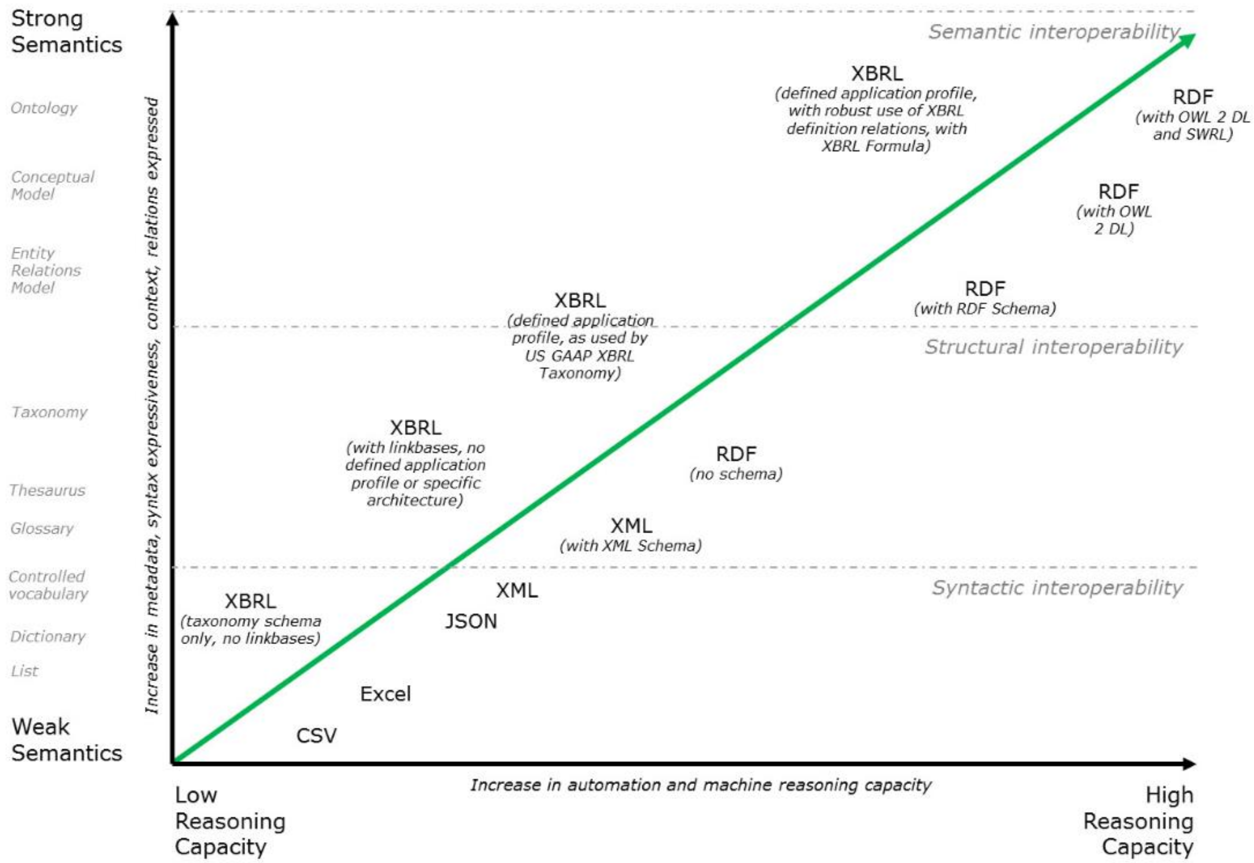
[10] *Applications of Ontologies in Software Engineering*, https://km.aifb.kit.edu/ws/swese2006/final/happel_full.pdf

Ontologies must be engineered (systematic, disciplined, qualified, etc.) by professionals who communicate effectively. All too often, a software engineer listens to a business professional, takes notes, and then implements what was written down in the notes. That is not engineering. Often, one needs to have the professional skills to "read between the lines" of what a business professional is saying in order to distill the true meaning from what was said.

Overcoming Limitations of Knowledge Representation Languages

No knowledge representation language is 100% complete. Each has limitations. One must be conscious of such limitations when creating a representation of some problem domain in machine readable form. The graphic below compares knowledge representation language expressive power with the achievable relative level of automation and/or reasoning capacity which can be achieved with that knowledge representation language.

Comparison of Knowledge Representation Language Expressiveness and Relative Automation/Reasoning Capacity



spired by similar comparisons from *An Intrepid Guide to Ontologies* http://www.mkbergman.com/date/2007/05/16/ and *Semantics Overview* http://prezi.com/prwsxj8po3ln/semantics-overview/

Neither XBRL nor OWL 2 DL + SAFE SWRL has 100% of what is necessary to represent 100% of what is necessary for digital financial reporting. Which is the best alternative is unknown at this point in time. The specific gap between the two in terms of expressive power is unknown at this time.

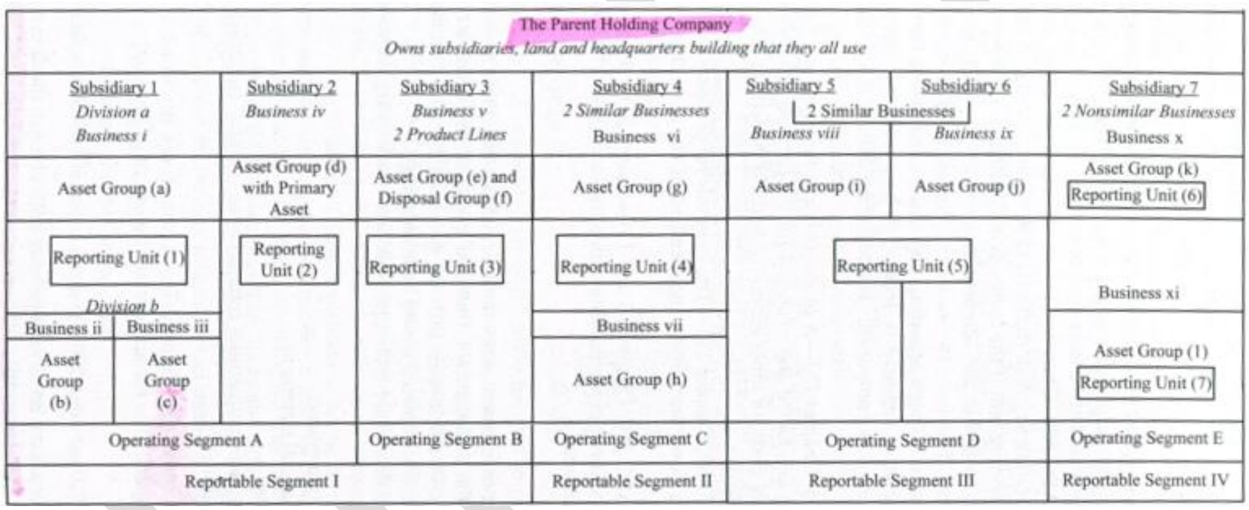
Pitfalls of knowledge engineering

There are many different ways to stumble when trying to represent the knowledge of a problem domain. The following is a summary of many common pitfalls which should be recognized and then avoided.

One rigid reality. Many of the things in a business problem domain are the invention of humans: the foot or meter, currency such as the US Dollar or the Euro, laws, regulations, accounting rules, concept of a legal entity. As such, to a large extent these things that are the creation of humans are malleable. At times there cannot be one single "correct" ontology for things in a problem domain because of inconsistencies in these human inventions. And so it can be the case that there is no single objectively correct answer, but possibly some set of pragmatically-based set of correct answers of some set of groups of users with clearly defined goals but with different sets of interests or self-interest of the specific set or group.

Fundamentally, excessive commitment to reality can lead to and inappropriate level of flexibility or inflexibility.

To make this point clear we use the following example pointed out in the *Wiley GAAP 2011, Interpretations and Applications of Generally Accepted Accounting Principles*, Bragg, page 46:



The segments into which a reporting entity can be broken down are defined inconsistently in the financial reporting literature. From FASB Accounting Standards Codifications, ASC 280 relates to the classification of assets and sometimes liabilities uses the terms operating segments and reportable segments of the business. ASC 350 which relates to impairment uses the term reporting unit. ASC 860 which relates to special-purpose entities and the master glossary uses the term business. ASC 360 which relates to long-lived assets uses the term asset groups and disposal groups.

As such, the following terminology is proposed by the Wiley GAAP Guide:

- Consolidated entity
- Parent holding company

- Operating segment (ASC 280)
- Reportable segment (ASC 280)
- Reporting unit (ASC 350)
- Business (ASC 805)
- Asset group (ASC 360)
- Disposal group (ASC 360)

There are two approaches to dealing with this issue: (a) get the FASB to fix the problem or (b) do something to address the symptoms of the problem because the FASB won't or can't address this issue.

Again, note that this is one specific example provided to show that reality is sometimes malleable. At other times reality is less malleable. This specific example is representative of a more general situation.

Overly complicated representation. On the one hand, one must be careful of the illusion of clarity and apparent rigor where, in fact, there is little or no rigor or clarity. These illusions mask problems definitions of things which can be exceedingly difficult and even problematic to correctly characterize or how things interact with one another. Some problem domain things can be untenable regardless if one attempts to articulate the things in machine-readable form. Not recognizing such issues provides a false sense of meaningful information exchange.

Overly complicated representations are spots where the illusion of clarity can hide. Making things obscure by adding unnecessary and perhaps inaccurate details. This also adds to complexity which is simply not necessary.

Blind trust of domain experts. Knowledge engineering calls for careful attention being paid to domain experts characterization of a domain by skilled knowledge engineers. But giving blind trust to domain experts is not appropriate. Knowledge engineers must have a critical side, analyzing and challenging representations for consistency and adequacy. Domain experts are not always right. Blind trust can lead to inappropriate tolerances and otherwise poorly constructed knowledge representations and ultimately an unworkable machine-readable representation.

One of the best ways to overcome this pitfall is to use rigorous testing in order to check understanding.

Misuse of highly-expressive languages. Using a highly-expressive language is no guarantee against sloppiness or process deficiencies. Highly-expressive languages offer the power and ability to articulate rich and precise rules for important classes and relations between classes. A weakly-expressive language encourages sloppiness and commonly leads to inaccuracies due to the deficiencies in ability of the weakly-expressive languages to articulate important classes and relations between classes. Where only weak-expressivity is available rich expressiveness is not even available to the knowledge engineer; the result can be a superficial representation which is not useable by the problem domain.

Recognize that pitfalls are avoidable

Pitfalls are avoidable. Limitations are many times unavoidable and must be worked around. While the real world is malleable and there are always options for representing classes and relations between classes in various ways; this does not mean that everything can be created in any way one pleases. Using one approach in one specific area can mean that options are constrained for some other area of the representation. Dysfunctional, irrational, nonsensical, illogical, inconsistencies, and other issues which cause problems must be discovered and dealt with.

There is a difference between conscious inconsistencies and unconscious inconsistencies. Conscious inconsistencies are generally choices which are made because things are truly different, perhaps only subtle differences or nuances. Unconscious inconstancies are generally due to sloppiness and lack of attention to detail and cannot be explained which pointed out and questioned.

Rigorous testing maximizes communication and quality

The best way of assuring that a machine-readable representation is not dysfunctional, irrational, nonsensical, illogical, inconsistent or has some other issue is comprehensive, thorough, rigorous testing. Another is examining empirical evidence. Testing is s robust and pragmatic approach to checking understanding and determining if communication has taken place between domain experts, knowledge engineers, and software engineers who ultimately must implement software.

A framework which cannot be measured for simplicity is a recipe for unnecessary complexity. Conscientious knowledge engineers are compelled to express a problem domain's classes and relations as richly as possible. With a highly-expressive language at a knowledge engineer's disposal it is possible to think through different representational options at a level of detail that is impossible with a weaker-expressive language. Stronger frameworks push one more than one using a weaker framework. Testing pushes one more than not using testing toward greater accuracy and comprehensiveness. As is said, "Ignorance is bliss." Limitations of expressivity of the representation language used should be exposed so that the limitations become conscious.

Global Standard Knowledge Engineering Framework

Empowered by this goal and with the intension of achieving this goal; the intelligent and wise direction of those who brought OWL 2 DL and *SROIQ* Description Logic (fragment of first-order logic which is decidable) together should be emulated.

At a minimum, there will be software vendors and others who desire to convert from OWL 2 DL + SAFE SWRL to XBRL, and from XBRL to OWL 2 DL + SAFE SWRL. No matter what the representation language, the meaning expressed should be equivalent as the "reality" being represented by the domain is the same. It is only the representation language which changes. While different representation languages have different limitations in terms of what can be expressed, what can be expressed should mean the same in each representation language.

Relations between things are business rules which should be managed by business professionals

As we pointed out earlier, an ontology provides a machine-readable representation of the important things and relations between the things of some problem domain. People refer to these relations in many different ways.

Some people use the terms "T Box[11]" and "A Box[12]". T Box statements describe the things, the terminology component or the controlled vocabulary of a problem domain. A Box statements describe the relations between the things, the assertions component of the problem domain.

[11] T Box, http://en.wikipedia.org/wiki/Tbox

[12] A Box, http://en.wikipedia.org/wiki/Abox

Another term used to describe relations is business rule. The Business Rules Manifesto[13] does a good job of describing what a business rule is. Article 9; *Of, By, and For Business People, Not IT People*; points out the need for these business rules to be managed by business professionals:

- 9.1. Rules should arise from knowledgeable business people.
- 9.2. Business people should have tools available to help them formulate, validate, and manage rules.
- 9.3. Business people should have tools available to help them verify business rules against each other for consistency.

Business professionals are the ones who understand the problem domain. As such, business professionals are the ones who understand the business rules or relations between the things in their problem domain.

One global standard digital financial report or multiple global standards?

Unless someone consciously and explicitly creates one global standard digital financial report specification then there is a risk that multiple digital financial report specifications will exist. While consciously and explicitly creating one global standard digital financial report specification *is no guarantee* that only one such specification will exist; if no one specification is created it is at least highly likely that multiple specifications will come into existence and those digital financial report specifications may or may not be interoperable. Further, if one global standard specification is not created it opens up the possibility of multiple proprietary standards which are even less likely to be interoperable.

While it is not the end of the world if there are two or perhaps even a few more global standards for digital financial reporting it is the business professional who will ultimately pay the price for unnecessary standards. And this is *not* to say that if two global standards exist for conscious reasons and with explicit differences in functionality which someone can point to and explain. There is nothing wrong with two global standards if business professionals require two global standards.

What would be a travesty is if there were 10 global standards when 1 global standard would have done and business professionals pay for the inattention which caused that problem to occur with higher priced software.

---

[13] *Business Rules Manifesto*, http://www.businessrulesgroup.org/brmanifesto.htm

Imagine a bank trying to implement digital financial reporting in order to reduce the costs of collecting and managing financial information in support of a commercial loan. Say that digital financial reporting was adopted and that for one reason or another 10 different standards for creating a digital financial report existed. Say the bank had 10,000 customers who had loans and who now must submit digital financial reports to the bank. Say the 10 different standards where used equally, 1,000 customers used each of the 10 different formats. How would that work out for the bank which needed to deal with 10 different formats?

Reading list

The following books are extremely helpful in trying to understand digital financial reporting. We strongly recommend that for anyone who wants to understand digital financial reporting well or who want to build rock-solid products/solutions to read the following books:

***Data and Reality*, by William Kent**: (162 pages) While the first and last chapters of this book are the best, the entire book is very useful. The primary message of the Data and Reality book is in the last chapter, Chapter 9: Philosophy. The rest of the book is excellent for anyone creating a taxonomy/ontology and it is good to understand, but what you don't want to do is get discouraged by the detail and then miss the primary point of the book. The goal is not to have endless theoretical/philosophical debates about how things could be. The goal is to create something that works and is useful. A shared view of reality. That enable us to create a common enough shared reality to achieve some working purpose.

***Everything is Miscellaneous*, by David Wenberger**: (277 pages) This entire book is useful. This is very easy to read book that has two primary messages: (1) Every classification system has problems. The best thing to do is create a flexible enough classification system to let people classify things how they might want to classify them, usually in ways unanticipated by the creators of the classification system. (2) The big thing is that this book explains the power of metadata. First order of order, second order of order, and third order of order.

***Models. Behaving. Badly.*, by Emanual Derman**: (231 pages) The first 100 pages of this book is the most useful. If you read the *Financial Report Semantics and Dynamics Theory*, you got most of what you need to understand from this book. But the book is still worth reading. It explains extremely well how it is generally one person who puts in a ton of work, figures something out, then expresses extremely complex stuff in terms of a very simple model and then thousands or millions of people can understand that otherwise complex phenomenon.

***Semantic Web for the Working Ontologist*, by Dean Allenmang and Jim Hendler**: (354 pages) The first to chapters are the most useful. This is an extremely technical book, but the first chapter (only 11 pages) explains the big picture of "smart applications". It also explains the difference between the power of a query language like SQL (relational database) and a graph pattern matching language (like XQuery). Querying can be an order of magnitude more powerful if the information is organized correctly.