1. Computational Professional Services

"The difficulty lies not so much in developing new ideas as in escaping from old ones." John Maynard Keynes

"If I had asked people what they wanted, they would have said faster horses." Henry Ford

The purpose of this section is to explain the notion of computational professional services. The term, computational professional services, is inspired by the excellent paper by Michael Genesereth of Stanford University's Center for Legal Informatics, *Computational Law: The Cop in the Backseat*[1]. In that paper, the author defines *computational law* as:

> "Computational Law is that branch of legal informatics concerned with the codification of regulations in precise, computable form. From a pragmatic perspective, Computational Law is important as the basis for computer systems capable of doing useful legal calculations, such as compliance checking, legal planning, regulatory analysis, and so forth."

My proposition is that these same ideas can be applied to accounting, reporting, auditing, and analysis. In fact, that paper provides an example from accounting to help the reader understand the notion of computational law.

> "Intuit's Turbotax is a simple example of a rudimentary Computational Law system. Millions use it each year to prepare their tax forms. Based on values supplied by its user, it automatically computes the user's tax obligations and fills in the appropriate tax forms. If asked, it can supply explanations for its results in the form of references to the relevant portions of the tax code."

I have briefly mentioned computational law[2], computational audit[3], computational economics[4], and computational regulation[5] on my blog. I would include each within the larger bucket of **computational professional services**. There are likely other business domains that might fit. But what exactly does "computational professional services" mean and how do you get it to actually work.

That is what is covered within this section. Computational professional services is exemplified by what the Data Coalition calls "smart regulation[6]" or what others call "algorithmic regulation[7]". Others use the term "rules as code". Deloitte use the term "the finance factory[8]". Others use different terms. For example, Craig Lewis, Chief Economist and Director of the Division of Risk, Strategy, and Financial Innovation

[1] Michael Genesereth, Stanford University's Center for Legal Informatics, *Computational Law: The Cop in the Backseat*, http://logic.stanford.edu/complaw/complaw.html

[2] Computational Law, http://xbrl.squarespace.com/journal/2020/8/24/computational-law.html

[3] Computational Audit, http://xbrl.squarespace.com/journal/2020/8/25/computational-audit.html

[4] Computational Economics, http://xbrl.squarespace.com/journal/2020/8/31/computational-economics.html

[5] Computational Regulation, http://xbrl.squarespace.com/journal/2020/9/1/computational-regulation.html

[6] Smart Regulation Graphic Show the Big Picture, http://xbrl.squarespace.com/journal/2012/11/12/smart-regulation-graphic-shows-the-big-picture.html

[7] Tim O'Reilly, Open Data and Algorithmic Regulation, https://beyondtransparency.org/chapters/part-5/open-data-and-algorithmic-regulation/

[8] Deloitte's Vision: The Finance Factory, http://xbrl.squarespace.com/journal/2019/2/20/deloittes-vision-the-finance-factory.html

(RiskFin) at the SEC used the term "robo cop". We will standardize on the term computational professional services. "Continuous auditing" and "continuous reporting" are other terms. "Finance transformation[9]" is yet another.

1.1. Computational

Let me first start by explaining what I mean by computational. Per Wikipedia, a **computation**[10] is defined as:

> "A computation is any type of calculation that includes both arithmetical and non-arithmetical steps and which follows a well-defined model (e.g. an algorithm)."

The key takeaway here is that by computation we don't just mean math. Computation means steps that can be followed by a computer.

Per Wikipedia (paraphrasing), **algorithm**[11] is defined as:

> "An algorithm is a finite sequence of well-defined instructions, typically to solve a class of problems or to perform a computation. Algorithms are always unambiguous and are used as *specifications* for performing calculations, data processing, automated reasoning, and other tasks."

Computational logic[12] is the use of logic to perform or reason about computation. Logic is used to define the steps and tasks. Logic is a set of principles that forms a framework for correct reasoning. We will dive into logic a little later.

Humans are very capable of carrying out steps and performing tasks. Mechanical devices can also be created to carry out steps and perform tasks, for example a vending machine is such a device. A calculator is also such a device. Computers via the software programs they run can likewise carry out steps and perform tasks.

Can computers perform all work? The answer is no. For example, computers cannot exercise professional judgment. But computers can help out with many routine, repetitious, monotonous, mechanical, boring, grueling tasks and processes of accounting, reporting, auditing, and analysis.

1.2. Professional Services

For completeness I want to provide a definition for **professional services**. Again, Wikipedia offers a good definition[13]:

> "Professional services are occupations in the service sector requiring special training in the arts or sciences. Some professional services require holding professional degrees and licenses and they also require specific skills such as architects, accountants, engineers, doctors, lawyers and teachers. Other professional services involve providing specialist business support to businesses of all sizes and in all sectors; this can include tax advice,

[9] Finance Transformation is a Thing, http://xbrl.squarespace.com/journal/2020/1/30/finance-transformation-is-a-thing.html

[10] Wikipedia, *Computation*, https://en.wikipedia.org/wiki/Computation

[11] Wikipedia, *Algorithm*, https://en.wikipedia.org/wiki/Algorithm

[12] Wikipedia, *Computational Logic*, https://en.wikipedia.org/wiki/Computational_logic

[13] Wikipedia, *Professional Services*, https://en.wikipedia.org/wiki/Professional_services

> supporting a company with accounting, IT services or providing management advice."

While what the different domains included within professional services do, there are many patterns that all domains have in common. One pattern that crosses all domains of professional services is the use of abstract symbols specific to that domain and rearranging those symbols. Let me explain.

1.3. Rearranging Abstract Symbols

Accounting, reporting, auditing, and analysis have a lot to do with professional accountants "rearranging abstract symbols". This is what I mean. In his book *Saving Capitalism*[14], Robert Reich describes three categories that all modern work/job tasks fit into:

- **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."

Shelly Palmer breaks work tasks down in another way[15]. He points out that almost every human job requires us to perform some combination of the following four basic types of tasks:

- Manual repetitive (predictable)
- Manual nonrepetitive (not predictable)
- Cognitive repetitive (predictable)
- Cognitive nonrepetitive (not predictable)

Manual involves using one's hands or physical action to perform work. ***Cognitive*** involves using one's brain or mental action or a mental process of acquiring knowledge/understanding through thought, experience, use of the senses, or intuition.

Predictable manual or cognitive tasks can be automated. ***Unpredictable*** manual or cognitive tasks cannot be automated.

---

[14] Robert Reich, *Saving Capitalism*, page 204-206, https://www.amazon.com/Saving-Capitalism-Many-Not-Few/dp/0345806220

[15] Shelly Palmer, *The 5 Jobs Robots Will Take Last*, https://www.linkedin.com/pulse/5-jobs-robots-take-last-shelly-palmer

Palmer gives the example of an assembly line worker that performs mostly manual repetitive tasks which, depending on complexity and a cost/benefit analysis, can be automated. On the other hand, a CEO of a major multinational conglomerate performs mostly cognitive nonrepetitive tasks which are much harder to automate.

Many cognitive repetitive tasks in accounting, reporting, auditing, and analysis are related to symbolic-analytic services are candidates for automation.

Computational professional services is about using computers to help professional accountants perform cognitive repetitive tasks related to the rearrangement of abstract symbols in accounting, reporting, audit, analysis, law, or other professional services domains. Work will be performed by humans augmented by machines.

1.4. Symbolic Systems

A **symbolic system**[16] is essentially a system built with symbols such as natural language, programming languages, mathematics, or formal logic. An interesting thing about such systems is that symbolic systems are understandable by both *humans* and by *computers*.

Interestingly, Stanford University has a popular undergraduate and graduate degree offering in symbolic systems[17].

Using financial accounting and reporting as an example; there is nothing natural about double entry bookkeeping[18], the accounting equation[19], the accounting ledger[20], the general purpose financial report[21], the spreadsheet[22], or the many other artifacts of accounting, reporting, auditing, and analysis[23]. All of these are tools and symbolic systems invented by man in order to help enable commerce. Arguably, without these accounting tools enterprises such as global multinational corporations would never be able to exist.

Today, in the fourth industrial era[24], in this information age[25], it is possible to improve upon many of the tasks and processes involved with accounting, reporting, auditing, and analysis of financial information; improving quality, reducing cost, and/or performing this work faster. The fourth industrial revolution allows for us to use computers, the internet, digital distributed ledgers, structured information, and artificial intelligence to perform these important tasks and processes that keep global

---

[16] Symbolic Systems, http://xbrl.squarespace.com/journal/2020/8/26/symbolic-systems.html

[17] Stanford University, Symbolic Systems Program, https://symsys.stanford.edu/about/span-dig-deep-solve-complex-problems

[18] The Math of Double Entry Bookkeeping, http://xbrl.squarespace.com/journal/2019/11/4/the-mathematics-of-double-entry-bookkeeping.html

[19] Accounting Equation, http://xbrlsite.azurewebsites.net/2020/master/ae/index.html

[20] Charles Hoffman and Andrew Noble, Introduction to the Fact Ledger, http://xbrlsite.azurewebsites.net/2018/Library/IntroductionToTheFactLedger.pdf

[21] Need for an Alternative to the General Purpose Financial Report, http://xbrl.squarespace.com/journal/2015/10/28/need-for-digital-alternative-to-general-purpose-financial-st.html

[22] Understanding the Semantic Spreadsheet, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingSemanticSpreadsheets.pdf

[23] Essence of Accounting, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf

[24] Adapting to Changes Caused by Fourth Industrial Revolution, http://xbrl.squarespace.com/journal/2019/8/4/adapting-to-changes-caused-by-the-fourth-industrial-revoluti.html

[25] Wikipedia, *Information Age*, https://en.wikipedia.org/wiki/Information_Age

commerce moving in better and more efficient ways. Things like "ten finger integration" (i.e. manual rekeying of information) can be reduced or perhaps even eliminated altogether. This will allow professional accountants to focus on more important and higher value added tasks that cannot be performed by computers.

1.5. Computers

To understand how to get a computer to do work, it is important to understand the strengths of computers and the obstacles that get in the way which we will highlight now along with a few other important details.

1.6. Strengths of Computers

Computers seem to perform magic. How computers do what they do tends to be a mystery to many people. But computers are simple machines that follow very specific instructions. The strengths of computers can be summarised as follows. Computers can:

- store information
- retrieve information
- process stored information
- make information accessible to individuals or other machines or software

1.7. Obstacles to Using Computers

The accounting profession is yet to fully leverage the strengths of computers mainly due to the following general obstacles that tend to get in the way:

- accountants use different terminologies to refer to exactly the same thing
- accountants differ in their understanding and interpretation of accounting standards
- accountants don't understand technologies' limitations
- IT professionals use different technology stacks and languages to achieve the same result
- IT and business professionals have an oversimplified view of accounting

Fundamentally, computers are "dumb beasts" that have to be led by the hand like you lead a baby.

1.8. Understanding what Computers Cannot Do

Key to understanding what work computers are capable of performing is understanding of 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
- 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, it is generally not practical.

1.9. Knowledge-based Systems

The better the capability of a system to represent knowledge[26], the better the ability for a software application to read and process that knowledge and perform useful work for the user of the system. The following are approaches to representing knowledge:

- A **dictionary** documents a simple flat inventory of ***terms*** with *no associations* between terms so there is *no hierarchy of information*.
- A **thesaurus** documents ***terms*** plus some of the ***associations*** between *broader* and *narrower* (i.e. type-subtype) terms, that tends to form *one hierarchy of information*.
- A **taxonomy** documents ***terms*** plus some of the ***associations*** between more *general* and more *specialized* terms (i.e. type-subtype), like a thesaurus, and tends to form *one hierarchy of information*.
- An **ontology** documents ***terms*** plus many different types of ***associations*** including broader/narrower AND general/special (i.e. type-subtype) PLUS whole/part (i.e. has-a, part-of) associations, many other sorts of associations which can be used to create many different ***structures*** which form a conceptual ***model***.
- A **logical theory** documents ***terms***, many different types of ***associations*** between terms (i.e. type-subtype), many different types of ***structures*** (i.e. has-a, part-of), which forms a ***model***; plus in addition documents *mathematical* ***rules***, and other *constraints* (another type of ***rule***).

I have created a logical theory that describes the mechanical aspects and dynamics of a financial report[27]. But to get a knowledge-based system to work, you have to

---

[26] YouTube.com, *Taxonomies, Ontologies, and Knowledge Graphs, Oh My!*, https://youtu.be/3KA__Dcb8Ns

[27] *Logical Theory Describing Financial Reports*, http://www.xbrlsite.com/2020/Theory/LogicalTheoryDescribingFinancialReport.pdf

put knowledge into that system. So, what exactly constitutes a knowledge-based system?

1.10. Components of a Knowledge-based System

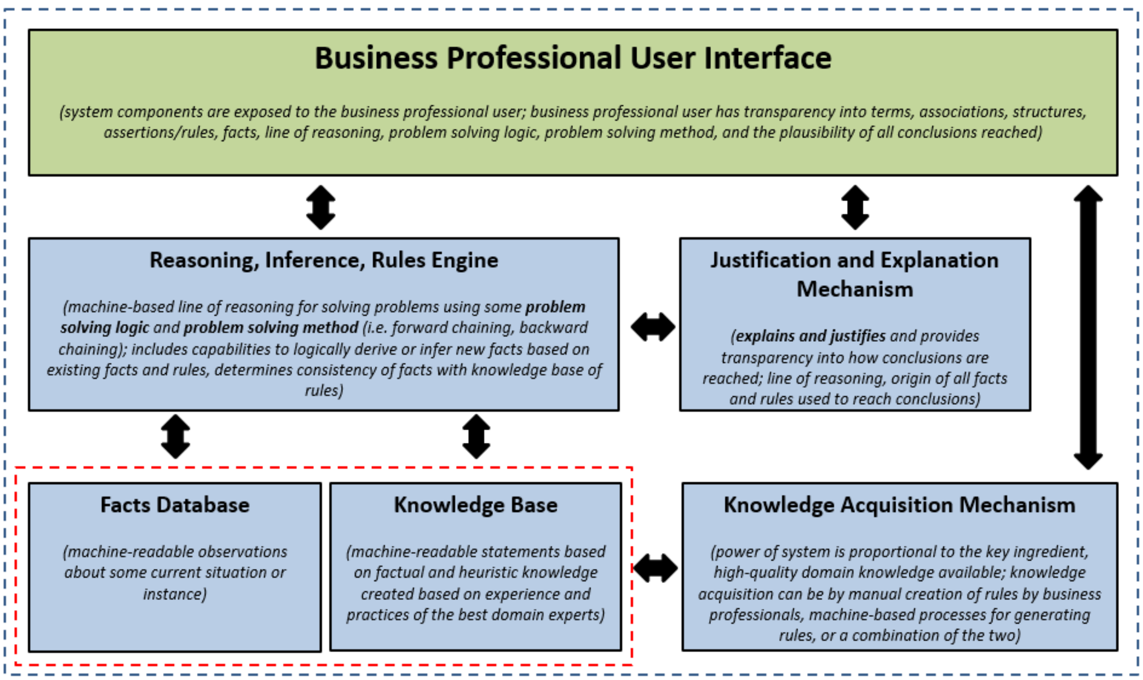
Wikipedia defines a **knowledge-based** system as follows:

> "A knowledge-based system is a computer program that reasons and uses a knowledge base to solve complex problems."

Information is acquired from skilled, knowledgeable professionals. This information is stored in a knowledge base and a fact database. The system applies problem solving logic using a problem-solving method. The knowledge-based system supplies an explanation and justification mechanism to help users understand the line of reasoning used to reach conclusions. The system then presents that information back to the user.

Nothing is a "black box". The origin of information used to reach conclusions is always apparent to the users of the application.

The following graphic provides a summary of the components of a knowledge-based system:



The following briefly describes each of those components:

- **Knowledge acquisition mechanism**: Somehow knowledge needs to be acquired and put into the knowledge-based system.
- **Knowledge base**: Somehow the knowledge acquired needs to be stored in machine-readable form such that it can be used by the system.
- **Fact database**: Similarly, facts need to be stored in machine-readable form such that they can be used by the system.

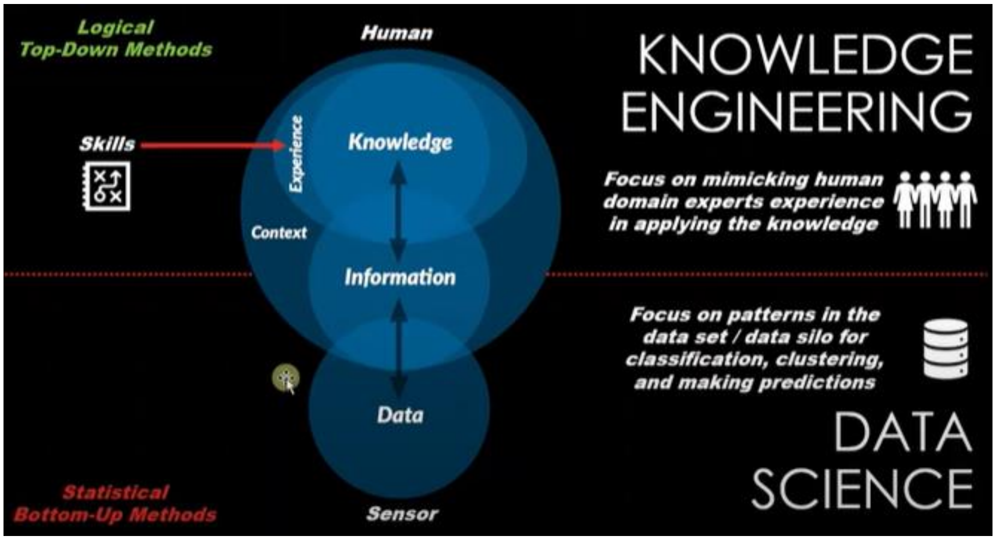
- **Reasoning, inference, rules engine**: Some rules engine is necessary to process the knowledge and facts. Deductive reasoning is essential; inductive reasoning is a nice-to-have.
- **Justification and explanation mechanism**: Nothing in the system should be a black box. Users of the system must be able to understand the origin of information (providence) and there needs to be an audit trail to understand every decision made and the reasoning behind the system.
- **Business professional user interface**: Business professionals need to interact with the system to be able to perform work on their terms. Technical complexity must be buried deep within the application, business professionals don't care about technical details. Domain complexity is what users should be working with.

1.11. Acquiring Knowledge

There are two general approaches to acquiring knowledge to store and then leverage within a knowledge base[28]:

1. **Rule based approach**: Tell the computer what the knowledge is. Accountants and auditors are highly trained and have the knowledge in their head. All we need is a way of capturing that knowledge and storing it in a knowledge base which could involve a professional accountant putting knowledge into machine-readable form. (i.e. knowledge engineering, expert systems)
2. **Pattern-based approach**: Let the computer work it out by using AI, machine learning or other approaches. This means, feed the computer a load of data and let it figure out the patterns. (i.e. machine learning, data science)

It is not an either-or question. But option 2 needs to be prioritised because it will provide the foundation for AI and machine learning to build on. Machine learning excels where there is a high tolerance for error. There is an extremely low tolerance for error in financial accounting, reporting, auditing, and analysis.



[28] YouTube.com, *Shawn Riley On Artificial Intelligence*, https://youtu.be/Ubq8lTUey7Q

A knowledge-based system draws upon the knowledge of human experts, i.e. accountants and auditors. High-quality curated knowledge can supercharge artificial intelligence application. The more knowledge in the knowledge base, the more the knowledge-based system can do.

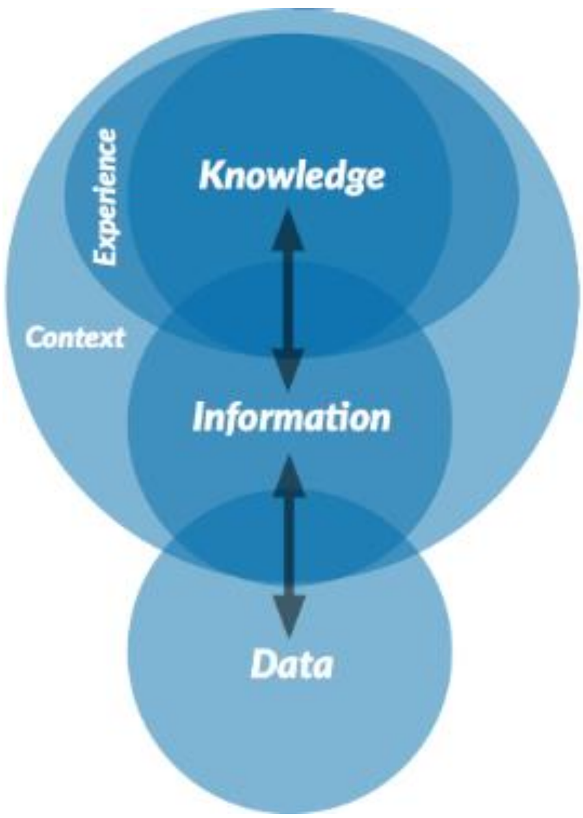
1.12. From Data to Information to Knowledge

Information is meaningful, data is not. Within professional services we are working with information, not data.

The difference between *data* and *information* is that data is the raw numbers and words where information is data in context. This is important to understand as most problems faced by accountants are an information problem, rather than a data problem. Getting data is easy. Knowing what that data represents and how the data fits together is difficult. Representing information in the form that a machine such as a computer can understand and use that information is difficult.

Knowledge is a set of data and information and a combination of skill, know-how, experience which can be used to improve the capacity to take action or support a decision making process.

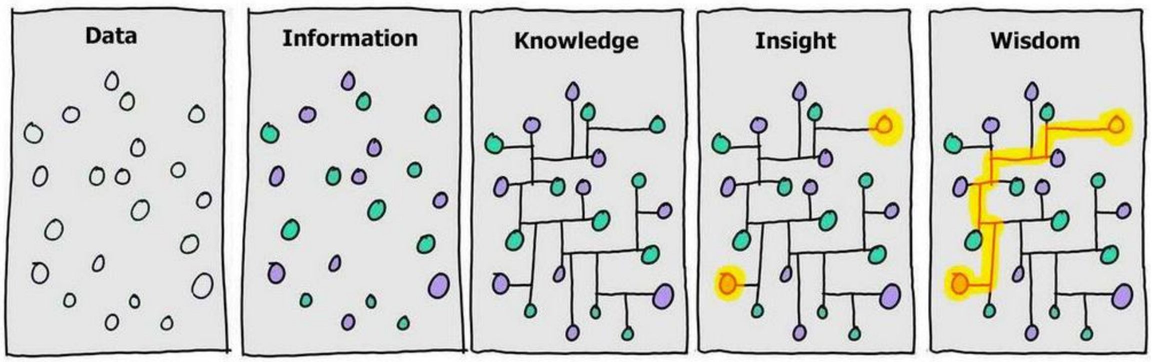
The following graph created by Shawn Riley shows the important to understand differences between data, information, and knowledge[29].



The important point to understand here is that it takes the skill and experience of human professionals to create information and knowledge.

[29] Shawn Riley, *Machine Learning versus Machine Understanding*, https://www.linkedin.com/pulse/machine-learning-vs-understanding-shawn-riley/

Here is another graphic that helps one understand the difference between data, information, knowledge, insight, and wisdom[30]:



Putting this succinctly:

Decision = Data + Knowledge

Knowledge = Ontology + Rules

Algorithm = Logic + Control

1.13. Difference Between Machine-readable, Machine-understandable, Machine-interpretable

In my popular video, *How XBRL Works*[31], I try and explain the difference between information structured for presentation and information structured for meaning. That video is worth watching. But I want to expand that explanation to include the notions of "machine-readability", "machine-understandable", and "machine-interpretable". This spectrum will help you dial in your understanding of the capabilities of computers. If you want all of the details for the examples provided below, please see the SFAC 6 Elements of Financial Statements[32] representation.

1.13.1. Machine-readable

Fundamentally, anything that a computer interacts with has to be structured in some way. Word processing documents, PDF documents, and HTML documents are structured and machine readable; but all of those document formats are structured for the presentation of information in the form of pages, paragraphs, tables, sentences, and other such presentation related structures that contain information and a computer. But a computer does not understand the information that is being conveyed by such documents.

Likewise, even an XBRL-based report is machine-readable but the information contained within the report is not understandable to the computer. Here is a screenshot of a small XBRL-based report[33]:

---

[30] Tumblr, Information vs Knowledge, https://informationversusknowledge-blog.tumblr.com/

[31] YouTube.com, How XBRL Works, https://www.youtube.com/watch?v=nATJBPOiTxM

[32] SFAC 6 Representation, http://xbrlsite.azurewebsites.net/2020/master/sfac6/

[33] SFAC 6 Representation, Machine Readable, http://xbrlsite.azurewebsites.net/2020/master/sfac6/instance.xml

```
<sfac6:Assets contextRef="I-2019" unitRef="U-U-USD" decimals="INF">0</sfac6:Assets>
<sfac6:Assets contextRef="I-2020" unitRef="U-U-USD" decimals="INF">3500</sfac6:Assets>
<sfac6:Liabilities contextRef="I-2019" unitRef="U-U-USD" decimals="INF">0</sfac6:Liabilities>
<sfac6:Liabilities contextRef="I-2020" unitRef="U-U-USD" decimals="INF">0</sfac6:Liabilities>
<sfac6:Equity contextRef="I-2019" unitRef="U-U-USD" decimals="INF">0</sfac6:Equity>
<sfac6:Equity contextRef="I-2020" unitRef="U-U-USD" decimals="INF">3500</sfac6:Equity>
<sfac6:Revenues contextRef="D-2020" unitRef="U-U-USD" decimals="INF">7000</sfac6:Revenues>
```


A computer program such as an XBRL processor can read that information shown above. But a computer is unaware of what information is being represented. A computer does not innately understand the information. Remember, computers are dumb beasts that need to be led by the hand in order to get them to understand.

1.13.2. Machine-understandable

While a computer may not understand the meaning of the information, there are some things that the computer does understand. A machine can take the same information shown above and turn it into this[34]:

Comprehensive Income Statement [Line Items]	Period [Axis] 2020-01-01 - 2020-12-31
Comprehensive Income [Roll Up]	
Revenues	7,000
(Expenses)	(3,000)
Gains	1,000
(Losses)	(2,000)
Comprehensive Income	3,000

If a machine is given additional information about how to render the information, then the machine can understand how to render the information that it is provided; but it still does not actually understand the information that it is working with.

Now, you can provide that understanding; that is not a problem. We will explain this more in a moment. But what we want you to recognize is that is exactly what you need to do; provide the understanding to the computer. Remember, computers are incredibly dumb beasts.

1.13.3. Machine-Interpretable

Once a machine does understand; to the extent of that understanding a machine can interpret information and then take action based on that interpretation.

So, for example, if you provide the understanding that "If the value of comprehensive income is less than $5,000; then SELL the company to remove the company from your investment portfolio," because you have knowledge (i.e. you

[34] SFAC 6 Representation, Human Readable, http://xbrlsite.azurewebsites.net/2020/master/sfac6/evidence-package/contents/Rendering-N1-RE8.html

believe) that companies with comprehensive income of less than $5,000 are not good investments.

1.14. Rules

The Merriam-Webster dictionary defines anarchy[35] as "a situation of confusion and wild behavior in which the people in a country, group, organization, community, etc., are not controlled by rules or laws." Rules prevent information anarchy[36].

Rules enable a knowledge bearer to describe information they are providing and verify that the information provided is consistent with that description. Rules enable a knowledge receiver to understand the description of information provided by the knowledge bearer and likewise verify that the information is consistent with that description.

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.

Rules arise from the best practices of knowledgeable business professionals. A rule describes, specifies, 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 rules are a choice of business professionals. The meaning of a rule is separate from the level of enforcement someone might apply to the rule.

1.15. "Rules as Code" (i.e. Machine-readable Rules)

In his Ted Talk, Jason Morris discusses the notion of "rules as code"[37]. Rules as code is a methodology for creating and applying legal rules, accounting rules, reporting rules, auditing rules and such in the digital age. Some of the important features of Rules as Code are[38]:

- Legislation, statutes, and regulations should be drafted in a natural language and in the form of machine-readable rules at the same time.
- Rules should be declarative.
- The platform on which the legislation or regulations are encoded should be open, accountable, transparent and standardized.
- Definitions should be consistent across all acts, statutes, and regulations; not only within them. Each law should add to the shared dictionary of terms.

---

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

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

37 YouTube.com, Jason Morris, *How programming can make the law more accessible*, https://youtu.be/d5Mt-Q9K7tU

38 American Bar Association, Jason Morris, *Rules as Code*, https://www.lawpracticetoday.org/article/rules-code/

1.16. Network Affect

The network effects[39] (Metcalf's Law[40]) have become an essential component of a successful digital businesses. Network effects typically account for 70% of the value of digitally-related companies. Metcalfe's Law states that a network's impact is the square of the number of nodes in the network.

The Internet itself has become a facilitator for network effects. As it becomes less and less expensive to connect users on platforms, those able to attract them in mass become extremely valuable over time. Also, network effects facilitate scale. As digital businesses and platforms scale, they gain a competitive advantage, as they control more of a market.

1.17. Logic

Logic is a set of principles that forms a **framework for correct reasoning**. Logic is a process of deducing information correctly. Logic is about the correct methods that can be used to prove a statement is true or false. Logic tells us exactly what is meant. Logic allows logical systems to be proven using a logical theory.

The principles of logic are topic-neutral, universal principles which are more general than say the single domain of law, biology, mathematics, accounting, or economics. Logic has to do with the meaning of concepts common to all domains and establishes general rules governing concepts.

Logical truths are necessary. The principles of logic are derived solely using reasoning and the validity of the universal principles are not dependent on any other feature of the world.

Logic is the process of deducing information correctly; **logic is not about deducing correct information**. Understanding the distinction between *correct logic* and *correct information* is important because it is important to follow the consequences of an incorrect assumption. Ideally, we want both our logic to be correct and the facts we are applying the logic to, to be correct.

The primary point here is that correct logic and correct information are two different things. If our logic is correct, then anything we deduce from such information will also be correct per the rules of logic.

As we pointed out, logic plays an important role in achieving computational professional services. But which logic?

1.18. Metalogic

Enter the notion of metalogic[41]. **Metalogic** relates to the comparison between the logic of different systems. As pointed on in *Specifying the Rule Metalogic on the Web*[42], interoperability issues can become problematic if you are using different logics to perform work and evaluate two different logical systems such as two

---

[39] Wikipedia, *Network Effect*, https://en.wikipedia.org/wiki/Network_effect

[40] Metcalf's Law, https://www.thegeniusworks.com/2020/02/metcalfes-law-explains-how-the-value-of-networks-grow-exponentially-there-are-5-types-of-network-effects/

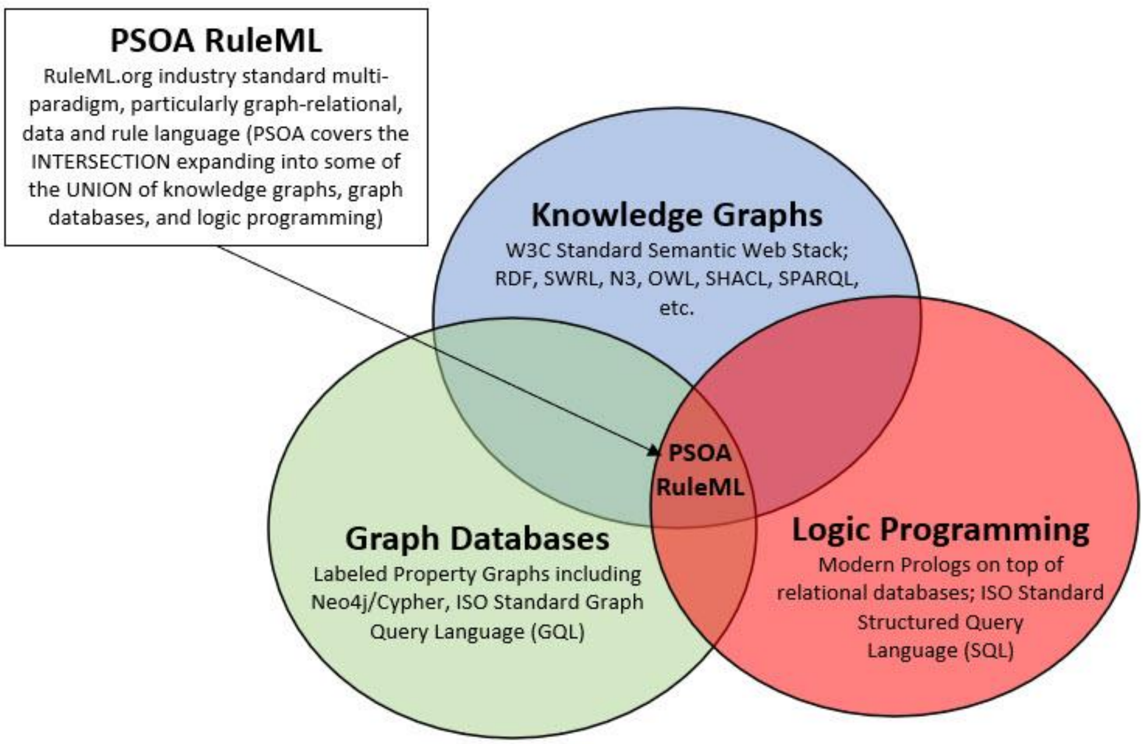
[41] Ted Sider, Logic of Philosophy, page 6, http://tedsider.org/books/lfp_sample.pdf#page=6

[42] Harold Boley, *Specifying the Rule Metalogic on the Web*, http://ruleml.org/ruleml.org/metalogic/RuleMetaLogic2011-10-29.pdf

different financial report models. Both systems, although different software applications, should derive the same logical conclusions.

Now, I have mentioned that there are a number of different logic systems that could be used to represent a logical system: OWL+SHACL+RDF, Modern Prolog, ISO Prolog, Datalog, PSOA, GQL/Cypher, XBRL+More, SQL+More[43] . We will discuss these different implementation alternatives in a moment.

And so, can you prove the same things in one of the systems mentioned above in another one of the systems above? Or saying this another way, is the logic of say OWL+SHACL+RDF equivalent to that of say Modern Prolog? The graphic below helps visualize why metalogic is important[44]:



Logical interoperability is important. In fact, interoperability in general is important.

1.19. Interoperability

Fads, trends, preferences, and other idiosyncrasies impact technology implementation choices. Multiple technical alternatives is a fact of life. As such, interoperability between different alternative choices is necessary.

There have been many different ways to explain what interoperability is and how to achieve it; it seems that there is convergence emerging on this four-level interoperability model[45,46]:

---

[43] Answering the Question of Which Logic, http://xbrl.squarespace.com/journal/2020/8/15/answering-the-question-of-which-logic.html

[44] Primary Problem Solving Logic Paradigms, http://xbrl.squarespace.com/journal/2020/9/15/primary-problem-solving-logic-paradigms.html

[45] HIMSS, Interoperability in Healthcare, https://www.himss.org/resources/interoperability-healthcare

- **Foundational** (Level 1): Establishes the *inter-connectivity requirements* needed for one system or application to securely communicate data to and receive data from another.
- **Structural** (Level 2): Defines the *format, syntax and organization of data exchange* including at the data field level for interpretation.
- **Semantic** (Level 3): Provides for *common underlying models and codification of the data* including the use of data elements with standardized definitions from publicly available value sets and coding vocabularies, providing shared understanding and meaning to the user.
- **Organizational** (Level 4): Includes *governance, policy, social, legal and organizational considerations* to facilitate the secure, seamless and timely communication and use of data both *within and between* organizations, entities and individuals. These components enable shared consent, trust and integrated end-user processes and workflows.

Standards help provide interoperability. For example,

- XBRL International[47] provides the XBRL technical syntax[48] global standard (including a conformance suite for testing software) to enable syntax interoperability
- Object Management Group[49] (OMG) provides the Standard Business Report Model[50] (SBRM) to provide a logical conceptualization of a business report
- RuleML[51] helps to provide business rule and logic interoperability
- *Logical Theory Describing Financial Report*[52] provides semantics of a financial report, building on the SBRM model of a business report

All this complexity makes it appear that computational professional services will be impossible for a business professional to make use of. Can't we just get rid of some of these details, reducing the complexity? Well, let us talk about complexity. But first, let me explain the notion of models and metamodels that help hide complexity.

1.20. Model

Models provide flexibility and controllability. Think of it this way. Intuit's TurboTax was provided as a rudimentary example of computational law. TurboTax works with tax forms. Tax forms are static, they don't change for a tax year. Because forms don't change, it is trivial to get a computer to control a form. At the other end of the spectrum is arbitrary. Computers cannot really work with arbitrary; things are too unpredictable. In between, where you get the control offered by a form but the necessary flexibility is by creating a model. Think of a model as multiple different possible forms.

---

[46] Shawn Riley, *What is Interoperability*, https://www.linkedin.com/posts/shawnriley71_what-is-interoperability-it-is-the-ability-activity-6703288197729918976-FFYD/

[47] XBRL International, XBRL, http://xbrl.org

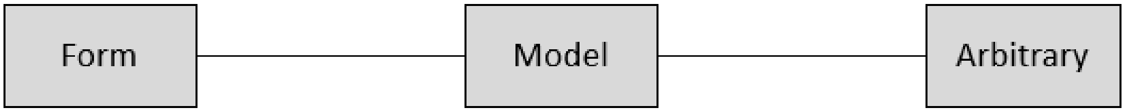
[48] XBRL International, Standards, https://specifications.xbrl.org/specifications.html

[49] OMG, https://www.omg.org

[50] OMG, Standard Business Report Model (SBRM), https://www.omg.org/intro/SBRM.pdf

[51] RuleML, http://wiki.ruleml.org/index.php/RuleML_Home

[52] Logical Theory Describing Financial Report, http://xbrl.squarespace.com/logical-theory-financial-rep/



For example, financial reports are not forms. Each company's report can have different line items and line items can be organized under different subtotals. But what line items can be used and what different organizations can be used follow patterns. Each of those permissible patterns constitutes a model[53]. The high level model of every financial report must fit into the double-entry accounting model and the accounting equation, metamodels.

1.21. Metamodel

Metamodels provide a specification for a model. For example, the Securities and Exchange Commission (SEC) provides a metamodel for an XBRL-based digital financial report. Essentially, the Edgar Filer Manual provides that metamodel. When a company creates their financial report model, they have to conform to that SEC metamodel.

The European Single Market Authority (ESMA) uses a slightly different metamodel than the SEC. Companies reporting to the ESMA us the model defined by the European Single Electronic Format (ESEF).

Both the SEC and ESEF follow the meta-metamodel defined by the OMG Standard Business Report Model (SBRM)[54]. SBRM provides a logical conceptualization of a business report.

1.22. Complexity

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?" Another version of the law of conservation of complexity: "Every application has an inherent amount of complexity that cannot be removed or hidden. Instead, it must be dealt with, either in product development or in user interaction."

Irreducible Complexity is explained as follows: A single system which is composed of several interacting parts that contribute to the basic function and where the removal of any one of the parts causes the system to effectively cease functioning.

So, for example, consider a simple mechanism such as a mousetrap. That mousetrap is composed of several different parts each of which is essential to the proper functioning of the mousetrap: a flat wooden base, a spring, a horizontal bar, a catch bar, the catch, and staples that hold the parts to the wooden base. If you have all the parts and the parts are assembled together properly, the mousetrap works as it was designed to work.

But say you remove one of the parts of the mousetrap. The mousetrap will no longer function as it was designed; it will not work. That is **irreducible**

[53] *Understanding Digital,* Intermediate Components, page 38, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingDigital.pdf#page=38
[54] OMG, Standard Business Report Model (SBRM), https://www.omg.org/intro/SBRM.pdf

complexity: the complexity of the design requires that it can't be reduced any farther without losing functionality.

1.23. Simple

Anyone can create something that is 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. Simplicity is "dumbing down" a problem to make the problem easier to solve. 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. But that is what is necessary to make computational professional services work as desired.

1.24. Viable Implementation Alternatives

Figuring out which logic to use is a "dance" between expressivity[55] and tractability[56], trying to get the right equilibrium for the task being performed. The logic needs to be as powerful as possible but also as reliable as possible (i.e. controllable).

My confidence is pretty high that all of the following seem to provide enough logic, but each also has specific known control issues associated with them so craftsmen need to make adjustments to make sure things work as would be expected:

- **Ontology + Rules**: For example, OWL[57] (or SWRL) + SHACL[58] + RDF[59] (or N3) provide sufficient fragments of first order logic. (Some call this Modern Symbolic AI[60])
- **Modern Prolog**: Prolog such as SWI Prolog[61] or Scryer Prolog[62] seem to have all of the necessary functionality. The up side is that there are a lot of Prolog implementations[63]. The down side is that none of these Prologs can call itself "the standard". Each has pros and cons. Prolog interoperates with relational (SQL) databases.
- **ISO Prolog**: ISO has created a standard Prolog[64]. ISO Prolog can be regarded as a subset of Full Prolog. There is solid motivation for implementations to support ISO Prolog as the international standard Prolog, many already do to one degree or another.

---

55 Revisiting the Knowledge Representation Spectrum, http://xbrl.squarespace.com/journal/2019/10/9/revisiting-the-knowledge-representation-spectrum.html
56 Dictionary.com, *Tractability*, https://www.dictionary.com/browse/tractability
57 W3C, *OWL*, https://www.w3.org/TR/owl2-overview/
58 W3C, *SHACL*, https://www.w3.org/TR/shacl/
59 W3C, RDF, https://www.w3.org/RDF/
60 Shawn Riley, *Modern Symbolic AI in 2020*, https://medium.com/@shawn.p.riley/modern-symbolic-ai-in-2020-dfcc27abbc5c
61 SWI Prolog, https://www.swi-prolog.org/
62 Scryer Prolog, https://github.com/mthom/scryer-prolog
63 Wikipedia, *Comparison of Prolog Implementations*, https://en.wikipedia.org/wiki/Comparison_of_Prolog_implementations
64 ISO, *ISO Prolog*, https://www.iso.org/standard/21413.html

- **Datalog**: Datalog[65], or "function-free Horn Logic", is more tractable than Horn Logic[66] (Pure Prolog) and ISP Prolog (Full Prolog). RuleML.org points out[67], "Datalog is the language in the intersection of SQL and Prolog. It can thus be considered as the subset of logic programming needed for representing the information of relational databases, including (recursive) views." So Datalog interoperates with relational databases.
- **PSOA RuleML**: PSOA[68] (Positional-Slotted Object-Applicative) RuleML is a multi-paradigm, particularly graph-relational, data and rule language. PSOA interoperates with graph and relational databases. RuleML.org points out[69], "PSOA RuleML's databases (fact bases) generalize the instance level of Graph and Relational Databases; its knowledge bases complement facts by rules for deductive retrieval (extending the Datalog-level, function-free expressiveness of Deductive Databases to the Horn-logic expressiveness of Logic Programming), interoperation, and reasoning, as well as for optionally emulating part of the schema level."
- **GQL/Cypher**: GQL[70] is an ISO project[71] to create a global standard query language (like SQL) for graph databases, graph query language. Open Cypher[72] which is based on Cypher is the query language of Neo4j.
- **SQL + More**: While it is proven[73] that you can store XBRL-based information in a relational database; you have to add functionality to process the information. Essentially, you have to construct a what amounts to the functionality of a rules engine to process the information and prove the system is properly functioning. This is very possible but tends to not be very efficient.
- **XBRL + SBRM + More**: XBRL[74] is an open standard technical syntax published by XBRL International, SBRM[75] is a forthcoming standard to be published by OMG that formalizes a logical conceptualization of a business report. While XBRL provides the functionality to represent all that is needed to express knowledge and much of what is necessary to process that knowledge and prove the knowledge is represented correctly. However, certain specific processing is missing that must be supplemented to create a complete system. As such, that additional processing logic must be provided.

There are undoubtedly other logic engines that can be used to process XBRL-based digital financial reports. Other completely different approaches such as the decision

---

[65] Wikipedia, *Datalog*, https://en.wikipedia.org/wiki/Datalog
[66] Wikipedia, *Horn Logic*, https://en.wikipedia.org/wiki/Horn_clause
[67] RuleML.org, http://ruleml.org/papers/Primer/RuleMLPrimer2012-08-09/RuleMLPrimer-p3-2012-08-09.html
[68] RuleML.org, *PSOA*, http://wiki.ruleml.org/index.php/PSOA_RuleML
[69] RuleML.org, *PSOA RuleML Bridges Graph and Relational Databases*, https://wiki.ruleml.org/index.php/PSOA_RuleML_Bridges_Graph_and_Relational_Databases
[70] GQL Standards.org, *GQL Standard*, https://www.gqlstandards.org/
[71] Wikipedia, *GQL Graph Query Language*, https://en.wikipedia.org/wiki/GQL_Graph_Query_Language
[72] OpenCypher.org, *Open Cypher*, https://www.opencypher.org/
[73] Proof representation, http://xbrlsite.azurewebsites.net/2020/master/proof/index.html
[74] XBRL International, https://www.xbrl.org/
[75] OMG, SBRM, https://www.omg.org/intro/SBRM.pdf

model approach[76] could possibly be used but would need to include an ontology-type component. Any syntax used should be 100% convertible to all other syntaxes and be able to round tripped back into the original syntax. Then, you could switch between whatever approach you wanted.

Converting between these logics is very possible. For example, converting between RDF and labeled property graphs is possible[77]. Converting from RDF to SWI Prolog is possible[78]. But 100% conversion is limited to the least common denominator, the set of logic that each alternative possesses.

1.25. Universal vs Domain Specific Applications

Having high-level metamodels such as the forthcoming Standard Business Report Model[79] (SBRM) and Logical Theory Describing Financial Report[80] (see). Plus, creating the base metadata, such as the US GAAP Financial Reporting Metadata[81] leveraging that high-level metamodel makes all of this extremely technical stuff far less technical to business professionals. How? A few trained professionals create the high-level metadata but every software application and business domain professional benefits from that metadata and high-level models. Further, software creation costs are reduced. How is that possible? Read on.

In his book *Systematic Introduction to Expert Systems*[82], Frank Puppe provides the graphic below. The graphic basically points out that **universal, general tools** are less restrictive but cost more to create than **domain-specific tools**. In addition to universal, general tools being more costly to create and more difficult to create; domain specific tools are easier to create and much, much easier for business professionals to use because of the restrictions.

So, a "restriction" is not a flaw. **The restriction is what makes the tool easier to use, cost less, and make software easier to develop**. You don't need the universe of all possible options for a specific domain; you only need to create what that specific domain needs. As **long as you get these restrictions correct**, they really are not "restrictions" of the domain, they are the "boundaries" of the domain. You don't need them.

Technical people don't typically understand these business domain boundaries. Many times, to play it safe technical people add flexibility in order to make certain that business domain user needs are being met. But this flexibility comes at a cost. Additional costs are incurred to create the flexibility and software is harder to use because business professionals need to figure out which option they should use.

---

[76] Wikipedia, *Decision Model*, https://en.wikipedia.org/wiki/Decision_model

[77] Neo4j, Jesús Barrasa, *RDF Triple Stores vs. Labeled Property Graphs: What's the Difference?*, https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/

[78] Samuel Lampa, *SWI-Prolog as a Semantic Web Tool for semantic querying in Bioclipse: Integration and performance benchmarking*, https://www.diva-portal.org/smash/get/diva2:398839/FULLTEXT01.pdf

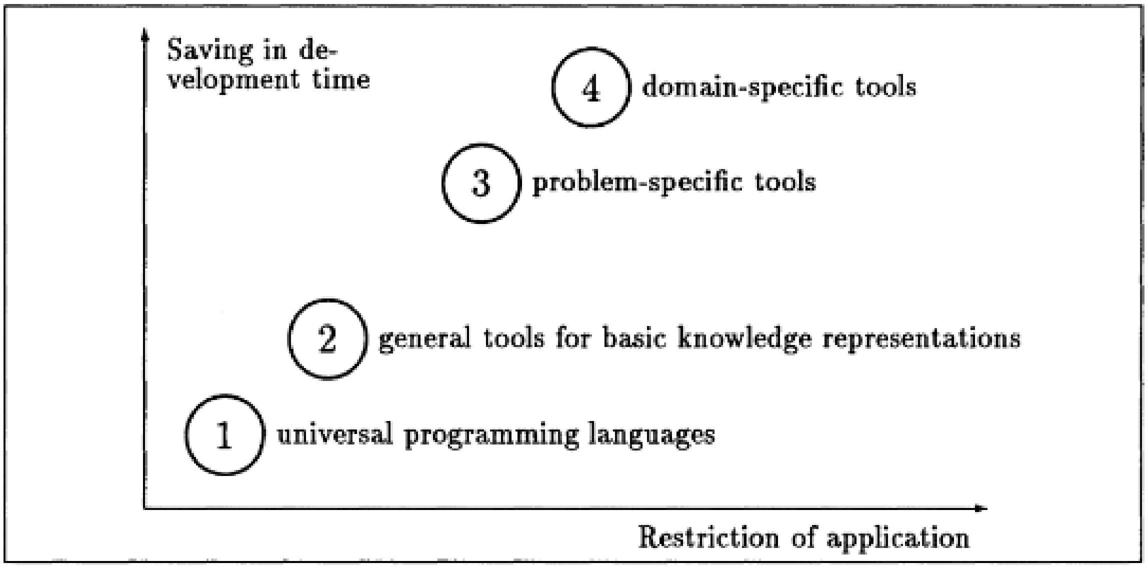
[79] *SBRM Progress Report*, http://xbrl.squarespace.com/journal/2020/1/30/sbrm-progress-report.html

[80] *Logical Theory Describing Financial Report*, http://xbrl.squarespace.com/logical-theory-financial-rep/

[81] *US GAAP Financial Reporting Scheme*, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/us-gaap/documentation/Home.html

[82] Frank Puppe, *Systematic Introduction to Expert Systems*, page 11, https://books.google.com/books?id=_kKqCAAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Business domain people do understand the boundaries if they think about them. Many business professionals cannot properly articulate the appropriate boundaries or restrictions. This communications problem tends to lead to software that costs more to create than is necessary and harder to use than necessary.



This is not an either-or choice. Sometimes universal tools are very appropriate. Other times domain-specific tools are appropriate. Being conscious of these dynamics will lead to the right software being created and the appropriate level of usability. Universal tools are not a panacea. Unconsciously constricting a domain-specific tool when it would have been better to create a more universally usable tool also can be a mistake one makes.

Today, everyone is competing at the "universal tool" level and not one of those universal tools is usable by business professionals. Computational Professional Services is a vertical; but it is an incredibly WIDE (i.e. horizontal) vertical market.

1.26. Process Control

Because, as we pointed out, financial reports are not static forms and therefore individual economic entities are allowed to make specific modifications to models; those modifications need to be controlled in order to maintain information quality. Said another way, permissible modifications to the model must be crystal clear to those making such modifications.

Lean Six Sigma[83] philosophies and techniques offer many insights and ideas related to process control. The 1-10-100 Rule is related to what's called "the cost of quality." Essentially, the rule states that prevention is less costly than correction which is less costly than failure. It makes more sense to invest $1 in prevention, than to spend $10 on correction. That in turn makes more sense than to incur the cost of a $100 failure[84].

[83] Charles Hoffman, CPA, Lean Six Sigma, http://www.xbrlsite.com/mastering/Part01_Chapter02.K_LeanSixSigma.pdf

[84] Michael Canic, *The Cost of Quality: The 1-10-100 Rule*, https://www.makingstrategyhappen.com/the-cost-of-quality-the-1-10-100-rule/

A kludge (or kluge) is an engineering/computer science term that describes what is best described as a workaround or quick-and-dirty solution that is typically clumsy, inelegant, inefficient, difficult to extend and hard to maintain; but it gets the job done. By contrast, elegance is beauty that shows unusual effectiveness and simplicity.

1.27. Building on the Shoulders of Giants

Rather than "reinventing the wheel", we are building on the shoulders of giants (i.e. existing, proven technologies).

A logical system[85] is a type of formal system[86]. A financial report is a type of formal system. To be crystal clear what I am trying to create is a **finite model-based deductive first-order logic system**[87].

"Finite" as opposed to "infinite" because finite systems can be explained by math and logic, infinite systems cannot. "Model-based" is the means to address the necessary variability (i.e. required flexibility) inherent in the required system. "Deductive", or rule-based, as contrast to inductive which is probability based which is not appropriate for this task. "First-order logic" because first-order logic can be safely implemented within software applications and higher order logics are unsafe. "System" because this is a system.

The point is to create a logical system that has high expressive capabilities but is also a provably safe and reliable system that is free from catastrophic failures and logical paradoxes which cause the system to completely fail to function. To avoid failure, computer science and knowledge engineering best practices seems to have concluded that the following alternatives are preferable:

- **Systems theory**: A system[88] is a cohesive conglomeration of interrelated and interdependent parts that is either natural or man-made. Systems theory explains logical systems.
- **Logical theory**: (a.k.a. logical system) There are many approaches to representing "ontology-like things" in machine-readable form[89], a logical theory being the most powerful.
- **Proof theory**: The ideas of proof theory[90] can be used to verify the correctness of logical systems and computer programs working with those machine-readable logical systems.
- **Set theory**: Set theory is foundational to logic and mathematics. Axiomatic (Zermelo–Fraenkel) set theory[91] is preferred to naïve set theory.

---

85 Wikipedia, *Logical Systems*, https://en.wikipedia.org/wiki/Logic#Logical_systems

86 Wikipedia, *Formal System*, https://en.wikipedia.org/wiki/Formal_system

87 Wikipedia, *First-order Logic, Deductive System*, https://en.wikipedia.org/wiki/First-order_logic#Deductive_systems

88 Wikipedia, *Systems Theory*, https://en.wikipedia.org/wiki/Systems_theory

89 Difference between Taxonomy, Conceptual Model, Logical Theory, http://xbrl.squarespace.com/journal/2018/12/11/difference-between-taxonomy-conceptual-model-logical-theory.html

90 Stanford University, *The Development of Proof Theory, The Aims of Proof Theory*, https://plato.stanford.edu/entries/proof-theory-development/#AimProThe

91 Wikipedia, *Set Theory, Axiomatic Set Theory*, https://en.wikipedia.org/wiki/Set_theory#Axiomatic_set_theory

- **Graph theory**: Directed acyclic graphs[92] are preferred to less powerful "trees" and graphs which contain cycles that can lead to catastrophic problems caused by those cycles.
- **Logic**: Logic is a formal communications tool. Horn logic[93] is a subset of first-order logic which is immune from logical paradoxes should be used as contrast to more powerful but also more problematic first order logic features. Note that deductive reasoning is leveraged for the process of creating a financial report and not inductive reasoning (i.e. machine learning)
- **Model theory**: Model theory is a way to think about flexibility. Safer finite model theory[94] is preferable to general model theory.
- **World view**: The following are common issues which appear when implementing logical systems in machine-readable form, the safest and most reliable alternatives are:
    - closed world assumption[95] which is used by relational databases is preferred to the open world assumption which can have decidability issues[96];
    - negation as failure[97] should be explicitly stated;
    - unique name assumption[98] should be explicitly stated;

Business professionals are (a) not capable of having precise discussions of these sorts of issues with software engineers, (b) don't care to have such technical discussions about these sorts of issues with software engineers, (c) are not interested in the theoretical or philosophical or religious debates that commonly exist related to these alternatives, (d) if the alternatives were appropriately articulated to a business professional, who tend to be very practical, they would **most often error on the side of safety and reliability**.

While some implementations can have decidability issues or termination problems or other such issues, a master craftsman is knowledgeable of these issues and so they can work around such problems.

1.28. Effective Computational Professional Services

Clearly for computational professional services to be useful, it actually needs to work, be reliable, be predictable, and provide benefits in terms of better, faster, or cheaper professional services offerings.

'Hope' is not a sound engineering principle. To make computational professional services work one needs "know how". Know how is a type of practical knowledge.

Rather than rushing into the details; take a step back consider the utility that a framework, theory, principles, and method would very likely provide. Then, brick-by-brick, much like building a house, business domain experts and software

---

92 Wikipedia, *Directed Acyclic Graph*, https://en.wikipedia.org/wiki/Directed_acyclic_graph

93 Wikipedia, *Horn Logic*, https://en.wikipedia.org/wiki/Horn_clause

94 Wikipedia, *Finite Model Theory*, https://en.wikipedia.org/wiki/Finite_model_theory

95 Wikipedia, *Closed World Assumption*, https://en.wikipedia.org/wiki/Closed-world_assumption

96 Wikipedia, *Undecidability Problem*, https://en.wikipedia.org/wiki/Undecidable_problem

97 Wikipedia, *Negation as Failure*, https://en.wikipedia.org/wiki/Negation_as_failure

98 Wikipedia, *Unique Name Assumption*, https://en.wikipedia.org/wiki/Unique_name_assumption

engineers can create tools that automate certain types of tasks in that process and let computational professional services unfold.

Computational professional services is not about computers replacing humans. Computational professional services is about using machines to do what they do best and humans to do what they do best. Computational professional services is about human-machine collaboration; much like how a calculator is used to help humans do math. Computational professional services is about augmenting the capabilities of humans by leveraging machines.

1.29. Framework, Theory, Principles

What is conspicuously lacking from most people's minds is a broad framework let alone a theory and principles on how to think about computational professional services.

A **framework** is an aid that enables a community of stakeholders with a set of rules, ideas, or beliefs which provides a structure in order to think about or implement something. A **theory** enables a community of stakeholders trying to achieve a specific goal or objective or a range of goals/objectives to agree on important statements used for capturing meaning or representing a shared understanding of and knowledge in some universe of discourse. **Principles** help you think about something thoroughly and consistently. Overcoming disagreements between stakeholders and even within groups of stakeholders is important and principles can help in that communications process.

The *Logical Theory Describing Financial Report*[99] provides a framework, theory, and principles for thinking about XBRL-based digital financial reporting. This can serve as an example of how to think about computational professional services.

1.30. Best Practices

A **best practice** is a method or technique that has been generally accepted as superior to any other known alternatives because it produces results that are superior to those achieved by other means or because it has become a standard way of doing things.

Best practices (or good practices) are techniques that have produced outstanding results in other situations, inside or outside of a particular organization and which can be validated, codified, and shared with others and recommended as models to follow[100].

1.31. Final Thoughts

While XBRL-based digital financial reporting does not have all the answers and is not perfect, it does provide a tremendous amount of useful insight. To understand more details, please read the 161 pages that make up the ***XBRL-based Digital Financial***

[99] Charles Hoffman, CPA, Logical Theory Describing Financial Report, http://xbrl.squarespace.com/logical-theory-financial-rep/

[100] European Guide to good Practice in Knowledge Management - Part 5: KM Terminology, page 3 (PDF page 9), http://arielsheen.com/wp-content/uploads/2019/10/CEN-CWA14924-05-2004-Mar.pdf#page=9

Reporting Jump Start[101]. Want even more details? Try ***Mastering XBRL-based Digital Financial Reporting***[102].

Fundamentally, the objective is effective, repeatable automation. To provide effective, repeatable automation you need:

- Repeatable processes that have control mechanisms that assure high-quality DATA and KNOWLEDGE within the system
- Process automation using machine-readable algorithms (ALGORITHM = CONTROL + LOGIC)
- Implementation of self-service KNOWLEDGE. (KNOWLEDGE = ONTOLOGY + RULES)

Automation[103] is the technology by which a process or procedure is performed with minimal human assistance. Stringing the right capabilities together can, and will, enable effective, repeatable, and therefore reliable computational professional services.

[101] XBRL-based Digital Financial Reporting Jump Start, http://xbrl.squarespace.com/journal/2020/8/6/xbrl-based-digital-financial-reporting-jump-start.html
[102] Charles Hoffman, CPA, Mastering XBRL-based Digital Financial Reporting, http://xbrl.squarespace.com/mastering-xbrl/
[103] Wikipedia, *Automation*, https://en.wikipedia.org/wiki/Automation