Comprehensive Introduction to Knowledge Engineering for Professional Accountants

By Charles Hoffman, CPA (Charles.Hoffman@me.com)

August 29, 2016 (DRAFT)

Professional accountants and auditors need to understand how computers work to properly understand the true capabilities of computers. They also need to understand how to control computers to get them to perform the work they desire to be performed. This understanding is necessary to interact with information technology professionals and knowledge engineering professionals that build the tools professional accountants and auditors need. This understanding will become increasingly important as software is created to perform more and more tasks to assist professional accountants and auditors in their day-to-day work tasks. Current advancements in areas such as artificial intelligence will contribute to an increased pace of change. This document summarizes, organizes, and synthesizes information helpful to professional accountants and auditors that want to gain this understanding. It also provides liberal use of footnotes that point to additional details should a reader desire more information.

1. Digital Age and Machine-readable Information

Engineering is the application of a systematic[1], disciplined[2], quantifiable[3], methodical[4], rigorous[5] approach to the development, operation, and maintenance of something. A kluge is a term from the engineering and computer science world that refers to something that is convoluted and messy but gets the job done.

Bridges are engineered when they are constructed. Engineering entails the skillful construction or creation of something leveraging known laws of how things interact with one another. A civil engineer does not simply throw concrete and steel together to construct a bridge. The bridge is engineered to balance cost, strength, likelihood that the bridge remains standing during high winds or an earthquake, etc. Likewise when we work with information using a computer, how we achieve our goals and objectives is an engineering process, not simply throwing a few things together. How computers work is governed by laws that are well understood. What a computer can do reliably and safely are well understood by skillful computer science and information technology professionals.

Professional accountants and auditors need is to understand how computers work and how to control the workings of computers to accurately understand what computers are capable of doing and what they are not capable of doing. Professional accountants need to understand how to get computers to do what they want them to do as these tools are increasingly important in today's digital age. Accountants have been called knowledge workers. The fact is everyone is a knowledge worker.

Knowledge[6] is justified true belief. Knowledge is the fact or condition of being aware of something; the range of one's information or understanding. Knowledge is justified with observable evidence that others can use to corroborate a belief, to support or justify the belief. Knowledge is provable.

In the past most knowledge was in human-readable form. For example, the knowledge of the financial condition and financial position of an economic entity was articulated in the form of a paper-based financial statement readable only by humans. In the digital age, that same knowledge, through the use of structured data[7] formats such as XBRL, are machine readable. Structured data is being used more and more.

Does information, such as the financial position and financial condition of an economic entity, change based on the format used to represent that information? Clearly not. And so, to live in our digital world, professional accountants and auditors need to work effectively with machine-based information and knowledge represented as structured data.

Digital has positive features, but just like anything else it can also have potentially negative or less favorable features. To harness the power of machines appropriately, professional accountants need to understand how these machines work and how to control them to get them to do what they want them to do. This document helps professional accountants gain that understanding.

---

[1] Dictionary.com, *Systematic*, http://www.dictionary.com/browse/systematic

[2] Dictionary.com, *Disciplined*, http://www.dictionary.com/browse/disciplined

[3] Dictionary.com, *Quantifiable*, http://www.dictionary.com/browse/quantifiable

[4] Dictionary.com, *Methodical*, http://www.dictionary.com/browse/methodical

[5] Dictionary.com, *Rigorous*, http://www.dictionary.com/browse/rigorous

[6] Merriam-Webster, *Knowledge*, http://www.merriam-webster.com/dictionary/knowledge

[7] *CFA Institute Calls for Broader and Deeper Use of Structured Data*, http://xbrl.squarespace.com/journal/2016/8/16/cfa-institute-calls-for-broader-and-deeper-use-of-structured.html

1.1. Digital age is causing rapid change

Whether you call it the information age, the digital age, or the era of cognitive computing[8], change is occurring rapidly. Machines beat human chess masters. Machines play Jeopardy and win against human champions. The navigation systems in our cars perform amazing tasks that serve us well. Siri and other intelligent agents are at our beckon call.

Some tend to have an optimistic view of the capabilities of computers, overstating their potential. Others tend to have a pessimistic view of potential capabilities, understating possible usefulness. Understanding how technology works can help one be more conscious of the true capability of computers to help get work done.

With rapid change comes hype, snake oil salesmen looking for easy targets, expensive mistakes if the wrong choices are made, and missed opportunities if action is not taken. It is not necessary, or even desirable, to be on the bleeding edge of technology. But you don't want to completely miss the boat either.

The Economist predicts[9] that 94% of accounting jobs will be replaced by computers over the next 20 years. That percentage is 98% for accounting clerks, audit clerks, and bookkeepers. While predictions may, perhaps, be overstated; change to some degree is not only inevitable, change is imminent.

Computers are machines. The first mechanical computers, called tabulating machines[10], were created in the 1900s. Since then the effectiveness and efficiency of those machines have improved by orders of magnitude. But fundamentally the machines we use today are no different than mechanical tabulating machines. The key word here is machine.

Professional accountants need to understand these machines which are tools of knowledge workers.

Digital financial reporting is the future of financial reporting. Please see the document *Conceptual Overview of XBRL-based, Structured, Digital Financial Reporting*[11] for more information.

1.2. Knowledge workers rearranging abstract symbols

Computers sometimes seem to perform magic. But computers are really simply machines that follow instructions. Skilled craftsmen who wield their tools effectively, providing the correct machine-readable instructions, create what seems to be magic.

Professional accountants and auditors need is to understand how computers work and how to control the workings of computers and to accurately understand what computers are capable of doing, what computers are not capable of doing, and how to ultimately get computers to successfully perform work which serves the needs and desires of accountancy.

In his book *Saving Capitalism*[12], Robert Reich describes three categories that all modern work/jobs fit into:

---

[8] *What is cognitive computing? IBM Watson as an example*, http://www.duperrin.com/english/2014/05/27/whats-cognitive-computing-ibm-watson-example/

[9] The Economist, Jan 18, 2014, *The future of jobs: The onrushing wave*, http://www.economist.com/news/briefing/21594264-previous-technological-innovation-has-always-delivered-more-long-run-employment-not-less

[10] Wikipedia, *Tabulating Machine*, https://en.wikipedia.org/wiki/Tabulating_machine

[11] *Conceptual Overview of XBRL-based, Structured, Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2016/Library/ConceptualOverviewOfDigitalFinancialReporting.pdf

- **Routine production services** which entails repetitive tasks
- **In-person services** where you physically have to be there because human touch was essential to the tasks
- **Symbolic-analytic services** which include problem solving, problem identification, and strategic thinking that go into the manipulation of symbols (data, words, oral and visual representations).

In describing the third category, symbolic-analytic services, Mr. Reich elaborates:

> "In essence this work is to rearrange abstract symbols using a variety of analytic and creative tools - mathematical algorithms, legal arguments, financial gimmicks, scientific principles, powerful words and phrases, visual patterns, psychological insights, and other techniques for solving conceptual puzzles. Such manipulations improve efficiency-accomplishing tasks more accurately and quickly-or they better entertain, amuse, inform, or fascinate the human mind."

Why this is interesting to me is the third category of work/jobs: symbolic-analytic services. Financial reporting, or at least many tasks related to financial reporting, fall into the symbolic-analytic service category.

How many professional accountants think of their job as "rearranging abstract symbols using a variety of analytic and creative tools". Not many. Most professional accountants just do the work. Besides, what the heck is an "abstract symbol"?

Computers can assist professional accountants in this task of rearranging abstract symbols. That is, if information is represented using machine-readable structured data and metadata.

A statement of financial condition and a statement of financial position are abstract ideas invented by humans. The stuff in the report is symbols. Rearranging abstract symbols, such as the creation of such financial reports, can be achieved using human-based processes and tools or using machine-based processes and tools. Likely a combination of human and machine based processes will be employed in the future; humans performing the tasks they do best, computers performing tasks that they do best.

1.3. Learning to code will not give you what you need

The "Learn to code" movement has become an international phenomenon. The movement is not just about getting everyone to learn to write software code, it also includes increasing everyone's understanding of computer science. This movement is misguided. I am not alone in having this view. Techcrunch suggests that coding is not the new literacy[13]. This blog post provides good information about the coding hysteria[14].

In the article *The End of Code*[15], Wired states, "Soon we won't program computers. We'll train them like dogs." That is a very succinct and accurate statement. As I will point out later in this document, business professionals will influence the behavior of software not by coding, but by configuring business rules.

---

[12] Robert B. Reich, *Saving Capitalism*, Alfred A. Knopf, page 204-206, (see a summary from *The Work of Nations* here http://www.oss.net/dynamaster/file_archive/040320/e8eb8748abfe77204a145d5fbcc892fb/OSS1993-01-37.pdf#page=6)

[13] Techcrunch, *Please do not learn to code*, https://techcrunch.com/2016/05/10/please-dont-learn-to-code/

[14] Hindley Milner, *On the "Learn to Code" Movement and Its Lies*, https://medium.com/@hindleymilner/on-the-learn-to-code-movement-and-its-lies-1982aab8aa8e

[15] Wired, *The End of Code*, http://www.wired.com/2016/05/the-end-of-code/

Please note that I disagree with one thing Wired is saying. Business professionals will train software using business rules created by the business professionals, not using deep learning and machine learning. Financial reporting is not the sort of thing machine learning was designed for. So don't believe the snake oil salesmen who tell you otherwise. I will get into this in more detail in later sections.

Further, if you are judging what you have to know about how to get computers to do work based on the current software that is available to create structured data and you don't consider the *Law of Conservation of Complexity* and the *Law of Irreducible Complexity*[16], then you are apt to reach the wrong conclusion as to what professional accountants and auditors even need to understand. Most structured data tools today expose far, far too much technology "stuff" to users of the software. That technical stuff will be buried far more deeply within software in the future.

Again I say, what professional accountants and auditors need is to understand how computers work and how to control the workings of computers to accurately understand what computers are capable of doing and what they are not capable of doing. You want to get the correct training? Go test drive a Tesla. Literally. If you go try out the driver assist feature and then think about your experience, you will learn way more about how machines like that work than if you learned how to code.

1.4. Learning XBRL technical syntax will not provide you what you need

Learning XBRL technical syntax will not provide you with what you need to understand either. Granted, a brief XBRL primer[17] would not hurt. Some specialists might choose to understand structured data at the technical syntax level, but the average professional accountant will not need this level of understanding. Why? Because the technical aspects of XBRL-based structured data will be buried deeply within software.

Do you know that today public companies that provide XBRL-based financial reports to the SEC are getting the XBRL technical syntax 99.99% correct; but are reporting their fundamental accounting concepts only 98.88% correct[18]? Only 83.8% of public companies get all of these basic fundamental accounting concept relations consistent with US GAAP in a report.

Neither learning to code nor learning the XBRL technical syntax will help professional accountants not make these sorts of mistakes. What professional accountants need to understand is the conceptual model of a financial report and a crash course in formal logic[19]. A CFA Institute paper, Data and Technology: Transforming the Financial Information Landscape points out,

> "The current system presumes that information is consumed by humans; in other words, it assumes a human consumption model, not a machine-readable format."

---

[16] *Understanding the Law of Conservation of Complexity*, http://xbrl.squarespace.com/journal/2015/5/24/understanding-the-law-of-conservation-of-complexity.html

[17] Charles Hoffman et. al., *XBRL for Dummies*, Chapter 4, XBRL Primer, https://www.amazon.com/XBRL-Dummies-Charles-Hoffman/dp/0470499796

[18] *Public Company Quantity Continues to Improve, 84% are Consistent*, http://xbrl.squarespace.com/journal/2016/7/1/public-company-quality-continues-to-improve-84-are-consisten.html

[19] YouTube.com, *Crash Course in Formal Logic, Part 1*, https://www.youtube.com/watch?v=ywKZgjpMBUU

Computers work differently than humans. What a human needs and what a computer needs are different. To understand what a computer needs, one needs to understand the true capabilities of computers and what it takes to make a computer do what you want it to do.

Remember the Wired article mentioned above which made the statement, "Soon we won't program computers. We'll train them like dogs." If you are going to train a dog you have to understand how dogs think and operate.

This document will help you realize that professional accountants need to understand the conceptual model of a financial report and machine-readable business rules that help keep information they represent within an XBRL-based financial report consistent with that conceptual model.

Professional accountants can create human-readable, paper-based financial reports with one hand tied behind their back. Now they need to be properly equipped to create machine-readable structured data that follows the rules of formal logic, the language of computers, and communicates the same financial position and financial condition as the paper-based information.

1.5. Understanding what computers cannot do

Key to understanding what work computers are capable of doing is understanding what computers are not capable of doing. Computers are good at repeating tasks over and over without variation. But computers are not good at any of the following sorts of tasks:

- Intuition
- Creativity
- Innovation
- Improvising
- Exploration
- Imagination
- Judgement (such as making a tough decision from incomplete information)
- Politics
- Law
- Unstructured problem solving
- Non-routine tasks
- Identifying and acquiring new relevant information
- Compassion

Some might argue that computers can be made to mimic some of the sorts of tasks in the list above. While such arguments might be valid, performance of computers in those sorts of tasks would likely be very costly and yield results that do not meet expectations. In other words, while theoretically possible using computers for such tasks is not practical.

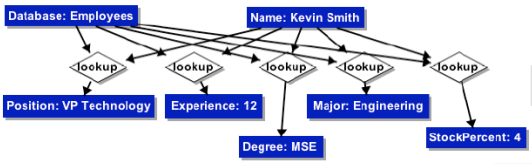
1.6. Understanding information

Most business professionals understand the notion of data and may even understand a few things about relational databases. But data and information are not the same thing.

Relational databases store data. If you took a relational database out from under one software application and then connected it to a different software application would the second application understand the data in the database? The answer is no. However, when you take an XBRL-based public company financial report out from an application (i.e. the creation software) and then connect it to another software application (i.e. the SEC Interactive Data Viewer) you can move the information and either application understands the exact same set of information. In fact, you could exchange that information to any of the 30 different software creation tools or other software vendors and each application would understand the information.

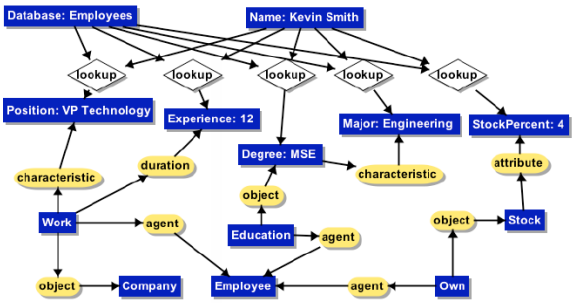
Database Values as Data

Name	Position	Yrs Experience	Degree	Major	Percent Stock
Karen Jones	VP Marketing	18	MBA	Marketing	3
Kevin Smith	VP Technology	12	MSE	Engineering	4
Keith Williams	VP Finance	15	BS	Accounting	3



Database Values as Information (data in context)

Name	Position	Yrs Experience	Degree	Major	Percent Stock
Karen Jones	VP Marketing	18	MBA	Marketing	3
Kevin Smith	VP Technology	12	MSE	Engineering	4
Keith Williams	VP Finance	15	BS	Accounting	3



Information is data in context. That context information is not stored in a relational database. The graphic above shows the context information which is basically additional business rules that explain the data in more detail, put that data into context, turn the data into information, and then allow the information to be exchanged between different software systems.

1.7. Difference between data, information, knowledge, and wisdom

There are specific differences between data, information, knowledge, and wisdom. This breakdown helps you understand the differences[20]:

- **Data**: The basic compound for intelligence is data. Data are measures, observations, symbols, phenomenon, utterances, and other such representations of the world around us presented as external signals and picked up by various sensory instruments and organs. Simplified: data is raw facts and numbers.
- **Information**: Information is produced by assigning relevant meaning related to the context of the data to the data. Simplified: information is data in context.

[20] Gene Bellinger, Durval Castro, Anthony Mills; *Data, Information, Knowledge, and Wisdom*; Retrieved February 24, 2016, http://www.systems-thinking.org/dikw/dikw.htm

- **Knowledge**: Knowledge is the understanding or interpretation, a justifiable true belief, of information and approach to act upon the information in the mind of the perceiver. Simplified: knowledge is the interpretation of information.
- **Wisdom (or Intelligence or Understanding)**: Intelligence or wisdom embodies awareness, insight, moral judgments, and principles to construct new knowledge and improve upon existing understanding. Simplified: wisdom is the creation of new knowledge.

An absence of data is noise. This functional difference between data, information, knowledge, and wisdom is called the DIKW pyramid[21].

1.8. Scenario from financial reporting to consider

Consider the following scenario from financial reporting and keep this scenario in the back of your mind as you read the information in this section. This scenario is what needs to be achieved to make digital financial work effectively.

> Two public companies, A and B, each have some knowledge about their financial position and financial condition. They must communicate their knowledge to an investor who is making investment decisions which will make use of the combined information so as to draw some conclusions. All three parties are using a common set of basic logical principles (facts known to be true, deductive reasoning, inductive reasoning, etc.) and common financial reporting standards (i.e. US GAAP, IFRS, etc.), so they should be able to communicate this information fully, so that any inferences which, say, the investor draws from public company A's input should also be derivable by public company A using basic logical principles and common financial reporting standards, and vice versa; and similarly for the investor and public company B.

The ultimate goal is to successfully communicate information between these three parties. In other words, the goal is the meaningful exchange of information between economic entities which create financial reports and the users of those financial reports. In the past, information was exchanged only in human-readable form. In the digital age, information will not only be readable by humans, but also will be machine-readable.

The current system presumes that information is consumed by humans; in other words, it assumes a human consumption model, not a machine-readable format. Humans and machines need different things in order to consume information effectively. Knowledge engineering relates to providing what machines need to effectively exchange such information.

1.9. Fundamental challenge: meaningful exchange of information

The fundamental challenge to get computers to perform useful work is the meaningful exchange of information between business systems. The only way a meaningful exchange of information can occur is the prior agreement as to[22]:

- technical syntax rules,

---

[21] Wikipedia, *DIKW Pyramid*, retrieved February 24, 2016; https://en.wikipedia.org/wiki/DIKW_Pyramid

[22] Video, *Introduction to HL7*, slide 4, Retrieved February 24, 2016, http://www.hl7.org/documentcenter/public_temp_D8292793-1C23-BA17-0C1CAB3A901C5581/training/IntroToHL7/player.html

- business domain semantics rules, and
- business domain workflow rules.

ISO TR 9007:1987 ("Helsinki principles") says[23] this in a slightly different way:

- Any meaningful exchange of utterances depends upon the prior existence of an agreed set of semantic and syntactic rules
- The recipients of the utterances must use only these rules to interpret the received utterances, if it is to mean the same as that which was meant by the utterer

1.10. Distinguishing technical syntax and domain semantics

One important aspect which you need to understand to understand the notion of a meaningful information exchange is the difference between syntax and semantics[24].

- **Syntax** is *how* you say something
- **Semantics** is the *meaning* behind what you said

Technical syntax is used to physically exchange information. Technical people are concerned with syntax. There are a handful of things that business professionals care about when it comes to technical syntax, mainly the power or expressiveness of the syntax.

Business professionals are far more concerned with semantics, the meaning behind what is being said. We will get into this more later, for now just recognize that syntax and semantics are two different things.

1.11. Important role of rules

Rules prevent anarchy. The Merriam-Webster dictionary defines anarchy[25] as "a situation of confusion and wild behavior in which the people in a country, group, organization, etc., are not controlled by rules or laws." Rules prevent information anarchy.

Rules guide, control, suggest, or influence behavior. Rules cause things to happen, prevent things from happening, or suggest that it might be a good idea if something did or did not happen. Rules help shape judgment, help make decisions, help evaluate, help shape behavior, and help reach conclusions.

Technical syntax rules arise from the best practices of information technology professionals. Business domain semantic rules[26] arise from the best practices of knowledgeable business professionals. A business rule is a rule that describes, defines, guides, controls, suggests, influences or otherwise constrains some aspect of knowledge or structure within some problem domain.

Don't make the mistake of thinking that business rules are completely inflexible and that you cannot break rules. Sure, maybe there are some rules that can never be broken. Maybe there are some rules that you can break. It helps to think of breaking rules as

---

[23] *Common Logic in Support of Metadata and Ontologies*, page 5, Retrieved June 24, 2016, http://cl.tamu.edu/docs/cl/Berlin_OpenForum_Delugach.pdf

[24] YouTube.com, *Introduction to the Semantic Web*, https://www.youtube.com/watch?v=OGq8A2zfWKq

[25] Anarchy definition, Merriam-Webster, http://www.merriam-webster.com/dictionary/anarchy

[26] *Comprehensive Introduction to Business Rules for Professional Accountants*, http://xbrlsite.azurewebsites.net/2016/Library/ComprehensiveIntroductionToBusinessRulesForProfessionalAccountants.pdf

penalties in a football game. The point is that the guidance, control, suggestions, and influence offered by business rules is a choice of business professionals. The meaning of a business rule is separate from the level of enforcement someone might apply to the rule.

Please see the *Comprehensive Introduction to Business Rules for Professional Accountants* for more information on the important topic of business rules.

1.12. Understanding workflow rules

Workflow[27] the sequence of processes/tasks through which a piece of work passes from initiation to completion. There are two categories of business workflow systems or models that business rules should be able to express rules that operate in both worlds.

- **Process-centric workflows** generally use business rules at the workflow task level to manage workflow tasks.
- **Data-centric workflows** generally use business rules within workflows to make decisions about individual items of data.

This is not an "either/or" situation, but rather leveraging both workflow models in the design and execution of workflows is the way to go.

By combining the two different workflow models, business rules can be undertaken at both the task level for automating different decisions and at the data level for implementing filters over the data. Business rules can also be used to define operational features of a workflow, such as what to do when a specific task fails. Different standard approaches exist for representing workflow rules in machine-readable form including:

- Business Process Modeling (BPM)[28]
- XML Process Definition Language (XPDL)[29]
- Business Process Execution Language (BPEL)[30]

1.13. Shared view of reality to achieve a specific purpose

In his book[31] *Data and Reality*, William Kent provides an excellent summary that discusses the realities of sharing information. In Chapter 9: Philosophy in the Third Edition and Chapter 12: Philosophy in the first edition (which is available online) he paints a picture of why you want to go through the trouble of sharing information using machine-based processes and the realities of what that takes. This is what William Kent points out which I have paraphrased as it relates to financial reporting:

> To create a shared reality to achieve a specific purpose: To arrive at a shared common enough view of "true and fair representation of financial information" such that most of our working purposes, so that reality does appear to be objective and stable. So that you can query information reliably, predictably, repeatedly, safely.

---

[27] Wikipedia, *Workflow*, retrieved August 30, 2016, https://en.wikipedia.org/wiki/Workflow

[28] Wikipedia, *Business Process Modeling*, retrieved August 30, 2016, https://en.wikipedia.org/wiki/Business_process_modeling

[29] Wikipedia, *XML Process Definition Language*, retrieved August 30, 2016, https://en.wikipedia.org/wiki/XPDL

[30] Wikipedia, *Business Process Execution Language*, retrieved August 30, 2016, https://en.wikipedia.org/wiki/Business_Process_Execution_Language

[31] William Kent, *Data and Reality*, Technics Publications, (See this resource which has CHAPTER 12: Philosophy from the first version of this book, http://www.bkent.net/Doc/darxrp.htm)

Meaningful information exchange that is reliable, repeatable, predictable, safe, cost effective, easy to use, robust, scalable, secure when necessary, auditable (track provenance) when necessary.

Prudence dictates that using the information contained in a digital financial report should not be a guessing game. Safe, reliable, repeatable, predictable, reuse of reported financial information using automated machine-based processes is obviously preferable to a guessing game.

The effective meaningful exchange of information is created by skilled craftsmen that know their craft well. The balance the system, bring it into equilibrium to achieve a specific purpose. Creating this shared view of reality which allows this specific purpose to be achieved has benefit to the financial reporting supply chain.

That purpose should be clearly defined so that everyone understands the objective and exactly what the system can, and cannot, deliver.

1.14. Understanding the benefits of double-entry accounting

Single-entry bookkeeping[32] is how 'everyone' would do accounting. In fact, that is how accounting was done before double-entry bookkeeping was invented.

Double-entry bookkeeping[33] adds an additional important property to the accounting system, that of a clear strategy to identify errors and to remove them from the system. Even better, it has a side effect of clearly firewalling errors as either accident or fraud. This then leads to an audit strategy. Double-entry bookkeeping is how professional accountants do accounting.

Double-entry bookkeeping, invented by the Italian mathematician and Franciscan Friar Luca Pacioli[34], is one of the greatest discoveries of commerce, and its significance is difficult to overstate.

Which came first, double-entry bookkeeping or the enterprise[35]? Was it double-entry bookkeeping and what it offered that enable the large enterprise to exist; or did the large enterprise create the need for double-entry bookkeeping?

Accountants think differently than non-accountants. Non-accountants don't realize this and accountants seem to forget. The quality difference between the set of "stuff" that makes up a financial report and all the support for that financial report tends to be much higher than the quality level of non-financial information, or rather, information that is managed by a non-accountant. Why? Because double-entry bookkeeping is ingrained in the processes and procedures of accountants.

Accountants need to keep this in mind as they design digital financial reporting.

---

[32] Wikipedia, *Single-entry Bookkeeping System*, retrieved August 30, 2016, https://en.wikipedia.org/wiki/Single-entry_bookkeeping_system

[33] Wikipedia, *Double-entry Bookkeeping System*, retrieved August 30, 2016, https://en.wikipedia.org/wiki/Double-entry_bookkeeping_system

[34] Wikipedia, *Luca Pacioli*, retrieved August 30, 2016, https://en.wikipedia.org/wiki/Luca_Pacioli

[35] Ian Grigg, *Triple Entry Accounting, A Very Brief History of Accounting, Which Came First - Double Entry or the Enterprise?,* http://iang.org/papers/triple_entry.html

2. Computers, Metadata, and Conceptual Models

Computers are machines. Machine-readable metadata that is also understandable to humans is key in getting computers to perform work for business professionals. A conceptual model is metadata.

2.1. Machine-readable metadata which is also readable by humans

Metadata[36] is simply data that provides information about other data. Machine-readable metadata adds perspective and context to data. People sometimes get into philosophical debates about what is data and what is metadata, but this is to completely miss the point.

This is what you need to know about metadata. Metadata is a good thing. More metadata is better. Standard metadata is even better. An example of metadata is the card catalog of a library. Metadata is generally organized into some sort of classification system.

There are three types of metadata[37]:

- **Descriptive**: describes and identifies information
- **Structural**: organizes the types and parts of information and how the parts are related to one another
- **Administrative**: provides other information that helps use some sort of system.

2.2. Three orders of order

In his book *Everything is Miscellaneous*[38], David Wenberger points out the three orders of order:

- **First order of order**. Putting books on shelves is an example the first order of order. (data)
- **Second order of order**. Creating a list of books on the shelves you have is an example of second order of order. This can be done on paper or it can be done in a database. (metadata)
- **Third order of order**. Adding even more information to information is an example of third order of order. Using the book example, classifying books by genre, best sellers, featured books, bargain books, books which one of your friends has read; basically there are countless ways to organize something. (more metadata)

David Wenberger also points out that metadata has strategic implications. Third order removes the limitations which people seem to assume exist when it comes to organizing information. Wenberger says this about the third order of order:

> "In fact, the third-order practices that make a company's existing assets more profitable, increase customer loyalty, and seriously reduce costs are the Trojan horse of the information age. As we all get used to them, third-order practices undermine

[36] Wikipedia, *Metadata*, https://en.wikipedia.org/wiki/Metadata
[37] YouTube, *Basics of Metadata*, https://www.youtube.com/watch?v=-0vc6LeVa14
[38] David Wenberger, *Everything is Miscellaneous*, Holt Paperbacks, 2007, page 17-23; https://goo.gl/oj8mkf

some of our most deeply ingrained ways of thinking about the world and our knowledge of it."

2.3. Classification systems

A classification system is a logical grouping of something based on some similarity or criteria. A classification system is a communications tool. A classification system structures information. A classification system can be informal or formal, more rigorously or less rigorously created, readable/usable by computers, or not. A classification system can be a controlled vocabulary. Classification systems can be classified as follows:

- A **dictionary** or list is a classification system that tends to provide descriptions without much, or any, structure. Dictionaries or lists simply provide a flat inventory of terms with no relations expressed between the terms. (But even a dictionary classifieds terms into noun, verb, adverb, etc.)
- A **taxonomy** is a classification system which tends provide descriptions and a limited amount of structure generally in the form of one hierarchy into which some list of terms is categorized. Categories are basically sets. A taxonomy is a tree of categories of things with only one relation expressed so terms appear in only one location in a hierarchy of categories. A creator of a taxonomy creates concepts, creates coherent definitions for those concepts, and puts concepts into "buckets" or categories.
- An **ontology** is a classification system which tends to provide descriptions and multiple structures and therefore tends to have more than one hierarchy into which terms are categorized. So an ontology can be thought of as a set of taxonomies. An ontology can express many different types of relations which includes traits/qualities of each term. An ontology is less like a tree and more like a graph[39] (network theory). This distinction is very important. The creator of an ontology identifies and establishes models explaining how things in a given ontology are related to one another, the kinds of relationships that exist, the rules of the model. If an ontology provides enough information, it can describe a conceptual model.

2.4. Conceptual model of the real world

An **ontology**, if complete enough, can be a form of conceptual model (sometimes also called a logical model or entity-relationship model).

A **conceptual model**[40] is an abstraction of things that exist in the real world which is used to help people understand the subject or domain the model represents and build software applications. A conceptual model is composed of concepts, categories or type/classes of concepts, and rules which describe relations between types/classes of concepts.

A **theory**[41] is a prescriptive or normative statement which makes up a body of knowledge about what ought to be. A theory provides goals, norms, and standards. To theorize is to develop this body of knowledge.

---

[39] Wikipedia, *Network Theory*, retrieved February 24, 2016; https://en.wikipedia.org/wiki/Network_theory

[40] Wikipedia*, Conceptual Model,* retrieved August 14, 2016; https://en.wikipedia.org/wiki/Conceptual_model

[41] Wikipedia, *Theory*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Theory

A theory is a tool for understanding, explaining, and making predictions about a system. A theory describes absolutes. A theory describes the principles by which a system operates. A theory can be right or a theory can be wrong; but a theory has one intent: to discover the essence of some system.

A theory is consistent if its theorems will never contradict each other. Inconsistent theories cannot have any model, as the same statement cannot be true and false on the same system. But a consistent theory forms a conceptual model which one can use to understand or describe the system. A conceptual model or framework helps to make conceptual distinctions and organize ideas.

A *conceptual model*, *ontology*, and *theory* all tend to serve the same general purpose which is to describe a domain of knowledge.

Conceptual models, ontologies, and theories help us overcome the obstacles of getting a computer system to perform work. Conceptual models, ontologies, and theories are formal specifications. Formal specifications are precise, concise and unambiguous. Formal specifications are communications tools. If these are machine-readable and therefore machine-checkable notation, a wide variety of automated checks can be applied to test the conceptual model, ontology, or theory to see if they act as designed. The disciplined approach of using formal specifications means that subtle errors and oversights will be detected and corrected.

And so conceptual models, ontologies, and theories which are machine-readable serve two roles: first, to describe and second to verify against that description. When creating information it is important to verify that what has been created is consistent with the expected description. When consuming information it is important to understand that the information being consumed is consistent with the expected description. Remember: nonsense-in-nonsense-out.

2.5. Need for a framework

A conceptual model, ontology, or theory serves as a framework. For example, an XBRL-based digital financial report has a framework. The following are definitions which help you understand what a framework provides:

- A framework is a broad overview or overarching conceptual structure.
- A framework is a system or set of principles, assumptions, ideas, concepts, values, rules, laws, agreements, and practices that constitutes a way of viewing reality or establishes the way something operates.

In a Ted Talk about brain science, Jeff Hawkins points out the need for a framework and theory[42]. Thing look complicated until you understand them. Once you understand them you can create a framework and theory.

What is conspicuously lacking from the XBRL International, XBRL US, the FASB and the SEC is a broad framework let alone a theory on how to think about digital financial reports. And that is why the *Financial Report Semantics and Dynamics Theory*[43] was created.

[42] Ted Talk, *Jeff Hawkins: How brain science will change computing*, https://www.youtube.com/watch?v=G6CVj5IQkzk

[43] *Financial Report Semantics and Dynamics Theory*, http://xbrl.squarespace.com/fin-report-sem-dyn-theory/

2.6. Systems thinking

Dr. W. Edwards Deming explains systems in the video *A Theory of Systems for Educators and Managers*[44]. Deming explains that the typical way of managing a complex system is to take the system, break it into parts, and then try and manage each part as well as possible. But that does not work because it is possible to improve the performance of each part, and destroy the system as a whole. Deming put it this way when describing systemic thinking:

> "Working together is the main contribution to systemic thinking as opposed to working apart separately."

There is a difference between analysis and synthesis:

- **Analysis** is separate the whole into parts and study each part individually. Analysis is the dominate mode of thought in the western world. You cannot explain the behavior of a system by analysis. You can reveal its structure and see how it works, but you cannot understand why it works the way it works.
- **Synthesis** the combination of ideas to form a theory or system. If you want to understand why something works the way it does you use synthesis to figure that out.

You need both analysis and synthesis. Analysis tells you how. Synthesis tells you why. If you want to find out how something works you analyze it. If you want to understand why it works the way it does, you use synthesis. You cannot explain the behavior of a system through analysis.

Working together is the primary benefit of systemic thinking. This is as opposed to working apart separately. The performance of the whole is not the sum of the performance of the parts separately. The performance of a system is the product of the interactions of the parts of the system.

Idealized redesign is thinking creatively about a system. Assume a system was completely destroyed and you could do whatever you want right now to replace the system. If you don't know what to do when you could do if you can do whatever you want; how could you possibly know what to do if you can't do whatever you want?

2.7. Formal systems

A formal system[45] is defined as any well-defined system of abstract thought based on the model of mathematics. Basically, formal systems can be explained and proven to work or show system flaws and inconsistencies using the language of mathematics. Every formal system has some sort of formal language[46] that explains that system. Every formal system can be tested to see if it works using a formal proof[47].

The Chomsky Hierarchy[48] categories formal languages into groups that provide different levels of reliability. Type-0, type-1, type-2, and type-3 languages all are Turing machines[49].

---

[44] YouTube.com, *A Theory of Systems for Educators and Managers*, https://www.youtube.com/watch?v=2MJ3lGJ4OFo

[45] Wikipedia, *Formal System*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Formal_system

[46] Wikipedia, *Formal Language*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Formal_language

[47] Wikipedia, *Formal Proof*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Formal_proof

[48] Wikipedia, *Chomsky Hierarchy*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Chomsky_hierarchy

[49] Wikipedia, *Turing Machine*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Turing_machine

A theory is a tool for understanding, explaining, and making predictions about a system. A theory describes absolutes. A theory describes the principles by which a system operates. A theory can be right or a theory can be wrong; but a theory has one intent: to discover the essence of some system.

A theory is consistent if its theorems will never contradict each other. Inconsistent theories cannot have any model, as the same statement cannot be true and false on the same system. But a consistent theory forms a conceptual model which one can use to understand or describe the system. A conceptual model or framework helps to make conceptual distinctions and organize ideas.

2.8. Formal logic

Aristotle[50] is said to be the father of formal logic. Logic is a discipline of philosophy[51]. Logic[52] is the study of correct reasoning. Logic is the science of argument evaluation. You put evaluate arguments using the rules of logic to see if the argument holds to be true. An argument is a set of statements[53].

The notation of what we call elementary school arithmetic took centuries to develop[54]. But today we take mathematics for granted.

Formal logic is the basis for mathematics. Mathematics is a formal system. Formal logic is the basis for describing theories and proving theories.

Formal logic was consciously broken into two groups: **first-order logic**[55] and **higher-order logic**[56]. There is a reason for this. Systems based on first-order logic can be proven to be **sound** (all provable theory statements are true in all models) and **complete** (all theory statements which are true in all models are provable using proof theory).

Basically, higher-order logics are less well-behaved than those of first-order logic. They are less predictable and therefore less reliable and they are significantly harder to implement using computers. That is why computer systems are generally based on first-order logic.

This is all well understood by good software engineers.

2.9. Need to limit first-order logic

The full set of first-order logic is not decidable so it must be limited. Description logics[57] are a family of representational languages. *SROIQ* Description Logic[58] is one such language which is based on a fragment of first-order logic that is decidable. However, *SROIQ* Description Logic does not include the ability to represent mathematical computations because the complete set of mathematics is not decidable.

---

[50] Wikipedia, *Aristotle*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Aristotle
[51] Wikipedia, *Philosophy*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Philosophy
[52] Wikipedia, *Logic*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Logic
[53] *Crash Course in Formal Logic, Part 1*, https://www.youtube.com/watch?v=ywKZgjpMBUU
[54] *Set Theory and Foundations of Mathematics*, retrieved August 29, 2016, http://settheory.net/
[55] Wikipedia, *First-order Logic*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/First-order_logic
[56] Wikipedia, *Higher-order Logic*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Higher-order_logic
[57] Description Logics, see http://en.wikipedia.org/wiki/Description_logic
[58] *A Description Logic Primer* describes the importance of *SROIQ*, http://arxiv.org/pdf/1201.4089.pdf

Another form of first-order logic is PROLOG which is a programming language[59]. PROLOG a general purpose logic programming language that is also declarative. PROLOG is based on first-order logic. The syntax of PROLOG is derived from Horn clauses[60] which is a subset of first-order logic that is decidable. Because PROLOG is declarative, program logic is expressed represented by facts, relations, and rules. Questions are asked and then answers are provided based on the facts, relations, and rules.

PROLOG has some undesirable aspects and so it was modified even further resulting in DATALOG[61]. DATALOG is both sound and complete.

What is the exact set of first-order logic which should be used to represent systems so that they are both sound and complete and maximize the expressive power of the language?

2.10. Understanding the importance of boundaries

First-order logic is very powerful and can be used to express a theory which fully and categorically describes structures of a finite domain (problem domain). This is achieved by specifying the things of the problem domain and the relations between those things.

No first-order theory has the strength to describe an infinite domain. Essentially what this means is that the things and the relations between things which make up a problem domain must have distinct boundaries. They must be made finite.

This is not to say that such a system cannot be flexible. For example, a form is not flexible. A financial report is not a form. This is not to say, however, that a financial report cannot be finite.

Extensibility is the ability to add things to a system. *Local extensibility* is extensibility that is "inside the walls" of one organization and all extensibility is explicitly coordinated and controlled within and by one organization. For example, a chart of accounts of an organization is an example of local extensibility. You have a framework for adding accounts and you can add whatever accounts you need to the systems.

Distributed extensibility is extensibility that is not explicitly controlled and coordinated by one specific organization but rather using standards-based mechanisms and rules. For example, XBRL-based financial reports submitted to the U.S. SEC is a type of distributed extensibility because while the entire system is controlled by some standard set of rules, each reporting entity has control and can extend the system; but they must stay within a set of rules which coordinates the extensibility.

The point is that one must correctly understand the notion of finite and boundaries. Even a distributed system which is extensible can have solid boundaries if the system is engineered correctly.

2.11. Describing systems formally

Deliberate, rigorous, conscious, skillful execution is preferable to haphazard, negligent, unconscious, inept execution if you want to be sure something works. Engineering a system to make sure it works as designed is a very good thing.

---

[59] Understanding the Importance of PROLOG to Digital Financial Reporting, http://xbrl.squarespace.com/journal/2015/7/23/understanding-the-importance-of-prolog-to-digital-financial.html

[60] Wikipedia, *Horn Clause*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Horn_clause

[61] Wikipedia, *DATALOG*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Datalog

A digital financial report[62] is a type of formal system. A digital financial report is mechanical and those mechanical aspects of how such a report works can be described using a conceptual model. The *Financial Report Semantics and Dynamics Theory*[63] describes the conceptual model of a digital financial report.

A system such as the digital financial report needs to be described precisely so that professional accountants understand the mechanics of how the system works so that the system can be used effectively and so the system works how the system was intended to work.

Z Notation[64] is an ISO/IEC standard for describing systems precisely. Z Notation is used to describe safety-critical systems such as nuclear power plants, railway signaling systems, and medical devices. But Z Notation is not machine-readable.

Common Logic[65] (CL), also an ISO/IEC standard, is a framework for a family of logic languages, based on first-order logic, intended to facilitate the exchange and transmission of knowledge in computer-based systems. Common Logic is machine-readable. Further, the logic allowed to be expressed by Common Logic is consciously limited to avoid logical catastrophes[66] which cause systems to break.

Common Logic is about being practical, something business professionals generally tend to like. Common logic is a conscious compromise in order to achieve reliability, predictability, and safety. Common Logic is a "sweet spot" that achieves high expressivity but consciously gives up certain specific things that lead to catastrophic results that cause systems to potentially break making a system unsound; so that a system will be sound. Common Logic establishes boundaries, allowing creators of systems to "stay within the lines" and if you do, you get a maximum amount of expressiveness with the minimum risk of catastrophic system failure. Thus, you get a more reliable, dependable system.

Semantics of Business Vocabulary and Business Rules[67] (SBVR) is an OMG standard that was designed and built to be logically equivalent to Common Logic.

What is the point? Ask yourself why ISO/IEC and OMG would go through the trouble to create specifications such as Z Notation, Common Logic, and Semantics of Business Vocabulary and Business Rules? The answer to that question is to enable systems to be described precisely so that they can be implemented successfully using computer software.

2.12. Understanding why logical catastrophes break systems

A logical catastrophe is a failure point. Logical catastrophes must be eliminated. Systems should never have these failure points. A basic example of a catastrophic failure is creating metadata that puts a process into an infinite loop that the software will not recover from. This type of catastrophic failure is resolved by simply not allowing the conceptual model to

---

[62] *Conceptual Overview of an XBRL-based, Structured Digital Financial Report*, http://xbrlsite.azurewebsites.net/2016/Library/ConceptualOverviewOfDigitalFinancialReporting.pdf

[63] *Financial Report Semantics and Dynamics Theory*, http://xbrl.squarespace.com/fin-report-sem-dyn-theory/

[64] *Understanding the Importance of Z Notation*, http://xbrl.squarespace.com/journal/2015/9/4/understanding-the-importance-of-z-notation.html

[65] *Understanding Common Logic*, http://xbrl.squarespace.com/journal/2016/6/23/understanding-common-logic.html

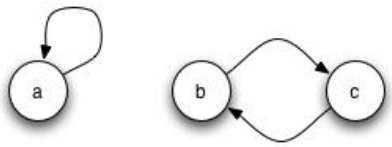
[66] *Brainstorming the Idea of Logical Catastrophes or Failure Points*, http://xbrl.squarespace.com/journal/2015/7/25/brainstorming-idea-of-logical-catastrophes-or-failure-points.html

[67] OMG, *Semantics of Business Vocabulary and Business Rules (SBVR)*, section 2.5 Conformance of an SBVR Processor, page 7, http://www.omg.org/spec/SBVR/1.0/

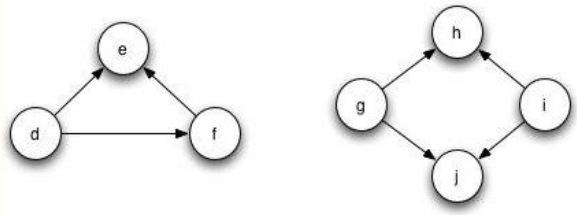
include such structures which cause the possibility of infinite loops. It really is that straight forward.

In network theory[68] there is a relation type called a directed cycle[69] which can cause infinite loops. The following graphics of a directed and undirected cycle helps you understand the potential problems of directed cycles and infinite loops:

Directed cycle



Undirected cycle



Directed cycles should be avoided and don't generally exist in many areas of reality. Tools for representing reality should not allow their users to unintentionally create directed cycles. Business professionals should be conscious of the difference between directed and undirected cycles.

Here are other types of logical catastrophes:

- **Undecidability**: If a question cannot be resolved to a TRUE or FALSE answer; for example if the computer returns UNKNOWN then unpredictable results can be returned. Logic used by a computer must be decidable.
- **Infinite loops**: If a computer somehow enters an infinite loop from which it cannot return because of a logic error or because the logic is too complex for the machine to work with; the machine will simply stop working or return nonsense.
- **Unbounded system structures or pieces**: Systems need boundaries for them to work correctly. If a system does not have the proper boundaries, then a machine can become confused or not understand how to work with information that is provided. For example, if an entirely new class of concept is added to a system that the system has no knowledge of, the system will not understand how to process that class of concept and will fail.
- **Unspecific or imprecise logic**: Confusing precise results with the capabilities of a computer to provide a statistically created result can cause problems. It is not expected that the business system at the level of describing the things in the system be able to support "fuzzy logic" or "probabilistic reasoning" or other such functionality.

[68] Network Theory, https://en.wikipedia.org/wiki/Network_theory

[69] Directed cycle described by the XBRL Technical Specification, http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#Directed-cycles

2.13. Understanding the critical importance of decidability

There are two fundamental approaches to viewing a system that one could take: the open world assumption and the closed world assumption. Formal logic and relational databases use the closed world assumption. Decidability means that a conclusion can be reached.

- In the **open world assumption** a logical statement cannot be assumed true on the basis of a failure to prove the logical statement. On a World Wide Web scale this is a useful assumption; however a consequence of this is that an inability to reach a conclusion (i.e. not decidable).
- In the **closed world assumption** the opposite stance is taken: a logical statement is true when its negation cannot be proven; a consequence of this is that it is always decidable. In other applications this is the most appropriate approach.

So each type of system can choose to make the open world assumption or the closed world assumption based on its needs. Because it is important that a conclusion as to the correct mechanics of a financial report is required because consistent and correct mechanics are necessary to making effective use of the information contained within a financial report; the system used to process a financial report must make the closed world assumption.

2.14. Setting the right expectation by understanding the capabilities of computers

First-order logic has limitations[70]. Business professionals need to understand these limitations so that they understand what computers can and cannot do, what is hard and what is easy to implement using computers, and to otherwise set their expectations appropriately. Remember, computers cannot perform magic. Computers fundamentally follow the rules of mathematics which follow the rules of formal logic. It really is that straight forward.

It is difficult to get computers to effectively work with information such as the following:

- fuzzy expressions: "It **often** rains in autumn."
- non-monotonicity: "Birds fly, penguin is a bird, but penguin does not fly."
- propositional attitudes: "Eve **thinks** that 2 is not a prime number." (It is true that she thinks it, but what she thinks is not true.)
- modal logic
    - possibility and necessity: "It is **possible** that it will rain today."
    - epistemic modalities: "Eve **knows** that 2 is a prime number."
    - temporal logic: "I am **always** hungry."
    - deontic logic: "You **must** do this."

While it is possible to implement this sort of functionality within computer systems using technologies such as probabilistic reasoning[71], those systems will be less reliable and significantly more difficult to create.

---

[70] Martin Kuba, Institute of Computer Science, *OWL 2 and SWRL Tutorial*, *Limitations of First-order logic expressiveness*, http://dior.ics.muni.cz/~makub/owl/

[71] Wikipedia, *Probabilistic Logic*, retrieved August 28, 2016, https://en.wikipedia.org/wiki/Probabilistic_logic

2.15. Limitations of classification systems

David Wenberger's book *Everything Is Miscellaneous* points out two important things to understand about classification systems:

- Every classification scheme ever devised inherently reflects the biases of those that constructed the classification system.
- The role metadata plays in allowing you to create your own custom classification system so you can have the view of something that you want.

Metadata and the correct architecture provide the flexibility necessary to create the sort of classification system you might desire which could be different that the desires of the creators of a classification system.

2.16. Need for a thick metadata layer

What is not in dispute is the need for a "thick metadata layer" and the benefits of that metadata in terms of getting a computer to be able to perform useful and meaningful work. But what is sometimes disputed, it seems, is *how* to get that thick metadata layer. There are two basic approaches to getting this metadata layer:

- **Have the computer figure out what the metadata is**: This approach uses artificial intelligence, machine learning, and other high-tech approaches to detecting patterns and figuring out the metadata.
- **Tell the computer what the metadata is**: This approach leverages business domain experts and knowledge engineers to piece together the metadata so that the metadata becomes available.

There is a lot of talk about neural networks[72] enabling things like machine learning[73] and deep learning[74]. There are two important points that business professionals tend to miss or software vendors creating such software tend to leave out of their sales pitches. First, the amount of training[75] that is necessary to get a neural network to work correctly. The training process is time consuming, expensive, and error prone.

Second, because the process is error prone; there are good uses for neural networks figuring out the "thick layer of metadata", and there are very bad uses. One description of what neural networks are best for is the following:

> Neural networks are universal approximators, and they work best if the system you are using them to model has a high tolerance to error. One would therefore not be advised to use a neural network to balance one's cheque book! However they work very well for: [76]
>
> - capturing associations or discovering regularities within a set of patterns;
> - where the volume, number of variables or diversity of the data is very great;

[72] *A Basic Introduction To Neural Networks*, http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html
[73] Wikipedia, *Machine Learning*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Machine_learning
[74] Wikipedia, *Deep Learning*, retrieved August 29, 2016, https://en.wikipedia.org/wiki/Deep_learning
[75] Hugo Larochelle, et. al., *Deep Learning and Applications in Neural Networks*, http://www.slideshare.net/hammawan/deep-neural-networks
[76] *A Basic Introduction To Neural Networks, What Applications Should Neural Networks Be Used For?*, http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html

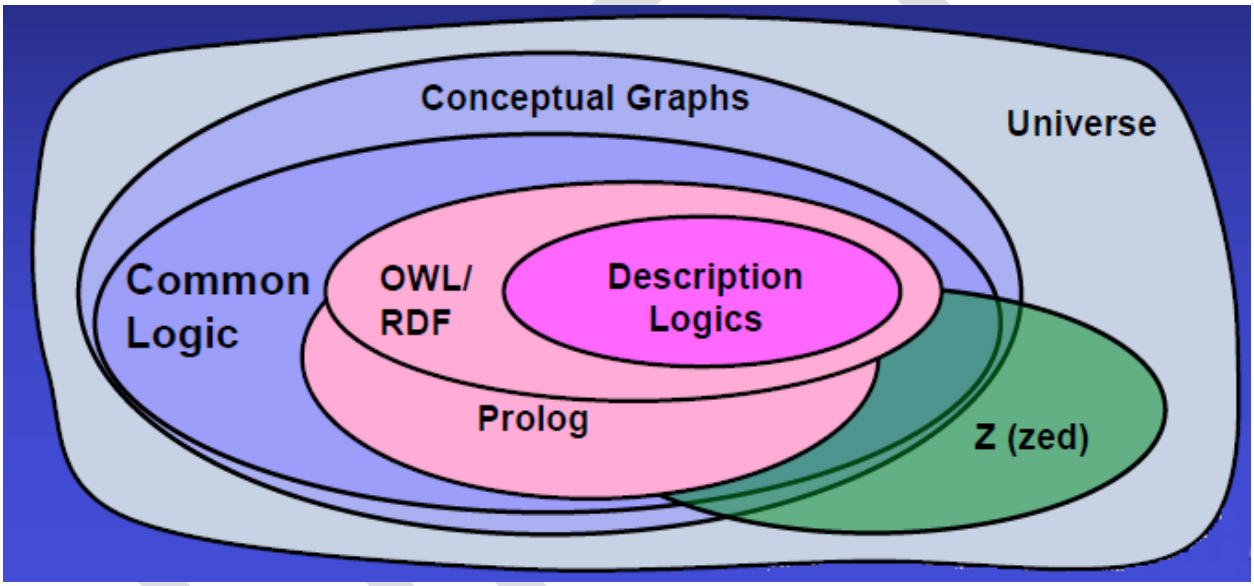
- the relationships between variables are vaguely understood; or,
- the relationships are difficult to describe adequately with conventional approaches.

And so, the probability of a neural network figuring out something like the US GAAP Financial Reporting XBRL Taxonomy is basically zero. However, that said; if such metadata is created and then that human-created metadata is used to train neural networks the probability that the neural network can create something useful goes up dramatically.

So, this is not an "either-or" proposition. This is about using the right tool for the right job and not being misguided by snake oil salesmen who don't have your interest in mind.

2.17. Comparing expressiveness

Expressiveness is the set of things that can possibly be expressed by some language. Below is a graphic which shows the relative expressiveness of Common Logic and Z Notation relative to the universe of all possible expressiveness[77].



Not even included in this comparison, because the expressiveness is so low is the Comma Separated Values[78] (CSV) technical format. CSV is a very popular data format and it is very easy to use. But CSV does nothing to help assure the quality of data represented in this technical format. Basically, you can articulate a simple list in CSV and you cannot provide information which helps a user of the information understand that the information is consistent with expectations in terms of representation (i.e. quality is high).

[77] *Common Logic in Support of Metadata and Ontologies*, Page 2, Retrieved June 24, 2016, http://cl.tamu.edu/docs/cl/Berlin_OpenForum_Delugach.pdf

[78] Wikipedia, *Comma Separated Values*, retrieved August 28, 2016, https://en.wikipedia.org/wiki/Comma-separated_values

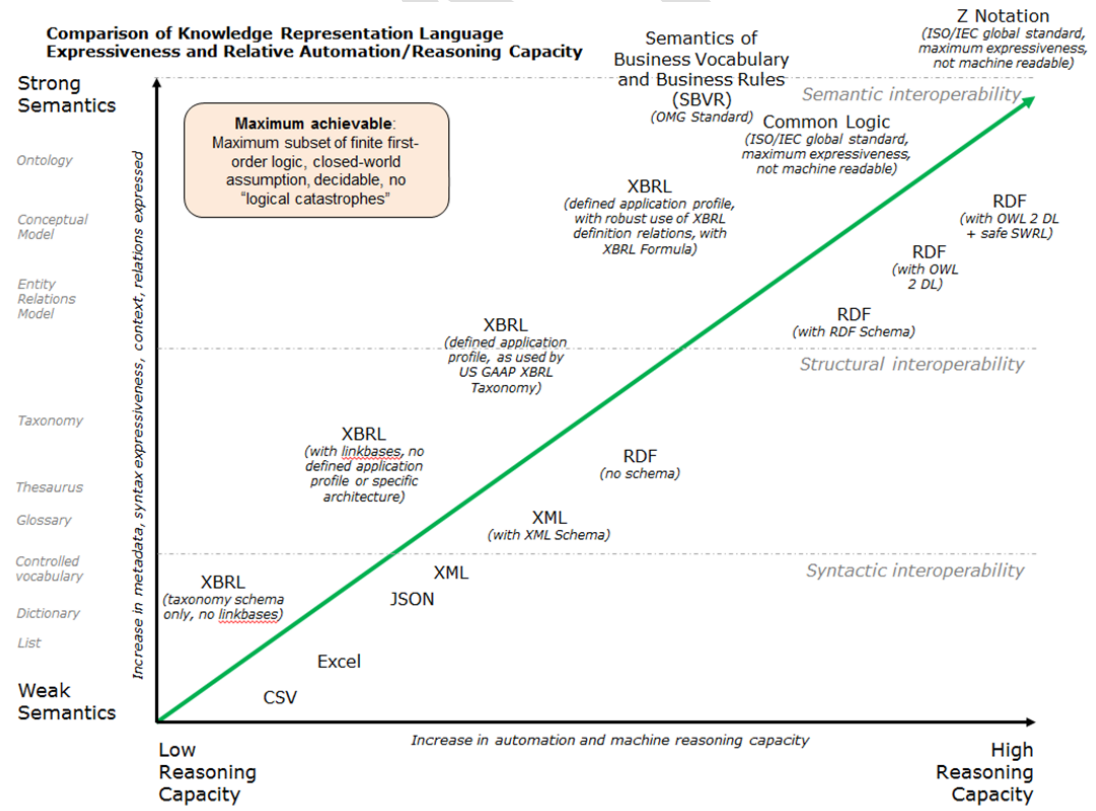
2.18. Understanding the relation between expressiveness and reasoning capacity

Why is the expressiveness of a language important? There are two reasons. First, the more expressive a language the more that language can provide in terms of describing the information being represented and verifying the consistency of what is being represented with expectations (i.e. quality).

But secondly, the more expressive the language is; the more a computer can do for a user of an application in terms of reasoning capacity. So, the two work together. Both the quality of the information being processed is higher and what the software can do is higher because of both the expressiveness of the language but also because of the quality of the information which is represented.

Another way to say this is "nonsense in, nonsense out". As has been pointed out, the only way to have a meaningful exchange of information is the prior existence of technical syntax rules (the language syntax), business domain semantics (the descriptive and structural metadata), and the workflow rules (protocols for what to do if say an amended financial report is submitted to a regulator).

This graphic below compares the relative knowledge representation language expressiveness and the relative automation and reasoning capacity which is achievable using that language.



Inspired 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/

At the bottom left hand corner of the graphic you see "CSV" which is not expressive (i.e. weak semantics). At the top left you see the ISO/IEC standard "Z Notation" which is highly

expressive (i.e. strong semantics). But remember, Z Notation is not machine-readable. But you also see Common Logic, Semantics of Business Vocabulary, and XBRL as having strong semantics. Those three formats are all machine-readable.

No knowledge representation language is 100% complete. Each has specific, knowable limitations. One must be conscious of such limitations when creating a representation of some problem domain in machine readable form.

A representation language or framework which cannot be measured for simplicity is a recipe for unnecessary complexity. Conscientious knowledge engineers are compelled to express a problem domain's conceptual model 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 languages push one more than one using a weaker language. 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.

2.19. Understanding artificial intelligence and intelligent software agents

Artificial intelligence is the automation of activities that we associate with human thinking and activities such as decision making, problem solving, learning and so on.

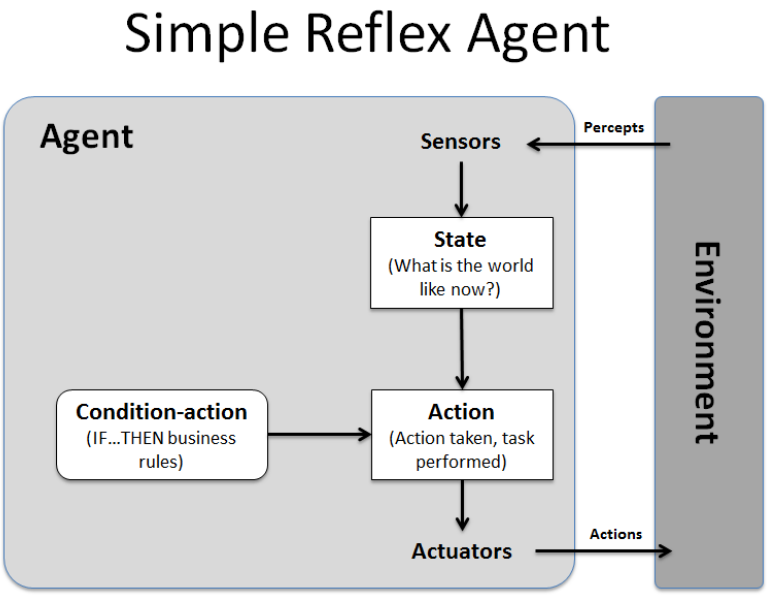
An intelligent software agent[79] is software that assists people and acts on their behalf. Intelligent agents work by allowing people to:

- delegate work that they could have done to the agent software.
- perform repetitive tasks,
- remember things you forgot,
- intelligently find, filter and summarize complex information,
- customize information to your preferences,
- learn from you and even make recommendations to you.

An agent is an entity capable of **sensing** the **state** of its **environment** and **acting** upon it based on a set of specified **rules**. An agent performs specific tasks on behalf of another. In the case of software, an agent is a software program. There are many different types of intelligent software agents[80].

[79] *Comprehensive Introduction to Intelligent Software Agents for Professional Accountants*, http://xbrlsite.azurewebsites.net/2016/Library/ComprehensiveIntroductionToIntelligentSoftwareAgentsForProfessionalAccountants.pdf

[80] Wikipedia*, Intelligent Agent,* Retrieved July 24, 2016; https://en.wikipedia.org/wiki/Intelligent_agent



The document *Comprehensive Introduction to Intelligent Software Agents for Professional Accountants* goes into significantly more detail on the topic of intelligent software agents.

2.20. Understanding expert systems

Expert systems is a branch of artificial intelligence. Expert systems are computer programs that are built to mimic or simulate or emulate human behaviour and knowledge; expert systems are computer application that performs a task that would otherwise be performed by a human expert. Expert systems solve problems by reasoning about knowledge represented in machine-readable form as "IF…THEN" rules that the machine simply follows.

Computers really are not thinking, they are only mimicking or simulating or emulating human though by following a clearly laid out set of machine-readable instructions to perform some task.

Frank Puppe[81] explains in his book *Systematic Introduction to Expert Systems* that there are three general categories of expert systems:

- **Classification or diagnosis type**: helps users of the system select from a set of given alternatives.
- **Construction type**: helps users of the system assemble something from given primitive components.
- **Simulation type**: helps users of the system understand how some model reacts to certain inputs.

A software based expert system has four primary components:

- **Database of facts**: A database of facts is a set of observations about some current situation or instance. The database of facts is "flexible" in that they apply to the

[81] Frank Puppe, *Systematic Introduction to Expert Systems, Knowledge Representations and Problem-Solving Methods*, page 11 (Note that you can read Parts I and II on Google Books here, https://books.google.com/books?id=_kKqCAAAQBAJ)

current situation. The database of facts is machine-readable. An XBRL instance is a database of facts.

- **Knowledge base**: A knowledge base is a set of universally applicable rules created based on experience and knowledge of the practices of the best domain experts generally articulated in the form of IF…THEN statements or a form that can be converted to IF...THEN form. A knowledge base is "fixed" in that its rules are universally relevant to all situations covered by the knowledge base. Not all rules are relevant to every situation. But where a rule is applicable it is universally applicable. All knowledge base information is machine-readable. An XBRL taxonomy is a knowledge base.
- **Rules processor/inference engine**: A rules processor/inference engine takes existing information in the knowledge base and the database of facts and uses that information to reach conclusions or take actions. The inference engine derives new facts from existing facts using the rules of logic. The rules processor/inference engine is the machine that processes the information.
- **Explanation mechanism**: The explanation mechanism explains and justifies how a conclusion or conclusions are reached. It walks you through which facts and which rules were used to reach a conclusion. The explanation mechanism is the results of processing the information using the rules processor/inference engine and justifies why the conclusion was reached.

Benefits from the use of expert systems include:

- **Automation**: elimination of routine, boring, repetitive, mundane, mechanical tasks that can be automate
- **Consistency**: computers are good at performing repetitive, mechanical tasks whereas humans are not; computers do not make mistakes and are good at repeating exactly the same thing each time
- **Diligence and tenacity**: computers excel at paying attention to detail; they never get bored or overwhelmed and they are always available and will keep doing their job until the task is complete with the same attention to detail
- **Reduced down-time**: computer based expert systems are tireless and do not get distracted
- **Availability**: computer based expert systems are always available simultaneously in multiple places at one time; you get quick response times and can replace absent or scarce experts
- **Training**: the best practices of the best practitioners can be available to those that are new to and learning about a domain of knowledge
- **Longevity and persistence**: computer based expert systems do not change jobs or retire so knowledge gathered by an organization can remain within that organization
- **Productivity**: computer based expert systems are cheaper that hiring experts and costs can be reduced a the same time that quality increases resulting in increased productivity
- **Multiple opinions**: Systems can integrate the view of multiple experts within a system and choose between the preferred view of multiple expert opinions in the same system

- **Objectivity**: computers apply the same inductive and deductive logic consistently; emotion and personal preferences can be eliminated where they should be eliminated

Financial report creation software of the future will be an expert system which operates similar to how CAD/CAM software for creating blueprints.

2.21. Strengths of computers

Computers have four fundamental strengths[82]:

- **Storage**: Computers can store tremendous amounts of information reliably and efficiently.
- **Retrieval**: Computers can retrieve tremendous amounts of information reliably and efficiently.
- **Processing**: Computers can process stored information reliably and efficiently, mechanically repeating the same process over and over.
- **Ubiquitous information distribution**: Computers can make information instantly accessible to individuals and more importantly other machine-based processes[83] anywhere on the planet in real time via the internet, simultaneously to all individuals.

So how do you harness this power provided by computers?

2.22. Major obstacles to harnessing the power of computers

There are a number of major obstacles to harnessing the power of computers to perform work for business professionals within one department, in an organization or across an entire supply chain. These obstacles include[84]:

- **Business professional idiosyncrasies**: Different business professionals use different terminologies to refer to exactly the same thing.
- **Information technology idiosyncrasies**: Information technology professionals use different technology options, techniques, and formats to encode information and store exactly the same information.
- **Inconsistent domain understanding of and technology's limitations in expressing interconnections**: Information is not just a long list of facts, but rather these facts are logically interconnected and generally used within sets which can be dynamic and used one way by one business professional and some other way by another business professional or by the same business professional at some different point in time. These relations are many times more detailed and complex than the typical computer database can handle. Business professionals sometimes do not understand that certain relations even exist.

---

[82] Andrew D. Spear, *Ontology for the Twenty First Century: An Introduction with Recommendations*; page 4, http://ifomis.uni-saarland.de/bfo/documents/manual.pdf#Page=4

[83] Wired: *The Web is Dead: Long Live the Internet*, http://xbrl.squarespace.com/journal/2010/9/24/wired-the-web-is-dead-long-live-the-internet.html

[84] Andrew D. Spear, *Ontology for the Twenty First Century: An Introduction with Recommendations*; page 4, http://ifomis.uni-saarland.de/bfo/documents/manual.pdf#Page=4

- **Computers are dumb beasts**: Computers don't understand themselves, the programs they run, or the information that they work with. Computers are "dumb beasts". What computers do can sometimes seem magical. But in reality, computers are only as smart as the metadata they are given to work with, the programs that humans create, and the data that exists in databases that the computers work with.

If two computers use the same information formats and other technology aspects but use different terminology or different information organization strategies, the two computers will find it difficult or even impossible to interoperate. If this is the case, the only way to cross the chasm between these two different computers is with human intervention. Often this involves re-keying information. Saying this another way, in order for two computers to interoperate it is essential that every aspect including terminology, world view, information formats, instructions and so forth necessary to translate from one computer to the second computer must be explicitly provided.

2.23. Business rules prevent anarchy

The Merriam-Webster dictionary defines anarchy[85] as "a situation of confusion and wild behavior in which the people in a country, group, organization, etc., are not controlled by rules or laws." Business rules prevent information anarchy[86].

Business rules[87] guide, control, suggest, or influence behavior. Business rules cause things to happen, prevent things from happening, or suggest that it might be a good idea if something did or did not happen. Business rules help shape judgment, help make decisions, help evaluate, help shape behavior, and help reach conclusions.

Business rules arise from the best practices of knowledgeable business professionals. A business rule is a rule that describes, defines, guides, controls, suggests, influences or otherwise constrains some aspect of knowledge or structure within some problem domain.

Don't make the mistake of thinking that business rules are completely inflexible and that you cannot break rules. Sure, maybe there are some rules that can never be broken. Maybe there are some rules that you can break. It helps to think of breaking rules as penalties in a football game. The point is that the guidance, control, suggestions, and influence offered by business rules is a choice of business professionals. The meaning of a business rule is separate from the level of enforcement someone might apply to the rule.

Business professionals interact with business rules every day and may not even realize it. Most business rules are in human readable form. But business rules can be represented in both human-readable form and machine-readable form. With the move to digital, more and more business rules are being represented in both human readable form and more importantly machine-readable form. Machine-readable business rules help automate processes which have been manual in the past.

---

[85] Merriam-Webster, *Anarchy* definition, http://www.merriam-webster.com/dictionary/anarchy

[86] *Understanding that Business Rules Prevent Anarchy*, http://xbrl.squarespace.com/journal/2016/7/15/understanding-that-business-rules-prevent-anarchy.html

[87] *Comprehensive Introduction to Business Rules for Professional Accountants*, http://xbrlsite.azurewebsites.net/2016/Library/ComprehensiveIntroductionToBusinessRulesForProfessionalAccountants.pdf

The *Business Rules Manifesto*[88] 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.

2.24. Conceptual models, ontologies and theories describe systems

Different domains use different terminology to describe the same ideas.

The term ontology has been used in philosophy for thousands of years going back to the father of formal logic, Aristotle[89] (400 B.C.). Ontology is defined as the study of the things and the relations between things that exist in reality. The goal of philosophical ontology is to provide deliberate, clear, coherent and rigorously worked out accounts of the basic structures found in reality.

In more current times, the term ontology has become prominent in the area of computer science and information science. In computer science the term ontology generally refers to the standardization of a terminology framework such that information repositories can be constructed. Ontologies used by philosophers like Aristotle were not machine-readable. Ontologies used by computers are machine-readable.

The aim of a conceptual model is to express the meaning of terms and concepts used by domain experts to discuss a problem, and to find the correct relationships between different concepts.

A theory is a tool for understanding, explaining, and making predictions about a system. A theory describes absolutes. A theory describes the principles by which a system operates. A theory can be right or a theory can be wrong; but a theory has one intent: to discover the essence of some system.

A theory is consistent if its theorems will never contradict each other. Inconsistent theories cannot have any model, as the same statement cannot be true and false on the same system. But a consistent theory forms a conceptual model which one can use to understand or describe the system.

A conceptual model, ontology, or theory all provide a framework which helps to make conceptual distinctions and organize ideas. Such frameworks overcome the four major obstacles of getting a computer system to perform work.

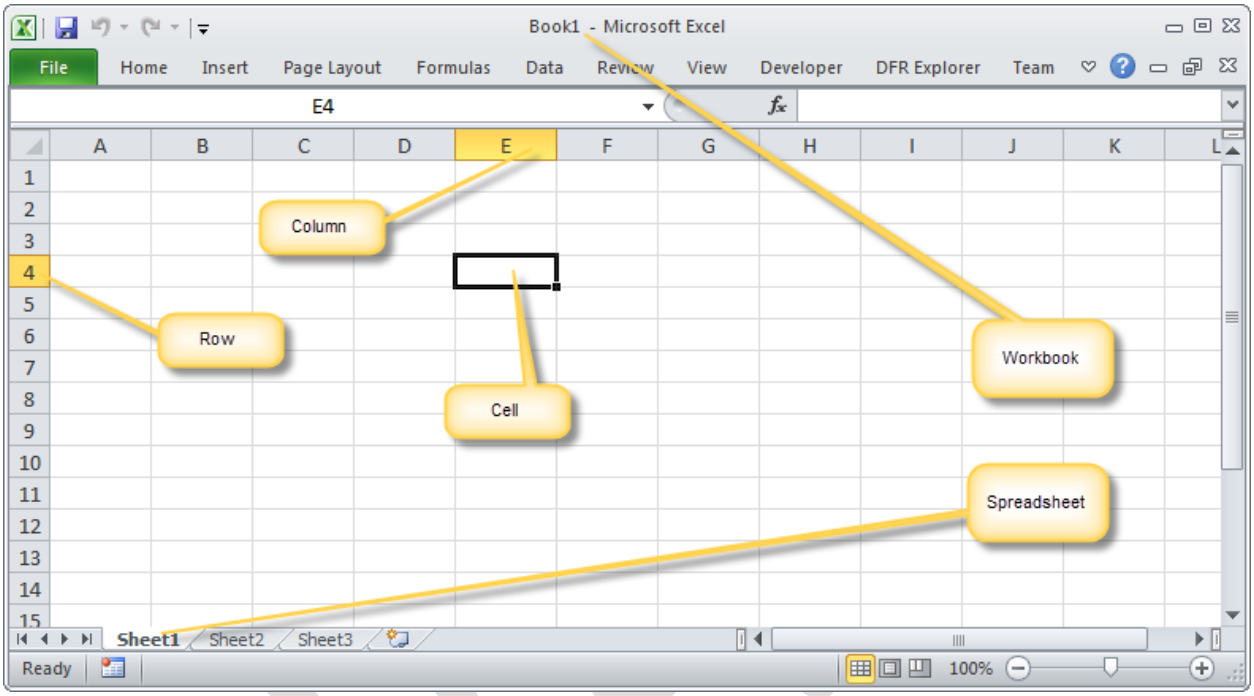
---

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

[89] Aristotle's epistemology, http://en.wikipedia.org/wiki/Aristotle#Aristotle.27s_epistemology

2.25. Using conceptual models

Business professionals work with conceptual models every day. For example, the workbooks, spreadsheets, rows, columns, and cells of an electronic spreadsheet are a conceptual model. The ease and simplicity of an electronic spreadsheet allows the average business professional to make use of this helpful tool.



2.26. Understanding the utility of the multidimensional model

Models help communication and provide a framework for understanding. The multidimensional model is a model for understanding information[90]. Every professional accountant works with multidimensional information every day and don't generally realize it.

Just like an electronic spreadsheet has a model (workbook, spreadsheet, row, column, cell); a digital financial report has a model. The model of a digital financial report follows the multidimensional model. Here are the high-level pieces of a digital financial report:

- **Fact**: A fact defines a single, observable, reportable piece of information contained within a financial report, or fact value, contextualized for unambiguous interpretation or analysis by one or more distinguishing characteristics. Facts can be numbers, text, or prose.
- **Characteristic**: A characteristic describes a fact (a characteristic is a property of a fact). A characteristic provides information necessary to describe a fact and

[90] *Introduction to the Multidimensional Model for Professional Accountants*, http://xbrl.squarespace.com/journal/2016/3/18/introduction-to-the-multidimensional-model-for-professional.html

distinguish one fact from another fact. A fact may have one or many distinguishing characteristics.

- **Fact Table**: A fact table is a set of facts which go together for some specific reason. All the facts in a fact table share the same characteristics.
- **Relation**: A relation is how one thing in a business report is or can be related to some other thing in a business report. These relations are often called business rules. There are three primary types of relations (others can exist):
    - **Whole-part**: something composed exactly of their parts and nothing else; the sum of the parts is equal to the whole (roll up).
    - **Is-a**: descriptive and differentiates one type or class of thing from some different type or class of thing; but the things do not add up to a whole.
    - **Computational business rule**: Other types of computational business rules can exist such as "Beginning balance + changes = Ending Balance" (roll forward) or "Net income (loss) / Weighted average shares = Earnings per share".
- **Grain**: Grain is the level of depth of information or granularity. The lowest level of granularity is the actual transaction, event, circumstance, or other phenomenon represented in a financial report. The highest level might be a line item on a primary financial statement such as a balance sheet.

2.27. Understanding taxonomic keys

A identification key or taxonomic key[91] is a printed or computer-aided device used for identifying some entity that is unknown. Keys are constructed so that the user is presented with a series of choices about the characteristics of the unknown thing; by making the correct choice at each step of the key, the user is ultimately led to the identity of the thing. Taxonomic keys are also helpful in classifying things into a standard taxonomy consistently.

There are two types of keys:

- **Single-access keys**: A single-access key (dichotomous key, sequential key, analytical key, or pathway key) is an identification key where the sequence and structure of identification steps is fixed by the author of the key.
- **Multi-access keys**: A multi-access key enables the user to freely choose the set and characteristics that are convenient to evaluate for the item to be identified.

Single-access keys and multi-access keys serve the same purpose. Each approach has its advantages and disadvantages.

One advantage of multi-access keys is that users can enter or select information about an unidentified thing in any order, allowing the computer to interactively rule out possible identifications of the entity and present the user with additional helpful information and guidance on what information to enter next. A disadvantage of multi-access keys is that you

[91] *Understanding Taxonomic Keys*, http://xbrl.squarespace.com/journal/2016/5/13/understanding-taxonomic-keys.html

have to understand a certain amount about a domain to use them; the more you understand about a domain the more useful multi-access keys can be.

One advantage of single-access keys is that if you don't understand the domain or don't understand enough the single-access keys can serve as bread crumbs that provide a path to the answer you are looking for.

2.28. Understanding prototype theory

Fundamentally there are two perspectives to understanding what something is:

- Aristotle's perspective was "A thing is a member of a category if it satisfies the definition of the category."
- The second perspective, **prototype theory**[92], is that we can know what something is even if it can't be clearly defined and even if its boundaries cannot be sharply drawn; concepts can be clear without having clear definitions if they're organized around undisputed examples, or prototypes.

For example, one can understand that something is a "chair" by understanding as many properties as possible about the thing you are looking at, looking at the properties of a chair as defined by a prototype (the undisputed example), and then predicting whether the thing you are looking at is a "chair" by comparing the properties you are looking at with the properties of a chair.

By contrast, the definitional view "draws sharp lines" whereas the prototype view works because "things can be sort of, kind of in a category". Prototype theory relies on our implicit understanding and does not assume that we can even make that understanding explicitly.

Computers need "handles" to be able to identify and work with things, see the section *Understanding the notion of identity*. If these handles are not provided, some other approach is necessary to work with something using a computer.

In addition to prototype theory, there is exemplar theory and multiple-prototype theory.

[92] Wikipedia, *Prototype Theory*, Retrieved February 24, https://en.wikipedia.org/wiki/Prototype_theory

3. Representing Domain Knowledge

Given the idiosyncratic tendencies of business professionals; interpretations which reflect the arbitrary peculiarities of individuals can sometimes slip in or mistakes can be made when expressing such terminology. Further, parts of our understanding of a business domain can be incorrect and even evolve, improve, or simply change over time.

If different groups of business professional use different terminology for the same concepts and ideas to express the exact same truths about a business domain; those business professionals should be able to inquire as to why these arbitrary terms are used, identify the specific reasoning for this, and specifically identify concepts and ideas which are exactly the same as other concepts and ideas but use different terminology or labels to describe what is in fact exactly the same thing. But to also understand the subtleties and nuances of concepts and ideas which are truly different from other concepts and ideas.

If idiosyncrasies result only in different terms and labels which are used to express the exact same concepts and ideas; then mappings can be created to point out these different terms used to express those same concepts and ideas. Such mappings make dialogue more intelligible and could get groups to accept a single standardized term or set of terminology for the purpose of interacting with common repositories of business information.

If the difference in terminology and expression are rooted in true and real theoretical differences between business professionals, and the different terms express and point out real and important subtleties and nuances between what seemed to be the same terms; then these differences can be made conscious, explicit, clear, and therefore they can discussed, in a rigorous and deliberate fashion because the differences are consciously recognized.

Business professionals must do this work. This work cannot be delegated to information technology professionals because they do not understand the subtleties and nuances of a business domain.

To perform this work, business professionals need to understand a few basic principles of representing knowledge in the form of a conceptual model, ontology, or theory.

3.1. Understanding the notion of identity

Computers need a way to grab onto information, a "handle" which software can use to identify something, grab it, and work with it. Things may or may not be unique. What does "us-gaap:StatementTable" mean if it is used to represent different things? To differentiate one use of us-gaap:StatementTable from another you need more information.

- **Isomorphic**: Has one meaning.
- **Polymorphic**: Has more than one meaning.

If no unique identity exists, a composite of multiple pieces of information might provide a unique identifier. For example, you need both the network and table to uniquely identify the fragments of an XBRL-based financial report submitted to the SEC by public companies. If unique identifiers are not available, then prototype theory can be used to identify the unique pieces of a financial report. (See the section *Understanding prototype theory* for more information.)

3.2. Differentiating a notion/idea/phenomenon, a name, and a preferred label

It is important to understand and properly differentiate between the following three things:

- **Notion, idea, phenomenon**: something that exists in reality that needs to be represented
- **Name**: helps computers identify some notion/idea/phenomenon that is a representation of reality within some machine-readable conceptual model
- **Preferred label**: alternative ways used to refer to name

Confusing these three things can cause problems when trying to create a conceptual model.

For example, the FASB defines the notion of "Equity" in the US GAAP conceptual framework. The FASB defines "Equity". The US GAAP XBRL Taxonomy defines the concept "us-gaap:StockholdersEquity". The FASB states specifically that "Net assets" is another preferred label for describing the notion of "Equity". "Stockholders' equity", "Partner capital", and "Proprietors' equity" are all preferred labels for the notion of "Equity".

3.3. Power of agreement

It is only through deliberate, methodical, rigorous and 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. This objective 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, convention, law, regulation, or general consent as a model or example

US GAAP contains many, many standard terms. For example, Equity, Assets, Liabilities, etc. The US GAAP XBRL Taxonomy names these terms, providing a standard.

3.4. Differentiating the important from the unimportant

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

- **Nuance**: a subtle but important difference in or shade of meaning, expression, or sound; a subtle but important distinction or variation
- **Subtle**: so delicate or precise as to be difficult to analyze or describe but important; hard to notice or see but important; not obvious but important
- **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.

Conceptual models, ontologies, and theories 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 real world. Such frameworks only need to represent the important things and serve as a "wireframe" or a "substrate" of reality. Getting bogged down in unimportant, insignificant, or inconsequential details at best serves no purpose, at worst can cause unnecessary complexity.

3.5. Difference between simplistic and simple

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**: Simplicity is "dumbing down" a problem to make the problem easier to solve. That is not what simple is about.
- **Simple**: 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.

Creating something that is simple takes conscious effort and is hard work.

3.6. Difference between a requirement and a policy

Sometimes things are required, other times things are a choice. Yet in other times setting some policy eliminates certain options which could have been previously considered.

- **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 and in light of given conditions to guide and determine present and future decisions
- **Requirement**: a thing that is needed or wanted; something that is essential or that must be done
- **Choice**: an act of selecting or making a decision when faced with two or more possibilities; the act of choosing; the act of picking or deciding between two or more possibilities
- **Option**: a thing that is or may be chosen; the opportunity or ability to choose something or to choose between two or more things

Any time a business professional is presented with an alternative; complexity increases because the business professional must choose. As the number of choices increases, complexity increases. Choices must be managed. Flexibility when it is not necessary is not a feature, it is a bug.

3.7. Differentiating between objective and subjective

There is a difference between something that is objective and something that is subjective.

- **Objective**: not influenced by personal feelings or opinions in considering and representing facts; based on facts rather than feelings or opinions; not influenced by feelings; facts are objective.

- **Subjective**: based on or influenced by personal feelings, tastes, or opinions; based on feelings or opinions rather than facts; relating to the way a person experiences things in his or her own mind; opinions are subjective.
- **Judgment**: the ability to make considered decisions or come to sensible conclusions; an opinion or decision that is based on careful thought; judgment is subjective.

Remember, computers are machines. Computers have no intelligence until they are instructed by humans. Computers only appear smart when humans create standards and agree to do things in a similar manner in order to achieve some higher purpose. It is easy to agree on things that tend to be objective. It is harder to agree where there is subjectivity. It is extremely difficult to impossible to get a machine to exercise judgment. A machine such as a computer can only mimic what humans tell the machine to do via machine-readable information.

3.8. Difference between explicit and implicit

In the process of agreeing, it is important to understand the difference between what is important and what is unimportant in that process of agreeing. It is likewise important to understand the difference between telling a machine something and requiring the machine to figure something out:

- **Explicit**: stated clearly and in detail, leaving no room for confusion or doubt; very clear and complete; leaving no doubt about the meaning.
- **Implicit**: implied though not plainly expressed; understood though not clearly or directly stated.
- **Ambiguous**: open to more than one interpretation; having a double meaning; able to be understood in more than one way; having more than one possible meaning; not expressed or understood clearly.
- **Derive or Impute**: assign (a value) to something by inference from the value of the products or processes to which it contributes; to deduce a conclusion about some fact using some other fact or facts and logical reasoning.

Machines do well with information which is explicitly provided. When information is not explicitly provided, software developers either make a choice or have to figure out ways to allow a business professional making use of the software to make a choice. Every time a software developer or business professional has to make an interpretation because something is ambiguous, there is the possibility that some unexpected or incorrect interpretation can be made. Not being explicit causes confusion and turns using ambiguous information into a guessing game.

3.9. Representing reality in a conceptual model

Attempting to describe reality can lead to complex philosophical, theoretical, and even religious debates. But if one sees the goal as not to debate, but rather to be pragmatic and achieve something useful, the discussion becomes less complicated. One can distill the perspectives one might take to viewing reality into two possible extremes.

One approach to viewing reality is to take the perspective that reality (the world) exists objectively in-and-of itself; that *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, conceptual models which provide representations of the world could get things wrong. Therefore, a conceptual model 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 completely 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, a conceptual model can represent one "reality camp". That implies that a conceptual model can be created for each reality camp. Therefore, the second approach becomes equivalent to the first approach.

And so, a conceptual model can be created to represent reality.

3.10. Pitfalls of knowledge engineering

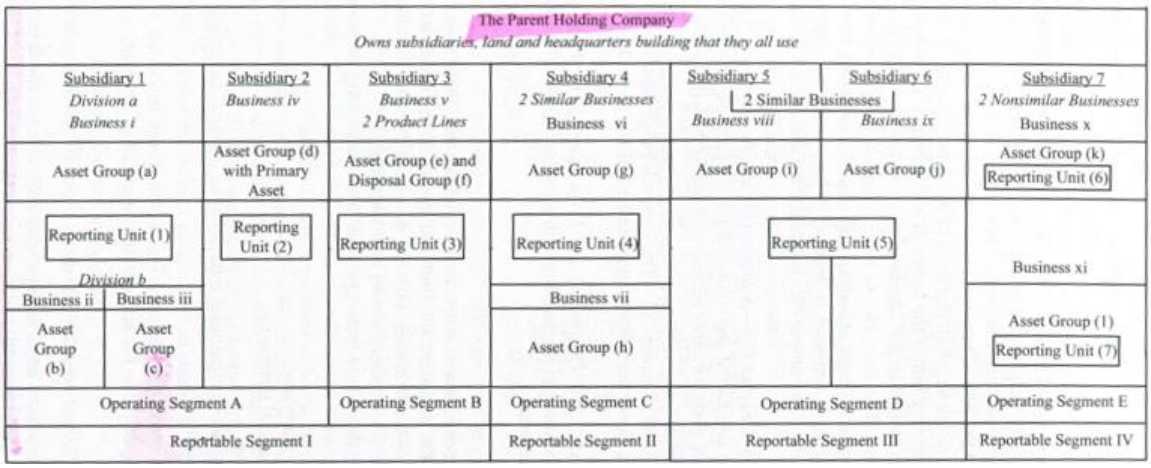
There are many different ways to stumble when trying to represent the knowledge of a problem domain in machine readable form. The following is a summary of many common pitfalls which should be recognized and then avoided.

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" conceptual model 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. Are all of these different sets of terminology necessary? Perhaps yes, perhaps no.

The following standard 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. Further, consider trying to compare US GAAP and IFRS reports if the components of an entity are inconsistent.

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 deliberate and 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 inconsistencies are generally due to sloppiness and lack of attention to detail and cannot be explained which pointed out and questioned.

3.11. 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, deliberate, rigorous testing. Another is examining empirical evidence. Testing is a 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.

3.12. Becoming conscious of the goal

As Stephen R. Covey pointed out in is seminal work *Seven Habits of Highly Effective People*[93], "Begin with the end in mind." You become conscious of what you need to do when you are conscious of the goal that you desire to achieve.

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

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

be exactly the same even if different software applications are used. Different software applications providing different results when the results should be the same is not a desirable outcome.

Conscious and skillful execution using this approach can create digital financial reporting which is simple and elegant; and yet a sophisticated and powerful tool. Information in a digital financial reports must be deliberately created to be clear, consistent, logically coherent, and otherwise unambiguous to make sure the guessing game never takes place.

- *Complete solutions* are better than *incomplete solutions*
- *Less expensive solutions* are better than *more expensive solutions*
- *Powerful solutions* are better than *simplistic solutions*
- *Easy to maintain solutions* are better than *hard to maintain solutions*
- *Easy to use solutions* are better than *hard to use solutions*
- *Good solution performance* is better than *poor solution performance*
- *More scalable solutions* are better than *less scalable solutions*
- *Standard solutions* are better than *proprietary solutions*

4. Other Useful Information

The following is other helpful information. This information tends to be more advanced and therefore more challenging for business professionals to understand.

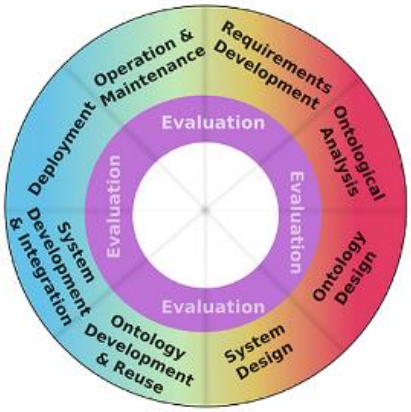
4.1. Understanding life cycle of a conceptual model

Just like many other things a conceptual model, ontology or theory has a life cycle. The paper *Towards ontology evaluation across the life cycle*[94] explains the problem of not understanding that life cycle and not being able to evaluate the quality of an ontology:

> "Problem: Currently, there is no agreed on methodology for development of ontologies, and there is no consensus on how ontologies should be evaluated. Consequently, evaluation techniques and tools are not widely utilized in the development of ontologies. This can lead to ontologies of poor quality and is an obstacle to the successful deployment of ontologies as a technology."

The paper points out that there are five aspects to the quality of ontologies which need to be evaluated: intelligibility, fidelity, craftsmanship, fitness, deployability.

The paper provides this diagram of the different stages of the ontology life cycle:



This is a list of the stages which are explained in the document: System design, Ontology design, Ontological analysis, Requirements definition, Operations/maintenance, Deployment, System development and integration, Ontology development and reuse.

4.2. Conceptual model interoperability

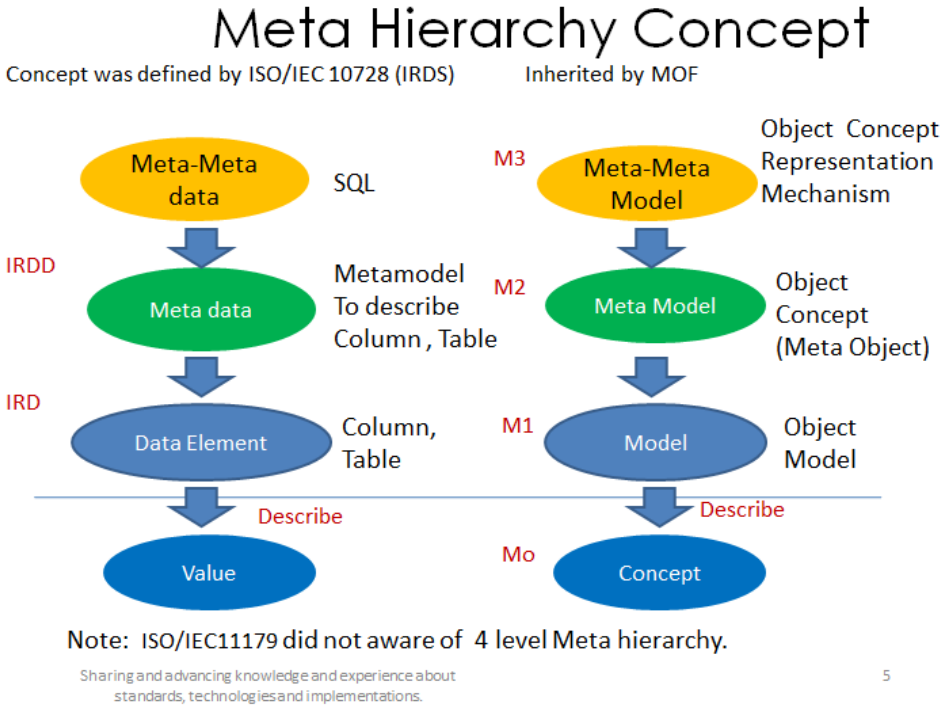
Different business domains and even different people in the same domain can create conceptual models differently. Yet, many times conceptual models need to interoperate. OMG[95] and ISO[96] have created a meta-meta model or hierarchy of concepts which are used to express conceptual models. This meta-meta model is intended to maximize interoperability between different conceptual models. This diagram provides an overview of that model.

---

[94] Towards ontology evaluation across the life cycle, http://www.researchgate.net/publication/260834360_Toward_Ontology_Evaluation_across_the_lifecycle

[95] OMG Meta Object Facility, https://en.wikipedia.org/wiki/Meta-Object_Facility

[96] ISO/IEC/IEEE 42010 Model and Metamodel, http://www.iso-architecture.org/ieee-1471/meta/



4.3. Reading list

The following books are extremely helpful in trying to knowledge engineering ideas. Anyone who wants to understand knowledge engineering in more detail should consider reading the following books to provide additional detail and build on the base of understanding derived from this conceptual overview:

Data and Reality[97], 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[98], 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.

[97] See, http://xbrl.squarespace.com/journal/2014/7/28/data-and-reality-what-is-the-purpose-of-sec-xbrl-financial-f.html
[98] See, http://xbrl.squarespace.com/journal/2011/1/31/us-gaap-taxonomy-build-it-to-allow-reoganization.html

Models. Behaving. Badly.[99], 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.

Systematic Introduction to Expert Systems: Knowledge Representation and Problem Solving Methods[100], by Frank Puppe: (350 pages) The first three chapters of this book are an excellent introduction to expert systems, about 25 pages, and is easily understandable to a business professional. The second section of this book explains how expert systems work and the moving pieces of expert systems. The last to sections get technical, but are still understandable, and provide what amounts to an inventory of problem solving approaches and how to best implement those approaches in software.

Semantic Web for the Working Ontologist[101], 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.

Ontology for the Twenty First Century: An Introduction with Recommendations[102], by Andrew D. Spear: (132 pages) The introduction first 45 pages are the best. This chapter is highly influenced by this resource. This can be challenging to make your way through but if you really want to understand all of the issues in creating useful ontologies; reading this is worth the effort.

[99] See, http://xbrl.squarespace.com/journal/2014/7/20/updated-financial-report-semantics-and-dynamics-theory.html

[100] See, http://xbrl.squarespace.com/journal/2015/7/15/understanding-expert-systems-applicability-to-financial-repo.html

[101] See, http://www.amazon.com/Semantic-Web-Working-Ontologist-Effective/dp/0123735564

[102] See, http://ifomis.uni-saarland.de/bfo/documents/manual.pdf