Comprehensive Introduction to Business Rules for Professional Accountants

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

August 4, 2016 (DRAFT)

The world is changing and professional accountants need to understand these changes in order to adapt to the new world that will exist in your future in order to remain relevant. The following is a summary of some of these changes:

- The ***volume of information is increasing***, becoming more complex, and becoming increasingly interconnected. Many people refer to this change as the move to "digital" or the "information age".
- More information is being stored in ***structured form***, for example financial reports were unstructured documents and now with technologies such as XBRL the information is being structured.
- Business rules used to be "coded" into software applications[1] which meant that changing a business rule meant that you needed to be a programmer and maintaining rules was hard. Now, using the ***business rules approach***[2], business rules are being separated from software applications making maintenance easier, enabling business professionals to manage their rules, and making systems more flexible. The business rules approach is becoming a business standard[3].

Business rules are important to business professionals. The information in this document helps you sort through these changes, understand the importance of business rules, and cut through the hype and misinformation that tends to exist about new technologies.

There are different camps with different views regarding how to implement business rules and leverage those rules to perform work for accounting professionals and other business professionals. There are the entrenched relational database software vendors that, over the years, have adapted relational database technology to meet the needs of their customers. Relational database systems offer proven, safe, reliable, predictable solutions. Then there is the semantic web camp. Their mantra is "anyone can say anything about anything" and with very flexible and powerful global standard technologies such as RDF and OWL 2 DL. There is the XBRL community with their offerings. There are niche worlds of PROLOG and DATALOG. Then there is the forty years of learning that the artificial intelligence community brings to the table. Each camp says that there answer is the best solution to every problem. What is the right answer? Does there even need to be only one right answer?

---

[1] Ronald G. Ross, *My Story: To Play the Game You Need Rules*, http://www.ronross.info/story.php

[2] Editors of the Business Rules Community, *History of the Business Rules Approach*, retrieved August 2, 2016, http://www.brcommunity.com/history.php

[3] Wikipedia, *Semantics of Business Vocabulary and Business Rules*, retrieved August 3, 2016, https://en.wikipedia.org/wiki/Semantics_of_Business_Vocabulary_and_Business_Rules

1. Understanding what Business Rules Do

The Merriam-Webster dictionary defines anarchy[4] 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[5].

Business rules guide, control, suggest, or influence behavior. Business rules cause things to happen, prevent things from happing, 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.

1.1. Human-readable and machine-readable business rules

Business professionals interact with business rules every day by 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.

1.2. Using business rules to influence software behavior

Key to making an expert system or intelligent software agent[6] work is business rules[7] of the domain being put into machine-readable form.

Artificial intelligence[8] is a branch of computer science. There are many good descriptions of artificial intelligence[9]. Here is one good definition:

---

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

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

6 Wikipedia*, Intelligent Agent,* retrieved July 24, 2016; https://en.wikipedia.org/wiki/Intelligent_agent

7 *Business Rules Manifesto*, http://www.businessrulesgroup.org/brmanifesto.htm

8 *Introduction to Artificial Intelligence Terminology*, http://xbrl.squarespace.com/journal/2016/7/21/introduction-to-artificial-intelligence-terminology.html

9 AlanTuring.net, *What is Artificial Intelligence?*, http://www.alanturing.net/turing_archive/pages/reference%20articles/What%20is%20AI.html

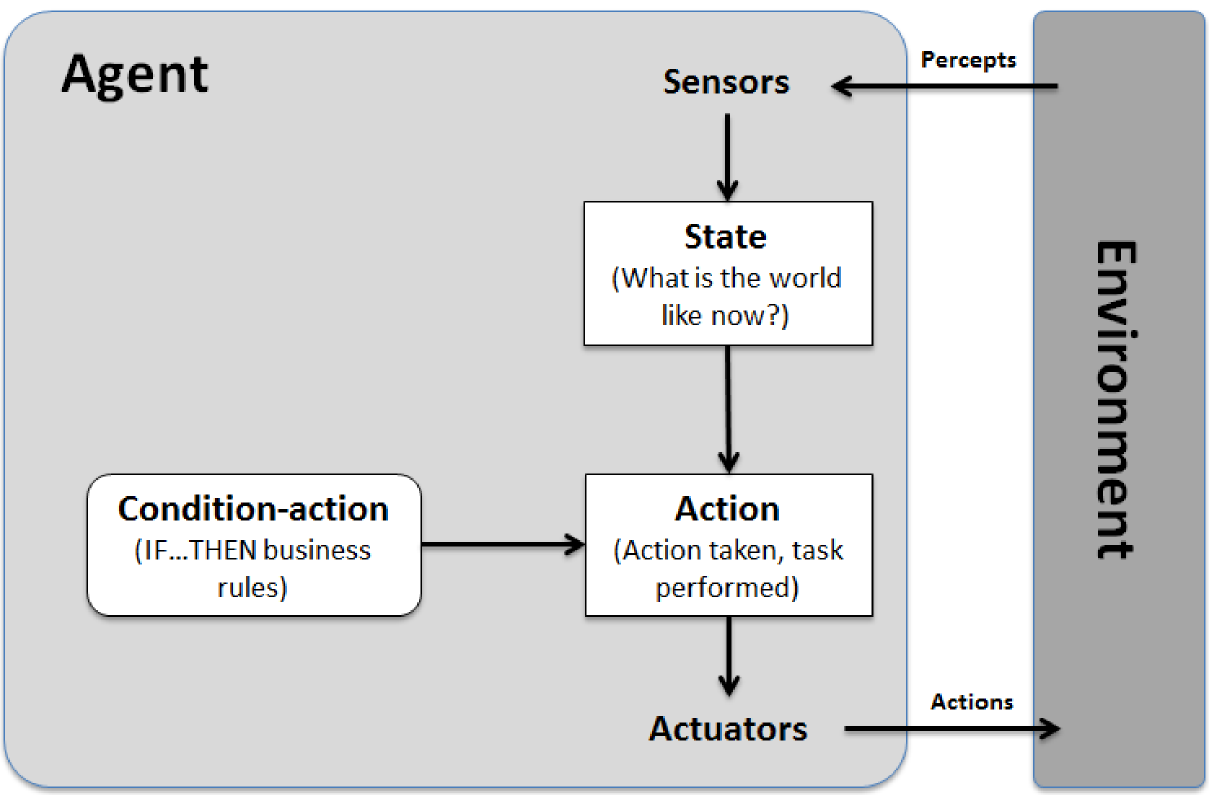
> Artificial intelligence is the automation of activities that we associate with human thinking and activities such as decision making, problem solving, learning and so on.

Those trying to make artificial intelligence work over the past 40 or so years have had limited success. But that is changing. Both under estimating or over estimating the capabilities the computer software will be able to achieve can have catastrophic consequences.

Expert systems[10] is a branch of artificial intelligence. The following is a definition of an expert system:

> Expert systems are computer programs that are built to mimic human behavior and knowledge. The computer program performs tasks that would otherwise be performed by a human expert. A model of the expertise of a domain of knowledge of the best practitioners or experts is put into machine-readable form and the expert system reaches conclusions or takes actions based on that information.

Intelligent software agents[11] are computer code written in a specific way. 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. Consider that definition of an agent and look at the graphic below to get an idea of how an intelligent agent software works:



[10] *Understanding the Components of an Expert System*, http://xbrl.squarespace.com/journal/2016/5/24/understanding-the-components-of-an-expert-system.html
[11] *Introduction to Intelligent Software Agents for Business Professionals*, http://xbrlsite.azurewebsites.net/2016/Library/02_IntroducingIntelligentAgents.pdf

An intelligent agent 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.

1.3. Business rules drive intelligent software agents and expert systems automating work

The global consultancy firm Gartner classifies XBRL as a transformational technology[12]. Gartner defines transformational as something that "*enables new ways of doing business across industries that will result in major shifts in industry dynamics*". Major shifts means lots of change and some winners and some losers.

An example of one major shift is provided by what professional accountants call the "disclosure checklist". Accountants creating financial reports often use accounting and reporting checklists or "disclosure checklists" as memory joggers to help them get the reports right[13]. These memory joggers were created to be read by humans and can be a couple hundred pages. What if a financial report was structured, such as an XBRL-based public company financial report that must be submitted to the U.S. Securities and Exchange Commission. What if these human-readable memory joggers could be made machine-readable. And what if an intelligent software agent could be created to automate the manual task of checking a financial report to make sure that report was mechanically correct.

Note the statement “mechanically correct”. This is a very important distinction. No computer program will ever have the judgement of a professional accountant. See the section *Setting the right expectations* later in this document. But computer programs can perform work if the financial report is structured and the necessary business rules are made machine-readable. How much of a disclosure checklist can be automated? That percentage is as-of-yet to be determined. Perhaps 20% can be automated or even 80% will be automated. Maybe even a higher percentage. The probability 0% of a disclosure checklist can be automated is extremely low.

1.4. Business rules are metadata and follow the rules of formal logic

Business rules provide a thick metadata layer that enables computer systems to perform useful work. The more business rules that exist, the more work a computer system can perform.

---

[12] Charles Hoffman and Liv Watson, *XBRL for Dummies*, page 145

[13] *Automating Accounting and Reporting Checklists*, http://xbrl.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html

A simple example of a business rule the accounting equation[14] which is universally accepted: *Assets = Liabilities and Equity*. The accounting equation is an axiom. *Axioms* describe self-evident logical principles that no one would argue with. The accounting equation is the foundation upon which double-entry accounting is built. *Theorems* are deductions which can be proven by constructing a chain of reasoning by applying axioms in the form of IF…THEN statements. These axioms and theorems are the basis for formal logic[15]. Formal logic is a discipline of philosophy which has been around since the days of Aristotle. Computers work based on formal logic. But you have to be careful. Computers are dumb beasts (see the section *Major obstacles to harnessing the power of computers*). Computers cannot follow all the rules of logic, only a limited set. To effectively get the tool, the computer, to perform work correctly, you need to follow certain guidelines and not push the tool beyond its capabilities.

1.5. Basic example of business rules, fundamental accounting concept relations

A basic example of business rules is the fundamental accounting concept relations[16]. The fundamental accounting concept relations build upon the accounting equation. For example, assets can be broken down into current and noncurrent portions. And so you can create the rule: Assets = Current assets + Noncurrent assets. Another rule is: Liabilities = Current liabilities + Noncurrent liabilities.

But accountants understand that not every economic entity breaks assets and liabilities down into their current and noncurrent portions. For example, banks report using an unclassified balance sheet, reporting only total assets and total liabilities.

To address this, another rule is created. Economic entities are partitioned into groups based on how they report, their reporting style. Those that provide classified balance sheets and therefore report current and noncurrent assets and liabilities are put into one group. Those that report using an unclassified balance sheet, reporting only totals for assets and liabilities, are put into a different group. And so the result is rules that are universally applicable to all economic entities. Not every economic entity uses every fundamental accounting concept relations rule, only rules that apply to the group that the economic entity is in are applicable. This is only a brief description of the fundamental accounting concept relations. See the actual fundamental accounting concept relations metadata[17] for a more comprehensive explanation.

An example of a more comprehensive set of business rules is accounting and reporting checklists[18] that are used by professional accountants to create external financial reports. Today, these business rules are organized in the form of a human-readable accounting and disclosure checklist which are used as a memory jogger by professional accountants creating a financial report.

---

[14] Wikipedia, *Accounting Equation*, retrieved August 3, 2016, https://en.wikipedia.org/wiki/Accounting_equation

[15] Wikipedia, *Logic*, retrieved August 3, 2016, https://en.wikipedia.org/wiki/Logic

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

[17] *Fundamental Accounting Concept Relations*, http://xbrl.squarespace.com/fundamental-accounting-concept/

[18] *Automating Accounting and Reporting Checklists*, http://xbrl.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html

1.6. Business rules can result in a theory

Business rules can result in a theory as to how the business domain covered by the business rules behaves. Looking at this from 180 degrees, the opposite perspective, if you have a set of business rules you can make computer software behave like the business domain.

For example, there are approximately 6,500 public companies that submit XBRL-based financial reports to the U.S. Securities and Exchange Commission. (Note that this explicitly excludes economic entities which are funds or trusts because they follow different rules which are not provided by this set of rules because I am not interested in funds or trusts.)

Those 6,500 economic entities can be broken down into 104 different reporting styles. About 80% of all economic entities fit into a set of only 12 different reporting styles, so there are some large groups and much smaller groups of reporting styles, some groups are comprised of just one economic entity.

As of the most current results[19], 99.88% of economic entities are consistent with each of the individual fundamental accounting concept relations applicable to that entity. A total of 83.8% of economic entities are consistent with all of the fundamental accounting concept relations for their specific reporting style.

Different software vendors and filing agents are more consistent with the fundamental accounting concept relations than are others. A total of 6 software vendors/filing agents have 94% or more of all of their customers consistent with these basic, fundamental accounting concept relations. Whereas there are 9 software vendors/filing agents that have 75% or less of their customers consistent with these basic relations.

Does this mean that there is some correlation between what software vendor/filing agent that is used and the fundamental accounting concept relations that exist? No. What this information points out is that different software vendors/filing agents are better, or worse, and creating XBRL-based financial reports that are consistent with the basic, fundamental accounting concept relations.

So again, now look at this from the opposite perspective. The business rules articulated by the fundamental accounting concept relations can influence the software of software vendors/filing agents so that they financial reports created are consistent with the fundamental rules of financial reporting. Said succinctly, software can help professional accountants create better financial reports.

2. Business Rules and Business Professionals

Business professionals interact with business rules every day by 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.

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

2.1. Business professionals create and maintain business rules

Business professionals create and maintain business rules. Imagine an information technology professional being responsible for maintaining the fundamental accounting concept relations rules. That simply would never work. To maintain the fundamental accounting concept relations rules one needs to have an intimate understanding of financial accounting and financial reporting.

As Article 9 of the *Business Rules Manifesto*[20] states, business rules are of, by, and for business people; not IT people.

- 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.

Rather than creating tools that only IT professionals can use because they are so complicated; business professionals need to demand software tools that properly expose functionality that exposes business rules to business users such that business users are working with business domain knowledge, not technical details that should have been buried deeply within the software applications.

Business professionals need to understand the *Law of Conservation of Complexity*[21] which states that complexity can never be removed from a system, but complexity can be moved. The *Law of Conservation of Complexity* states: "Every application has an inherent amount of irreducible complexity. The only question is: Who will have to deal with it - the user, the application developer, or the platform developer?"

2.2. Reusing and sharing business rules

The *Business Rules Manifesto*, Article 4[22], points out that business rules should be declarative rather than procedural. The declarative approach has important advantages including that your business rules become reusable across both processes and software platforms. As such, the rules become both highly re-engineerable and highly re-deployable.

Declarative involves stating THAT something is the case. Procedural involves stating HOW to do something. The following is a simple example of procedural rules and declarative: Suppose you desire a cup of coffee.

Procedural:

1. Go to kitchen.
2. Get water, coffee, sugar, cream.
3. Heat the water on the stove until the water boils.
4. Put the coffee, sugar, and cream into the water.

---

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

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

[22] *Business Rules Manifesto, Article 4. Declarative, Not Procedural*, http://www.businessrulesgroup.org/brmanifesto.htm

5. Bring the result to me.

Declarative:

1. Get me a cup of coffee.

Taking a procedural approach you define the entire process and provide each step necessary to obtain the desired result. Taking a declarative approach you state the desired result, and let the system determine the best way to get that result; all you care about is the result without worrying how the result will be achieved.

A procedure is used in only one way, but a declarative specification can be used in many different ways[23].

Next, business rules should not be mixed within software application code. Why? Three reasons. First, if business rules are within application code then it takes a programmer to change the code. Second, if the business rules are embedded within one software application that it is challenging to reuse those same rules within another application. Third, sharing business rules becomes easy.

Think of what it would be mean if you could create a spreadsheet and test the spreadsheet against a shared set of rules provided somewhere on the Internet.

This does not mean that all business rules become publically available. Using security rules could be available only to a department, or a company, or even across an entire supply chain.

Basically, just as applications and the databases they use were separated from one another (they were combined in the early days of software); business rules and applications are now being separated.

2.3. Example of business rule reuse

An example of business rule reuse can be seen via the fundamental accounting concept relations rules[24,25]. All the metadata for these rules are declarative in nature and represented using the XBRL global standard. One software vendor, XBRL Cloud, uses the fundamental accounting concept relations rules to validation XBRL-based financial filings which are submitted to the SEC[26]. Another software vendor uses the exact same metadata and completely different software to query XBRL-based financial filings[27][28].

---

[23] John F. Sowa, *Fads and Fallacies about Logic*, page 3, http://www.jfsowa.com/pubs/fflogic.pdf

[24] *Fundamental Accounting Concept Relations*, http://www.xbrlsite.com/2016/fac/v3/Documentation/#home

[25] *Crash Course in the Fundamental Accounting Concepts*, http://www.xbrlsite.com/2016/Prototype/FundamentalAccountingConceptsPureXBRL/Documentation/FundamentalAccountingConcepts.html

[26] *XBRL Cloud's Validation Report*, http://xbrlsite.azurewebsites.net/2016/ReferenceImplementation/2016-04-15/evidence-package/index.html#USFACRenderingSummary.html

[27] *Reporting Style Examples*, http://www.xbrlsite.com/2016/fac/v3/Examples/Index.html

[28] Comparison Formats, http://www.xbrlsite.com/2016/fac/v3/Examples/Index_Compare.html

2.4. Rearranging abstract symbols

Professional accountants refer to themselves as knowledge workers but few really understand what that really means. In his book *Saving Capitalism*[29], Robert Reich describes three categories that all modern work/jobs fit into: (page 204)

- **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"?

But that is what computers can do; rearrange abstract symbols, if information is represented using machine-readable metadata. Computers are good at repeating without variation. As information is rearranged, business rules help make sure mistakes are not made and even assist professional accountants during the process of performing that work.

2.5. Understanding the difference between data and information

Understanding the difference between data and information is important to gain a proper understanding of business rules[30].

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

---

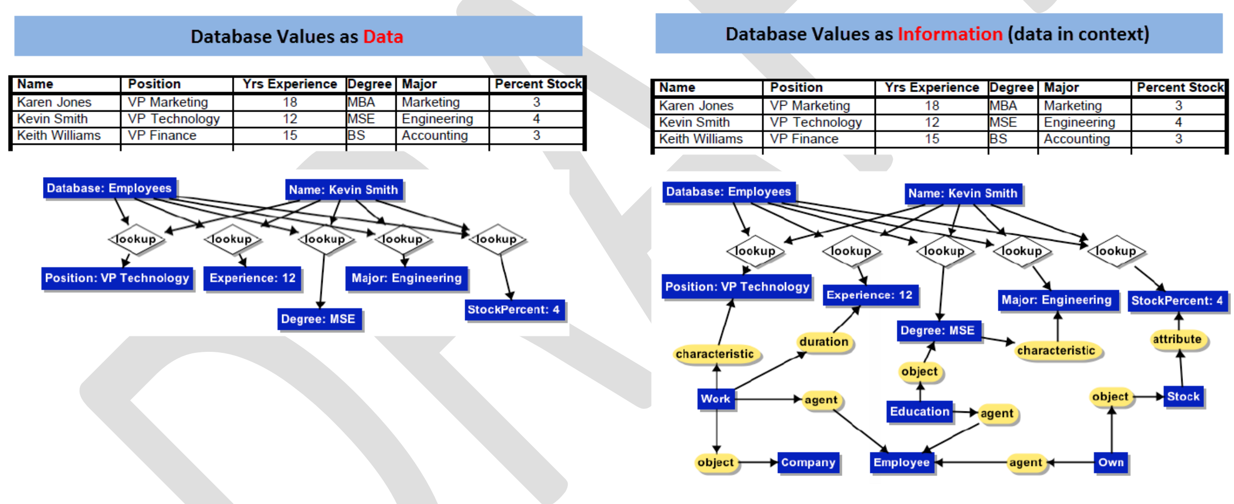
[29] Robert B. Reich, *Saving Capitalism*, Alfred A. Knopf, page 204-206, (see a summary from The Work of Nations here http://www.oss.net/dynamaster/file_archive/040320/e8eb8748abfe77204a145d5fbcc892fb/OSS1993-01-37.pdf#page=6)

[30] 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 subjective interpretation 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)**: 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.

Most people are familiar with and understand how to work with data. Fewer understand the advantages of working with information.

The diagram below is inspired from another similar graphic[31]. I have modified the graphic to make it more understandable. On the left you see the meaning that is commonly represented in a typical database, meaning about the data. On the right you see the same data but with more semantics that put the data into context and therefore turn the data into information. This distinction is important to understand because it is not possible to have a meaningful exchange of data between two business systems using automated processes. Humans would have to get involved to put the data into the proper context. However, it is possible to have a meaningful information exchange between two different business systems is information is exchanged. The meaning, or semantics, of information is richer than the meaning of data.



2.6. Learning about automation from CAD/CAM

Contrasting something new that does not yet exist to something similar that does exist is one way of understanding something. Digital financial reporting has the opportunity to do for the financial report and the financial reporting supply chain what CAD/CAM did for not only the blueprint, but for the entire product design and manufacturing life cycle.

Computer-aided design[32] (CAD) is the use of computer systems to aid in the creation, modification, analysis, or optimization of a design. CAD software is used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing. CAD output is often in

[31] Harry S. Delugach, *Common Logic in Support of Metadata and Ontologies*, Page 7, Retrieved June 24, 2016, http://cl.tamu.edu/docs/cl/Berlin_OpenForum_Delugach.pdf

[32] *Computer-aided Design*, https://en.wikipedia.org/wiki/Computer-aided_design

the form of electronic files for print, machining, or other manufacturing operations. Computer-aided manufacturing[33] (CAM) is the use of software to control machine tools such as numerically controlled machines (NC).

In CAD/CAM software architectural objects have relationships to one another and interact with each other intelligently. For example, a window has a relationship to the wall that contains it. If you move or delete the wall, the window reacts accordingly.

In addition, in CAD/CAM software machine-readable architectural objects maintain dynamic links with construction documents and specifications, resulting in more accurate project deliverables. When someone deletes or modifies a door, the door schedule is automatically updated in your local application's database and perhaps even in the database of the door supplier. Spaces and areas are update automatically when the size of a room is changed and calculations such as total square footage are always up to date. That means, say, that the amount of paint necessary to cover a room or an entire building is always updated. Blueprints can be sent directly to numerically controlled[34] (NC) machines.

Imagine what it would be like to construct a 100 story sky scraper, an iPhone, or a Boeing 777 if all the blueprints were paper-based. Turning this around, digital blueprints enable process and other improvements which allow more sophisticated products to be created effectively and efficiently.

CAD/CAM software is driven by business rules. Likewise, digital financial reports will be driven by business rules. The business rules will be different because the domains are different. The important thing to recognize is that for the same reason CAD/CAM software does work, digital financial reports will also work. All that is necessary are the specific rules for digital financial reports.

3. Examples of Types of Business Rules

The best way to understand what business rules are is to have some examples of business rules. Business rules can be put into general groups by type of rule. Different people group business rules into different groups such as "structural rules" or "behavioral rules". Others break them down into "quality logic" and "business logic".

3.1. General types of business rules

At their essence, business rules articulate information about something or about the relationship between one thing and some other thing. Some examples that can help you better understand exactly what they are:

- **Assertions**: For example asserting that the balance sheet balances or "Assets = Liabilities + Equity".

[33] *Computer-aided Manufacturing*, https://en.wikipedia.org/wiki/Computer-aided_manufacturing

[34] *Numerical control*, https://en.wikipedia.org/wiki/Numerical_control

- **Computations**: For example, calculating things, such as "Total Property, Plant and Equipment = Land + Buildings + Fixtures + IT Equipment + Other Property, Plant, and Equipment".
- **Constraints**: For example, specific behavioral consraints that control when it is appropriate to create, update, or remove information.
- **Process-oriented rules**: For example, the disclosure checklist commonly used to create a financial statement which might have a rule, "If Property, Plant, and Equipment exists, then a Property, Plant and Equipment policies and disclosures must exist."
- **Regulations**: Another type of rule is a regulation which must be complied with, such as "The following is the set of ten things that must be reported if you have Property, Plant and Equipment on your balance sheet: deprecation method by class, useful life by class, amount under capital leases by class . . ." and so on. Many people refer to these as reportability rules.
- **Instructions or documentation**: Rules can document relations or provide instructions, such as "Cash flow types must be either operating, financing, or investing."
- **Relations**: How things can be related, such as whole-part relations. For example, how the business segments of an economic entity are related.

3.2. *Categories of business rules from the XBRL perspective*

Business rules can be categorized. The Business Rules Group provides a good summary of business rule categories. But sometimes how terms are used differ between groups using such terms. Below I have provided a set of business rule categories inspired by the Business Rules Group categories which is then reconciled to XBRL terminology. This is my FIRST PASS at reconciling these terms. I am soliciting feedback from others to get this correct. Please consider this a DRAFT at this point!

- **Definition of business terms**: The very definition of business terms is a category of business rule. Each term is a rule. In XBRL, the report elements defined in an XBRL taxonomy schema is how business terms are defined. Terms are essentially identifiers. In XBRL, terms are grouped into one of the following categories of terms: Network, Hypercube (a.k.a. Table), Dimension (a.k.a Axis), Member, Primary Items (a.k.a. Line Items), Primary Item (a.k.a. a concrete Concept), and Abstract(a.k.a an abstract Concept or Primary Item). Business professionals have to go through a process of naming things that exist in reality and giving them names as contrast to providing additional preferred labels for names that already exist. These terms describes how business professionals think and talk about real world notions, ideas, and other such phenomenon. The definition of terms in the past have been documented in the form of human-readable glossaries. We now make these terms human-readable and machine-readable by defining them in XBRL taxonomy schemas. Information technology professionals sometimes define terms in the form of an entity/relationship model.
- **Structural assertions**: This term appears to describe two types of structures in XBRL: what XBRL calls a "fact" and what XBRL calls "relations" (presentation, calculation, definition, XBRL formula). Structural assertions can be documented as

natural language sentences or described graphically has hierarchies as relationships, qualities, and other such structures. There are two important distinct types of structures in XBRL

  - **XBRL Fact**: A fact in XBRL is something that is reported within an XBRL instance. A fact is a structure comprised of other structures generally defined in the form of terms in an XBRL taxonomy schema but there are a few things defined in the XBRL instance itself (entity identifier, period, XBRL footnotes). So, a fact is a hard-coded structural assertion defined by the XBRL technical specification. A fact has an aspect model. This is the same as what I call the multidimensional model of XBRL.
  - **Other XBRL relations**: This category of structural assertions includes all other relations definable using XBRL including presentation, calculation, definition, and XBRL formula relations.

- **Action assertions**: Action assertions constrain or influence behavior in some way. Action assertions cause things to happen or prevent things from happening. They can also be used to make suggestions. XBRL Formula provides for existence assertions, consistency assertions, and value assertions. (I think a value assertion is a derivation.)
- **Derivations**: A derivation is a mathematical algorithm or a logical inference (induction or deduction) that is used to derive, or what I have called impute, other structural relations (i.e. XBRL facts or other relations). Derivations create new knowledge based on existing knowledge. XBRL Formula has a mechanism for creating new facts.

A bit of clarification is helpful to make sure all of the above is clear. The notion of derivations might not be familiar to some people or you might be familiar with it in different terms. Here is some clarifying information that distinguishes between explicitly provided facts and derived facts (see here for similar information from the Business Rules Group):

- **Base Fact**: a base fact is a fact that has been explicitly reported in a financial report. For example, if you report the fact "us-gaap:Assets" for a specific economic entity for a specific period.
- **Derived Fact**: a derived fact is a fact whose value is created by an inference or mathematical computation. For example, if the base facts "us-gaap:Assets" and "us-gaap:AssetsCurrent" are reported then the fact "us-gaap:AssetsNoncurrent" can be derived because of two pieces of information: (1) the values of us-gaap:Assets and us-gaap:AssetsCurrent are known and (2) the business rule "Assets = Current assets + Noncurrent assets" is known; so deductive reasoning can be used to obtain the derived fact "us-gaap:AssetsNoncurrent'.
- **Derivation**: a derivation is an algorithm used to infer or compute a Derived Fact. (i.e. a business rule). In the derived fact example above, the derivation is "Assets = Current assets + Noncurrent assets". There are two types of derivations:
  - **Logical inference**: a logical inference is a Derivation that produces a Derived Fact using logical induction (from particulars) or deduction (from general principles).

- **Mathematical inference**: a Derivation that produces a Derived Fact according to a specified mathematical algorithm.

3.3. Logical layers of business rules

Business rules can be organized into convenient logical layers that interact with one another. These layers contribute to helping business professionals manage, use, and otherwise organize and interact with business rules.

- **Flow logic:** (sequence, process or flow)
  - **Procedural logic** – model sequence, loop, or iterative procedures.
  - **Flow logic** – fully automated sequence of operations, actions, tasks, decisions, rules.
    - **Workflow logic** – type of flow logic, semi-automated or manual processes that need an action to be taken from outside the system by another system or human.
- **Information assertions:** (compliance, quality, consistency, completeness, accuracy)
  - **Business terminology logic** – definition of business concepts
  - **Validation logic** – validate action assertions.
    - Action assertions
    - Structural assertions
  - **Decision logic** – type of validation logic, handles execution que and conflict resolution
    - Execution logic
    - Conflict resolution logic
  - **Derivation logic** – deviations which derives new facts using existing facts, rules, and logical or mathematical reasoning
    - Logical inference
    - Mathematical inference

[CSH: This is one of the better explanations of these different layers that I have run across, http://wiki.flexrule.com/index.php?title=Logic]

4. Meaningful Exchange of Information

The only way a meaningful exchange of information can occur is the *prior agreement* as to:

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

One key phrase from above is "prior agreement".

4.1. Consider a scenario from financial reporting

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.

4.2. The Helsinki principles

ISO TR 9007:1987 ("Helsinki principles") state[35]:

- 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

4.3. Specific fundamental strengths of computers

Computers have specific fundamental strengths[36]:

- **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 anywhere on the planet in real time via the internet, simultaneously to all individuals.

4.4. Major obstacles to harnessing the power of computers

In order to harness the power of computers, there are major obstacles which must be overcome[37].

---

[35] Harry S. Delugach, *Common Logic in Support of Metadata and Ontologies*, retrieved August 3, 2016, http://cl.tamu.edu/docs/cl/Berlin_OpenForum_Delugach.pdf#page=5

[36] 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

[37] 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

- **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.
- **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.

4.5. Understanding what computers cannot do

Understanding what computers cannot do is just as important as understanding what they can do; computers cannot replicate:

- Intuition
- Creativity
- Innovation
- Improvise
- 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

4.6. Setting the right expectations

Business professionals need to understand what computers can and cannot do. Computers cannot perform magic. Computers fundamentally follow the rules of mathematics which follow the rules of formal logic. It really is that straight forward. Computers cannot 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."

Computers can try and mimic the sorts of behaviors mentioned above using probabilistic reasoning. But this sort of technology is complicated, untested, unproven, and difficult and expensive to implement.

The point is that professional accountants need to be able to differentiate what is easy, safe, and reliable from what is hard, unsafe, and may not be reliable.

5. Intelligent software agents assisting humans

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[38].

Expert systems is a branch of artificial intelligence. An intelligent agent is software that assists people and acts on their behalf. Intelligent agents work by allowing people to:

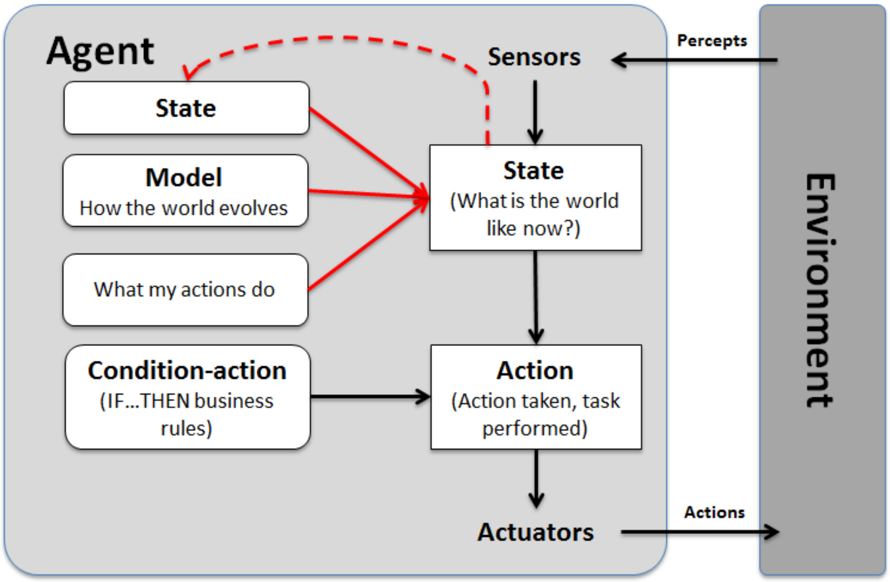
- delegate work that they could have done to the agent software,
- perform repetitive tasks,
- remember things you forgot,
- intelligently find, filter and summarize complex information,
- customize information to your preferences,
- learn from you and even make recommendations to you.

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.

The main difference between a software agent and an ordinary program is that a software agent is autonomous; that is, it must operate without direct intervention of humans or others. There are many different types of intelligent software agents[39].

---

[38] *Introduction to Artificial Intelligence Terminology*, http://xbrl.squarespace.com/journal/2016/7/21/introduction-to-artificial-intelligence-terminology.html



Intelligent agents can perform sophisticated work. A rational agent is one that acts so as to achieve the best outcome or, when there's uncertainty, the best expected outcome. Rationality as used here refers to following the rules of logical reasoning, making correct inferences, and selecting the appropriate action that will lead to achieving the desired goal.

Machine-readable business rules are key to creating intelligent software agents that provide the functionality within an expert system.

5.1. Intelligent agent sophistication differs

Intelligent software agents have different levels of sophistication. Some agents are sophisticated and can learn new information; others are less sophisticated and non-learning.

Important distinctions between agents:

- **Rational agent**: A rational agent is one that acts so as to achieve the best outcome or, when there's uncertainty, the best expected outcome. Rationality as used here refers to following the rules of logical reasoning, making correct inferences, and selecting the appropriate action that will lead to achieving the desired goal.
- **Autonomous agent**: An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.

Categories of agents according to their functionality:

- **Reactive agent**: A reactive agent is capable of maintaining an ongoing interaction with the environment and responding in a timely fashion to changes that occur in it.
- **Pro-active agent**: A pro-active agent is capable of taking the initiative; not driven solely by events, but capable of generating goals and acting rationally to achieve them.

[39] *Introduction to Intelligent Agents for Business Professionals (DRAFT)*, http://xbrlsite.azurewebsites.net/2016/Library/02_IntroducingIntelligentAgents.pdf

- **Deliberative agent**: A deliberative agent symbolically represents knowledge and makes use of mental notions such as beliefs, intentions, desires, choices and so on. (This is implemented using a belief-desire-intension model.)
- **Hybrid agent**: A hybrid agent is one that mixes some of all the different architectures.

5.2. Benefits offered by expert systems

In the future, the accounting and reporting rules will exist in both human-readable and machine-readable form and will drive the expert systems and intelligent software agents which professional accountants use to create financial reports. Benefits from the use of expert systems and intelligent software agents include:

- **Automation**: elimination of routine, boring, repetitive, mundane, mechanical tasks that can be automated
- **Consistency**: computers are good at performing repetitive, mechanical tasks without variation 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**: such 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 single 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

Critical to understanding the sorts of tasks that expert systems will be capable of performing and should not or will never be able to perform takes the understanding of a domain professional. While computer based expert systems can effectively automate some work, this does not imply that these systems will automate all work or replace humans. They simply won't because they cannot. Computers are dumb beasts. There is a difference between subjectivity and objectivity; there is a difference between a mechanical task and a task requiring professional judgement. Professional accountants need to understand the difference[40].

6. Understanding the Business Rules Engine

Rather than a computer programmer writing a bunch of IF…THEN sequential computer code to enforce business rules as had been the case in the past, today business rules enforced differently. There are two pieces to enforcing business rules:

1. Writing the business rules
2. Using a business rules engine to process business rules

This separation of concerns produces many significant benefits. Business rules can be repurposed for many things, flexibility, easier maintenance, and so forth. So what is a business rules engine? What type of business rules engine do you need?

- **Semantic reasoner**[41]: A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with.
- **Inference engine[42]**: The inference engine applied logical rules to the knowledge base and deduced new knowledge.
- **Business rules engine[43]**: A business rules engine is a software system that executes one or more business rules in a runtime production environment. Some problem solving logic[44] is used by the business rules processor.

6.1. Business rules engine terminology

This blog post[45] about semantic reasoners provides the following definition which points out that there are many different terms that refer what might be, or might not be, the same sort of thing:

> A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to **infer logical consequences from a set of asserted facts or axioms**. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. The inference rules are

---

40 *Knowledge Engineering Basics for Professional Accountants*, http://www.xbrlsite.com/2016/Prototype/00_KnowledgeEngineeringBasicsForAccountingProfessionals.pdf

41 Wikipedia, *Semantic Reasoner*, retrieved August 2, 2016, https://en.wikipedia.org/wiki/Semantic_reasoner

42 Wikipedia, *Inference Engine*, retrieved August 2, 2016, https://en.wikipedia.org/wiki/Inference_engine

43 Wikipedia, *Business rules engine*, retrieved August 2, 2016, https://en.wikipedia.org/wiki/Business_rules_engine

44 *Comprehensive Introduction to the Notion of Problem Solving Logic for Professional Accountants*, http://xbrlsite.azurewebsites.net/2016/Library/ComprehensiveIntroductionToNotionOfProblemSolvingLogicForProfessionalAccountants.pdf

45 Hello Semantic Web, *Semantic Reasoners*, retrieved August 2, 2016, http://hellosemanticweb.blogspot.com/2011/04/semantic-reasoners.html#axzz2URUuQy00

commonly specified by means of an ontology language, and often a description language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining and backward chaining.

Breaking these terms out, these are the important pieces to understand:

- **Machine or engine**: an apparatus using or applying mechanical power and having several parts, each with a *definite function* and together performing a *particular task*.
- **Inference**: a business rules engine either does, or does not, include the ability to infer new information using existing information and logical rules.
- **First-order predicate logic**: Formal logic was consciously broken into two groups: first-order logic and higher-order logic. 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.
- **Logical catastrophes[46]**: There are things that someone can do under first-order predicate logic that cause what I refer to as logical catastrophes or system failure points. For example, inadvertently putting a system into an infinite loop from which it cannot escape is a logical catastrophe. The possibility of all such catastrophes must be eliminated from business rules systems.
- **Horn Clauses**: Horn Clauses is a safe subset of first-order predicate logic. PROLOG is limited by Horn Clauses. However, PROLOG still has specific cases which causes systems to inadvertently break. As such, DATALOG was created to create a safer set of first-order predicate logic.
- **Multidimensional model**: Transaction processing systems (OLTP[47]) and analytical systems (OLAP[48]) are used for different things. OLAP leverages a multidimensional model which makes querying information more flexible and efficient. Business rules engines might need to have an inherent understanding of the multidimensional model[49]. XBRL provides a global standard dimensional model[50]. RDF also provides a global standard dimensional model[51].
- **Mathematics**: Business rules engines or semantic processors need to have an inherent understanding of mathematics to the extent that mathematics is used within business rules.

---

46 *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

47 Wikipedia, *Online Transaction Processing*, Retrieved August 3, 2016, https://en.wikipedia.org/wiki/Online_transaction_processing

48 Wikipedia, *Online Analytical Processing*, https://en.wikipedia.org/wiki/Online_analytical_processing

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

50 *XBRL Dimensions 1.0*, https://specifications.xbrl.org/work-product-index-group-dimensions-dimensions.html

51 *The RDF Data Cube Vocabulary*, http://www.w3.org/TR/vocab-data-cube/ also see https://dvcs.w3.org/hg/gld/raw-file/29a3dd6dc12c/data-cube/index.html

- **Standard or proprietary**: There are standard approaches to implementing rules engines and there are proprietary approaches and tools. Some business rules engines might be open source.
- **XBRL Formula processor**: An XBRL formula processor is a business rules engine. There is a global standard specification[52] for expressing business rules using XBRL.

6.2. Business rules engines

So, what is the best business rules engine? There are different categories of semantic reasoners:

- **OWL reasoners**[53]: Business rules engines, or reasoners, that work with OWL.
- **XBRL Formula processors**: Business rules engines that work with XBRL Formula. For example Arelle (http://arelle.org/) is XBRL software that includes an XBRL Formula processor.
- **Proprietary rules engines**: Business rules processors can use global standard and/or proprietary information and rule formats. Here are a few such business rules engines:
    - FlexRule: http://www.flexrule.com/
    - Fluent Editor: http://www.cognitum.eu/semantics/FluentEditor/
    - Ergo Suite: http://coherentknowledge.com/product-overview-ergo-suite-platform/
- **SBVR processors**[54]: SBVR processors are conformant to the Semantics of Business Vocabulary and Business Rules (SBVR) OMG standard.

6.3. Problem solving logic

Computers work using the rules of mathematics. Mathematics works using the rules of logic. A problem solving logic[55] is how a computer reasons.

To understand the notion of problem solving logic one first needs to understand the notion of logic and how logic can be applied to solving a problem. This section is dedicated to setting your perspective. The section provides specific definitions, deconstructing the pieces so that we can subsequently put the pieces back together.

The XBRL technical syntax is a global standard logic for representing knowledge. While much of the logic such as XBRL elements, relations between elements, mathematical relations between concepts and facts (XBRL calculation relations and XBRL Formula relations), dimensional relationships between concepts and facts, and other such relations (expressible using XBRL definition relations); not all such relation logic is standard.

---

52 *XBRL Formula Specification*, https://specifications.xbrl.org/work-product-index-formula-formula-1.0.html

53 W3C, *OWL Reasoners*, https://www.w3.org/2001/sw/wiki/OWL/Implementations

54 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/

55 *Comprehensive Introduction to the Notion of Problem Solving Logic for Professional Accountants*, http://xbrlsite.azurewebsites.net/2016/Library/ComprehensiveIntroductionToNotionOfProblemSolvingLogicForProfessionalAccountants.pdf

XBRL Formula processers have specific deficiencies in their processing capabilities[56]. To overcome these deficiencies, the following capabilities must exist or need to be added to XBRL Formula Processors:

- Support **normal global standard functionality** that high-quality XBRL Formula processors support (i.e. Arelle, UBmatrix/RR Donnelley, Fujitsu, Reporting Standards, etc.)
- **Support inference** (i.e. deriving new facts from existing facts using logic, what inference engines do)
- Improved support validation and use of **structural relations** (i.e. XBRL Taxonomy functions; this was consciously left out of the XBRL Formula specification in order to focus on XBRL instance functionality)
- Support **forward chaining** and possibly also backward chaining in the future (i.e. chaining was also proposed but was left out of the XBRL Formula specification)
- Support a **maximum amount of Rulelog logic** which is safely implementable and is consistent with ISO/IEC Common Logic and OMG Semantics of Business Vocabulary and Business Rules
- **Additional XBRL definition arcroles** that are necessary to articulate the Rulelog logic, preferably these XBRL definition relation arcroles would end up in the XBRL International Link Role Registry and be supported consistently by all XBRL Formula processors (i.e. these general arcroles, and these financial disclosure related arcroles; this human readable information is helpful to understand the arcroles)

While added functionality might not be global standard functionality, the functionality is necessary to prove the logic of US GAAP based financial reporting or IFRS based financial reporting. US GAAP and IFRS semantics are relatively clear. What is not clear to some business professionals is how to represent that meaning using the XBRL global standard. Proprietary techniques for applying XBRL can be used to fill any gap. However, the logical rules used by any proprietary techniques should follow the logic of Common Logic, SBVR, and RuleLog.

6.4. Differentiating forward and backward chaining

I'm still trying to figure out the best processing approach for the fundamental accounting concept relations and other business rules to best take advantage of the power XBRL has to express business rules. One part of that question is whether a forward chaining or backward chaining approach is best.

The blog post Forward and Backward Chaining: Part 2, by Charles Forgy, PhD, provided the best answer to that question. In a nutshell, here is Dr. Forgy's answer: use both if possible:

In conclusion, forward and backward chaining systems both use subgoals to control the operation of a rule base. Pure forward chaining systems are more powerful than pure backward chaining systems, but pure forward chaining systems require the developer to write all the subgoaling rules. Modern forward chaining systems such as the RulesPower

[56] Specific Deficiencies in Capabilities of Existing XBRL Formula Processors, http://xbrl.squarespace.com/journal/2016/9/26/specific-deficiencies-in-capabilities-of-existing-xbrl-formu.html

system integrate automatic backward chaining with forward chaining. These systems combine the best features of both forward and backward chaining.

Dr. Forgy distills the difference between forward and backward chaining down to the following two salient points:

- Rule firing:
    - 1.Forward-chaining systems fire rules whenever the rules' If parts are satisfied.
    - 2.Backward-chaining systems attempt to fire rules only when those rules can potentially satisfy a goal.
- Subgoal creation:
    - 1.Backward-chaining systems automatically create new subgoals when more information is needed to determine whether a given rule is satisfied.
    - 2.Forward-chaining systems do not automatically create subgoals.

Dr. Forgy explains why you would want to use forward-chaining when backward-chaining automatically creates new subgoals but forward-chaining does not:

You might ask why you would want to use a forward chaining system if you have to write rules to manage subgoals. After all, backward chaining systems automatically manage the subgoals. There answer is very simple: Backward chaining systems are more limited than forward chaining systems. There are many kinds of tasks that can be handled easily with a forward chaining system that are either difficult or impossible with a backward chaining system. Backward chaining systems are good for diagnostic and classification tasks, but they are not good for planning, design, process monitoring, and quite a few other tasks. Forward chaining systems can handle all these tasks.

So there seems to be a tradeoff. You can use forward-chaining and satisfy all your needs, but the price you pay is having to create subgoals.

Alternatively, you could use a system that supports forward-chaining which automatically switches to a backward-chaining mode when needed.

If you want to understand more details about forward an backward chaining, Dr. Forgy first part in this series, Forward and Backward Chaining Part 1, is worth reading. This Wikipedia article on business rules engine types is also useful.

But man, why can't this be easy! Note this statement in the last paragraph on the Wikipedia article about business rule engine types:

A fourth class of rules engine might be called a deterministic engine. These rules engines may forgo both forward chaining and backward chaining, and instead utilize domain-specific language approaches to better describe policy. This approach is often easier to implement and maintain, and provides performance advantages over forward or backward chaining systems.

When a forward chaining approach is used, the rules are traversed from the problem to the solution to the problem.

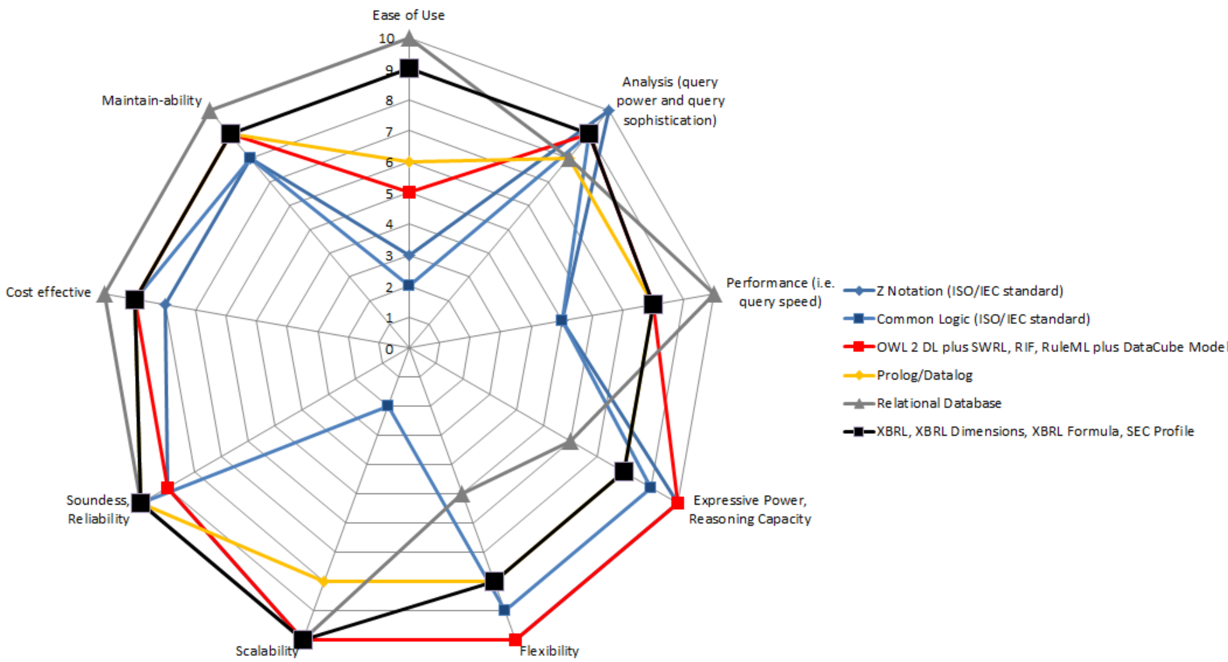
- If A then B
- If B then C
- If C then D

A chain traversed from a hypothesis back to the facts that support the hypothesis is a backward chain.

- If D then C
- If C then B
- If B then A

7. Achieving Balance and Equilibrium

Every solution is a basket of pros and cons. The best solution depends on the requirements of the problem that is being solved. And so understanding your requirements correctly is paramount when it comes to being sure you select the right approach to solving any problem.



The list of requirements that I have come up with include the following:

- **Ease of use**: How easy is the system for business professionals to use?
- **Analysis (query power and query sophistication)**: When you are gathering information to make use of that information, how sophisticated are the queries?
- **Performance (query speed)**: What is the raw speed at which information is provided for use?
- **Expressive power, reasoning capacity**: What is the expressive power of the business rules? There is a direct correlation between expressive power and the reasoning capacity the system can offer.
- **Flexibility**: Is the system flexible enough to meet your needs?

- **Scalability**: Will the system scale if your needs grow?
- **Soundness, reliability**: Is the system reliable and sound?
- **Cost effectiveness**: What is the cost of the system relative to what the system provides?
- **Maintainability**: How hard is it to maintain the system?