1. Logical Digital Twin of Financial Reports

They are referred to as many different names including knowledge graphs, knowledge base, graphs, networks, digital twins, mirror world, logical twins, or logical digital twin. I will use the term *logical digital twin* or professional knowledge graphs.

A *Logical Digital Twin* is a special, formal type of knowledge graph[1] or d*igital twin*[2] used by professionals. The freemasons of the information age[3] will use these tools to reconstruct how work is performed in the information age.

Effectively, a logical digital twin is a knowledge graph of logic that has a logical schema that enforces the logic of the logical digital twin to always assure that the information represented by the logical digital twin is a verifiably properly functioning logical system within the specific logical boundaries of that logical schema.

You can think of a logical digital twin as a specific point in a spectrum of knowledge graphs. A logical digital twin is a professional quality knowledge graph that is used when high quality knowledge graphs are necessary such as an XBRL-based digital financial report.

A financial report is a knowledge graph[4]. Obviously financial report knowledge graphs must be professional, high-quality knowledge graphs. I have been promoting the notion of using a logical schema[5] to enforce the logic within a financial report knowledge graph in order to maintain and prove the quality of those knowledge graphs for years. In addition to report quality, such a logical schema can enable expert system software applications to be created to process such reports (i.e. construct the report, or analyze the information contained within the report).

The examples of professional knowledge graphs used for financial reporting provide an example of what is possible with other applications of business reporting. Apparently, I have a unique approach to creating knowledge graphs[6].

For years, financial reporting teams have used countless spreadsheets and a patchwork of tools for collaboration, workflow management, project management, file transfers, approvals, and such to complete the tasks and processes required to construct financial reports. No more.

This document contains information that helps the reader to understand the moving pieces involved with having a precise and accurate understanding of the notion of knowledge graphs generally and professional quality logical digital financial reports in particular.

---

[1] Wikipedia, Knowledge Graph, https://en.wikipedia.org/wiki/Knowledge_graph

[2] Digital Twin for Financial Status and Performance of Economic Entity, https://digitalfinancialreporting.blogspot.com/2023/09/digital-twin-for-financial-status-and.html

[3] Freemasons of the Information Age, https://digitalfinancialreporting.blogspot.com/2023/10/freemasons-of-information-age.html

[4] Charles Hoffman, CPA, *Financial Report Knowledge Graph*, https://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf

[5] Charles Hoffman, CPA, *Logical Schema of Financial Report*, http://xbrlsite.com/seattlemethod/LogicalSchemaOfFinancialReports.pdf

[6] Seeing Digital Financial Reporting Differently, https://digitalfinancialreporting.blogspot.com/2023/12/seeing-digital-financial-reporting.html

1.1. Audit is Broken

A Fortune article, *Wirecard shows auditing is broken. Here's why—and how to fix it*[7], points out that a September 2019 investigation by the Project on Government Oversight, a Washington, D.C., watchdog group, revealed that when the PCAOB has inspected Big Four audits, it found frighteningly high failure rates. In the most recent figures available, inspectors found Deloitte got one in 20% wrong, PwC botched 23.6%, EY screwed up 27.3%, and KPMG flopped fully 50% of the time.

Another Forbes article, *22 years after the $63 billion Enron collapse, a key audit review board finds the industry in a 'completely unacceptable' state*[8], point out; audit is out of control and the situation is "completely unacceptable" per the PCAOB.

CFO magazine also says audit is broken[9]. In fact, it seems hard to find someone that will tell you that audit is not broken.

What if there was a better way of audit? Engine B talks about the notion of "always-on" audit[10]. MindBridge talks about AI assisted audit[11]. The AICPA says AI is a gamechanger for audit[12].

1.2. Outdated Old School Financial Report Creation Process

The current process of creating financial reports is inefficient, error-prone, many aspects are manual, outdated, and in need of being revamped. Here is what some are saying[13]:

- **CFA Institute**: calls for "...greater efficiencies within the current inefficient system" [of creating financial reports].
- **Gartner**: "...average Fortune 1000 company used more than 800 spreadsheets to prepare its financial statements"
- **Ventana Research**: "...for larger companies, assembling the periodic external reports typically is an inefficient and error-prone process."
- **PriceWaterhouseCoopers**: "...old school manual processes..." and "commonly cut and pasted, rekeyed, or manually transferred into word processing and spreadsheet applications used for report assembly and review process steps"

The problems with spreadsheets tend to be well known and well understood[14]. Yet, accountants love their spreadsheets. Is there another way?

---

[7] Fortune, *Wirecard shows auditing is broken. Here's why—and how to fix it*, https://fortune-com.cdn.ampproject.org/c/s/fortune.com/2020/06/25/wirecard-auditing-is-broken-fintech-ey-ernst-and-young/amp/

[8] Fortune, *22 years after the $63 billion Enron collapse, a key audit review board finds the industry in a 'completely unacceptable' state*, https://fortune-com.cdn.ampproject.org/c/s/fortune.com/2023/07/26/pcaob-audit-completely-unnacceptable-error-rate-enron-big-4-consulting/amp/

[9] CFO.com, *Audits Are Broken. Here's a Radical Way to Fix Them*, https://www.cfo.com/news/audits-are-broken-heres-a-radical-way-to-fix-them/657144/

[10] Engine B, Audit Ethics, https://engineb.com/2020/11/audit-ethics-and-technology-what-is-it/

[11] MindBridge, https://try.mindbridge.ai/book-demo/

[12] AICPA, *Artificial intelligence is a game changer for auditors*, https://www.aicpa-cima.com/news/article/artificial-intelligence-is-a-game-changer-for-auditors

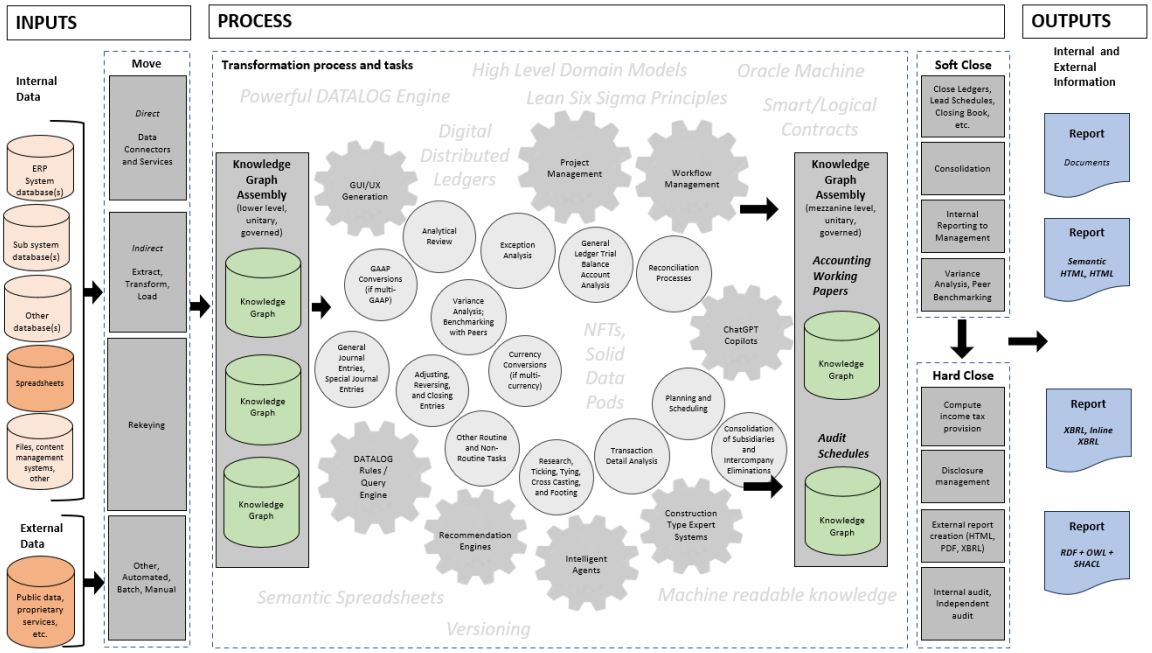
[13] *Old School Financial Report Creation Process*, http://xbrlsite.azurewebsites.net/2017/Library/OldSchoolProcessesIneffecient.pdf

1.3. Fundamental Business Use Case

Businesses exchange information. All kinds of information. One common type of information exchanged is financial information such as the general-purpose financial report[15]. Another more general tool for exchanging information and also for the preparation of general-purpose financial reports is the electronic spreadsheet. In the past, these general-purpose financial reports and electronic spreadsheets were presentation oriented. That means while computers could read and understand how you want the information in those artifacts presented; the computer did not understand what information was provided in those artifacts or what to do with that information.

But what if a computer could understand the information contained in a general-purpose financial report or electronic spreadsheet? What if we created a new type of general-purpose financial report[16] and a new, modern type of spreadsheet[17]?

What does making general purpose financial reports and electronic spreadsheets understandable to computer software actually mean? What are the real capabilities of tools like artificial intelligence to help financial accountants, auditors, and analysts perform work? How do you apply these new technologies to get them to actually work to solve real world business problems? What new skills to you need to make this actually possible? What tasks and processes can be improved by using these new capabilities that enhance the skills of professional accountants, auditors, and analysts?



[14] *The Problem with Spreadsheets*, https://www.youtube.com/watch?v=wbiVK6HKHHg

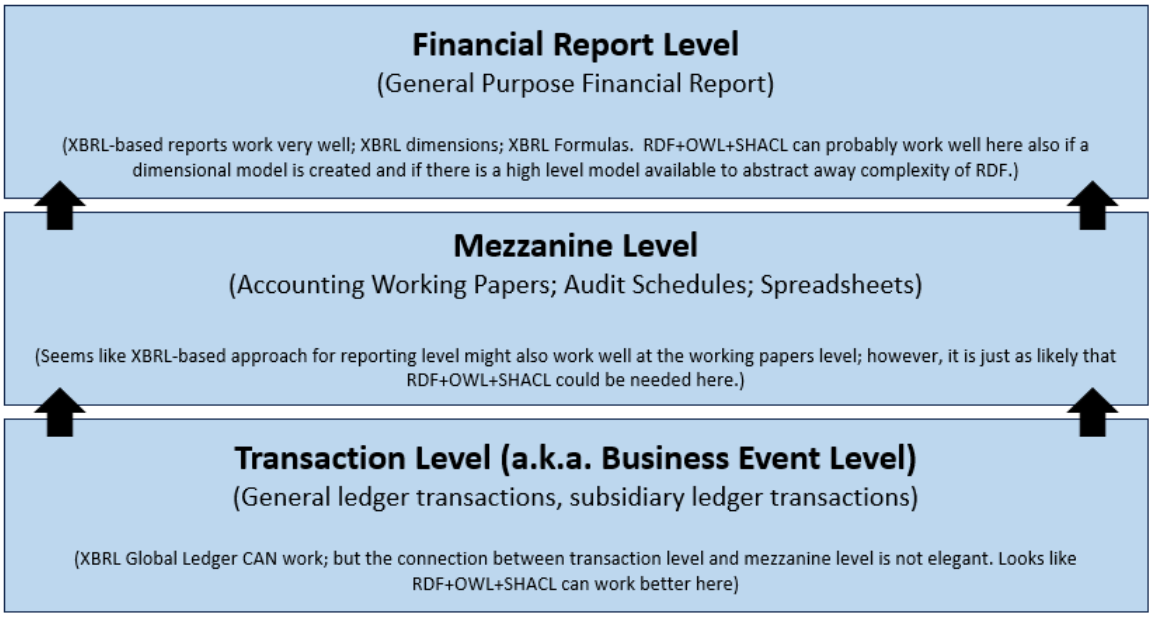
[15] General Purpose Financial Reporting Support for XBRL, https://digitalfinancialreporting.blogspot.com/2023/02/general-purpose-financial-reporting.html

[16] Universal Technology of Accountability, https://digitalfinancialreporting.blogspot.com/2023/02/universal-technology-for-accountability.html

[17] Modern Spreadsheet, https://digitalfinancialreporting.blogspot.com/2023/05/modern-spreadsheet.html

These tools do not just work for general purpose financial reports and electronic spreadsheets. These new tools enable the creation of entirely new business models and completely different paradigms for performing the tasks and processes that make up the day-to-day work that is performed by accountants, auditors, and analysts.

Considering the flow of financial and non-financial information from an enterprise resource planning (ERP) system and other systems that support the creation of the information that then ends up in the form of a general-purpose financial report. Those artifacts seem to fit into three broad categories:



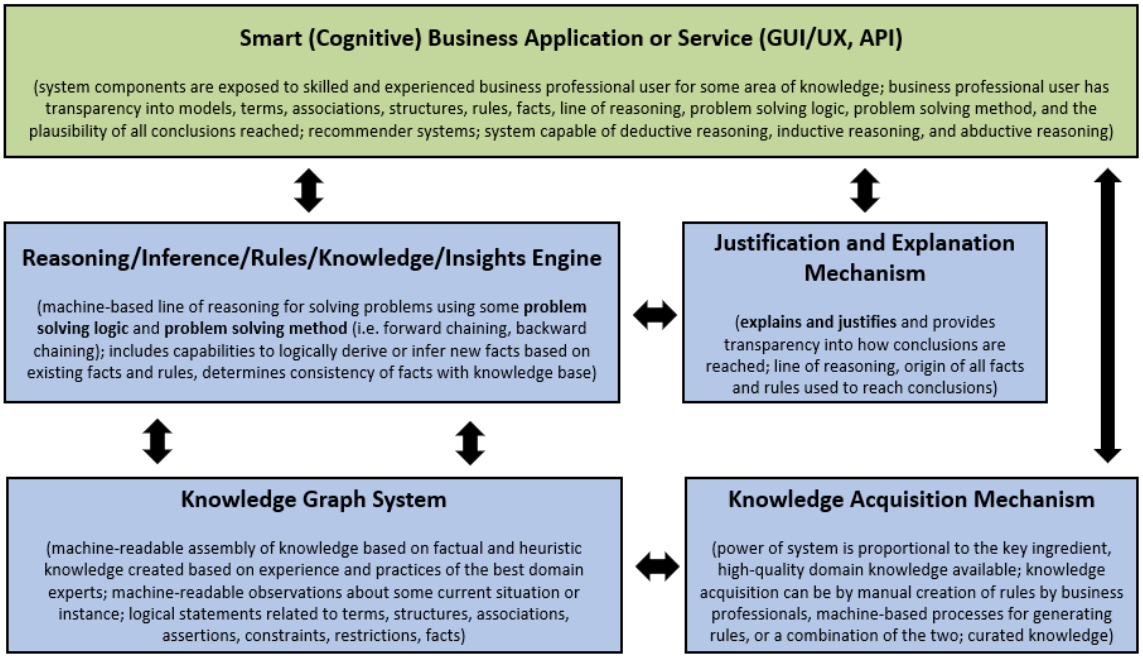
There are many technologies that are converging creating new possibilities and more modern approaches to financial accounting, reporting, auditing, and analysis[18].

1.4. Smart Systems

How much do the electronic word processing documents that you type information into or the electronic spreadsheets you use understand financial reporting, accounting, auditing, or analysis?

Well, what if they could understand? Guess what. They can. This graphic below shows the high-level components of a smart business application or service:

[18] Converging Paths to a Modern Approach to Financial Accounting, Reporting, Auditing, and Analysis, https://digitalfinancialreporting.blogspot.com/2023/09/converging-paths-to-modern-approach-to.html



Using the proper tools provided by artificial intelligence and structured information; very reliable and useful capabilities can be created. Other capabilities can take advantage of other technologies such as machine learning, large language models (LLMs), transformers, expert systems, intelligent agents.

1.5. Today's Professional Data Janitors

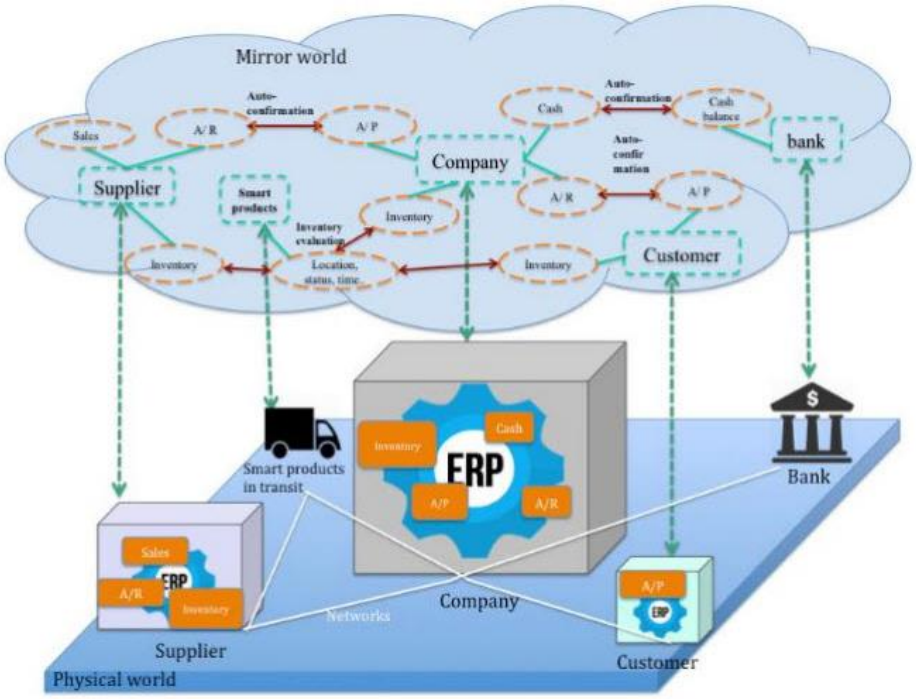
Today, professional accountants, auditors, and analysts perform what amounts to data janitorial services on the data in their information systems that tend to be improperly configured and otherwise set up in a world that currently has poor semantic hygiene practices. This information systems hairball that has been glued together with spreadsheets can best be described as a kludge that can get the job done; but at what cost? The "bailing wire" and "band aids" that are used provide systems that are brittle, error prone, and all this is held together by professional accountants who tend to be overwhelmed much of the time working long hours to keep these systems running.

Those same technologies that are causing the ever-increasing volume and complexity of information that is overwhelming us is also the solution to the information overload that is being experienced.

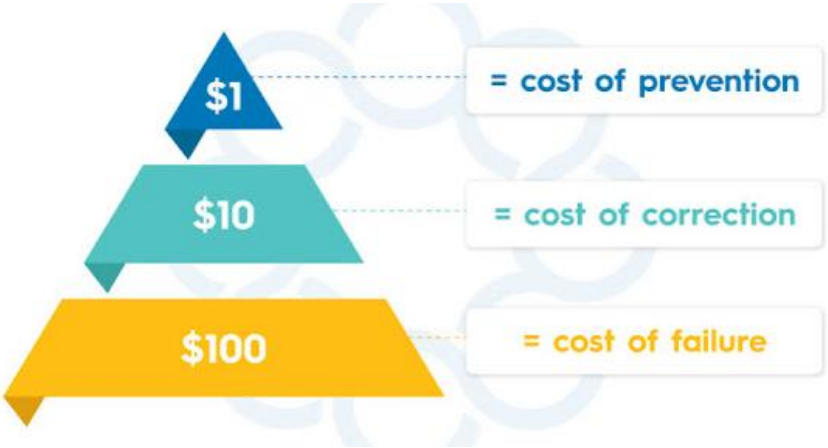
1.6. Understanding the Opportunity

The first I have heard about the notion of a "digital twin" was in the paper, *Imagineering Audit 4.0*, written by Jun Dai and Miklos Vasarhelyi of Rutgers University[19]. In that paper they used the term "mirror world" and used that term to describe the use of technology to create machine readable virtual copy of the real world.

[19] Jun Dai and Miklos Vasarhelyi, *Imagineering Audit 4.0*, page 10, figure 3, https://publications.aaahq.org/jeta/article-abstract/13/1/1/9242/Imagineering-Audit-4-0



One of the principles of Lean Six Sigma[20] is the 1-10-100 rule as explained by Inspectorio[21]: Comparing relative cost of preventing, correcting, and cost of errors, consider this. In relative terms, fixing a system to prevent a problem costs say $1 whereas having to correct a problem costs $10 as contrast to having to deal with the cost of the failure related to not detecting the problem is about $100.



The principles and techniques of Lean Six Sigma have been used to improve the quality of the products output by industries such as the manufacturing of automobiles and airplanes. These same ideas, principles, and techniques can be applied to financial accounting, reporting, auditing, and analysis. A first step in your journey into a new way to work, partnering with computer software applications that augment your skills, is to think about a financial report as a logical system.

[20] Lean Six Sigma, http://www.xbrlsite.com/mastering/Part01_Chapter02.K_LeanSixSigma.pdf

[21] Solving the Problems of the "Accidental Taxonomists" and the "Data Janitors", https://digitalfinancialreporting.blogspot.com/2024/01/solving-problems-of-accidental.html

Many people misinterpret the market size for XBRL-based reports because they confuse regulator mandates to submit XBRL-based reports and the real opportunity structured XBRL-based reports offer. An entirely new financial reporting paradigm is possible. This table below provides an approximate size of the market for creation of general-purpose financial reports:

Market segment	Approximate market size
Public companies financial reporting using US GAAP to SEC	About **10,000** public companies in US
Private company financial reporting in support of commercial loans to banks and others using US GAAP	About **27.9 million** private companies in US; **18,500** private companies with 500 employees or more
Not-for-profit entities financial reporting for commercial loans, federal grants using US GAAP	About **320,000** not-for-profit entities
State and local governmental entities financial reporting using governmental accounting standards; CAFR in US, IPSAS in rest of world	About **90,000** state and local governmental entities (probably similar number or double in the rest of the world)
Listed companies reporting under IFRS	About **10,000 to 30,000** globally
SMEs (small and medium size entities) reporting under IFRS	About **23 million in Europe, 40 million in China, 1.9 million in India, 2.4 Brazil** (private companies, SMEs)
Employee benefit plan annual audit reports, Department of Labor's Employee Benefits Security Administration (EBSA)	About **800,000** audited plans in US under ERISA; similar reporting in Australia, about **300,000** plans
Personal financial statements	About **1,000,000** high-net worth individuals or more

Also think about the infrastructure related to the compilation, review, or audit of such reports. Also consider internal management reporting, tax related reporting, and business reporting in general.

1.7. Fool Proof Accounting = Good, Usable Reporting

Everyone loves sexy information dashboards that they can use to do things like manage a small business effectively. But here is the deal: if the information in that dashboard is wrong and/or not timely or is otherwise not trustworthy; then what good is the dashboard?

I am a certified public accountant (CPA) and have been doing financial accounting for about 45 years with a focus on accounting information systems. In addition, I studied and received an MBA in what was then referred to as "world class manufacturing techniques" then and is now called Lean Six Sigma.

Many small businesses (and a lot of big businesses) have problems with their reporting systems because information that is put into their systems is incorrect or incomplete. Further, the systems are not set up well which causes further struggles when you try and get information out of the system. Another problem is that descriptive metadata that helps you get information out of the system effectively is not put into the system at all or added to the system late in the process which causes problems getting information in a timely manner. Another big problem is trying to figure out where in the accounting system database information is stored so

that you can get the information out of the system and put that information into a report.

The larger an organization gets, the higher the probability that multiple systems exist to store information and the systems don't talk to one another effectively. The typical solution today? Hire expensive accountants and IT people to overcome these integration problems, delay reporting so problems can be fixed, or rekey or copy/paste information, etc.

But there is a better way. Build better systems.

Remember the Lean Six Sigma principle referred to earlier as the "1-10-100 rule" that helps you understand the true cost of poor quality? For every $1 you spend to fix a system to get rid of a problem; you would spend $10 to discover errors and then fix those errors; or you would spend $100 to deal with the consequences of that mistake if the mistake is not detected and corrected.

These are the problems I see with financial accounting systems over and over and over:

1. **Inadequate account type scheme**: Accounting systems generally have a scheme for assigning a "type" to an account; basically, a method of classifying the accounts in a chart of accounts. Categories tend to be "Assets", "Liabilities", "Equity", "Income", "Expenses". Or, there may not be an account type. This is completely inadequate and causes problems. This problem tends to be caused by the design of an accounting system. Proper account types need to be used to avoid work later.
2. **Poorly set up chart of accounts**: It has been my observation that accountants are very good at working with accounting systems; but they are terrible at setting those systems up. A key to setting an accounting system up effectively is to create a good, well-thought-out chart of accounts that help you report rather than getting in the way.
3. **Missing transaction metadata**: Critically important information that enables the proper categorization of transactions tends to be missing from accounting systems and then added later in the process, generally using spreadsheets. To understand this issue, let me point out how Workday solves this issue. Workday, an ERP system, has the notion of the "work tag"[22]. A work tag is an informal approach to adding this missing transaction metadata.
4. **Recording business event information incorrectly**: There are many reasons business event information gets recorded incorrectly, those reasons tend to fall in the following buckets: poorly trained personnel, bad mismatch of skills/experience to tasks being performed, bad accounting system design, sloppiness.
5. **Delays in fixing mistakes**: Any delay in fixing a mistake causes potential mistakes in using that information. First, systems should be fixed so mistakes are avoided. But if a mistake does occur, that mistake should be detected and fixed as soon as possible. Waiting until the end of the month or end of the year and letting the CPA fix it is not a good strategy.

[22] Workday, Tales of the Cloud: The Story of Worktags, https://blog.workday.com/en-us/2012/tales-of-the-cloud-the-story-of-worktags.html

6. **Avoid spreadsheets**: Spreadsheets are not a solution to a problem; they are the problem. Spreadsheets tend to be informal work-arounds.
7. **Overly manual process control mechanisms**: Process control mechanisms today tend to be overworked accounting professionals that have to manually control process quality. This manual approach is expensive, not reliable enough, and ultimately causes more important work to be delayed or simply left undone as the current, immediate crisis is addressed. The fix? Augment manual processes with automated processes and let machines help overworked humans get work done. Leverage things like Lean Six Sigma philosophies and techniques.

Fundamentally, don't fight symptoms; solve the problem that is causing the symptom. This can be hard at times but the proper investments in the right areas will pay dividends in the long term. Financial reporting is a manufacturing or construction process. What is manufactured/constructed is the information.

Fool proof accounting results in good reporting. Without very good reporting, it is impossible to really have algorithmic regulation.

1.8. Algorithmic Regulation

The document *The Great Transmutation*[23] discusses the changes that are occurring and the notion that many call algorithmic regulation. People refer to this paradigm shift in different ways. Here are how some people package this paradigm shift:

- MIT refers to this as **Algorithmic Business Thinking**[24]
- Carnegie Mellon University refers to this as **Computational Thinking**[25]
- Harvard University refers to this as **Regulation, the Internet Way**[26]
- Vanderbilt University refers to this as **Regulation 2.0**[27]
- The Data Coalition calls this **Smart regulation**[28]
- Tim O'Reilly Founder and CEO O'Reilly Media Inc. calls it **Algorithmic regulation**[29]
- Deloitte refers to this as "**The Finance Factory**" and **Digital Finance**[30]

---

[23] Charles Hoffman, CPA, *The Great Transmutation*, http://xbrlsite.azurewebsites.net/2022/Library/TheGreatTransmutation.pdf

[24] MIT, Accelerating Digital Transformation with Algorithmic Business Thinking, https://executive.mit.edu/course/accelerating-digital-transformation-with-algorithmic-business-thinking/a056q00000URaaQAAT.html

[25] Carnegie Mellon Center for Computational Thinking, https://www.cs.cmu.edu/~CompThink/

[26] Harvard University, Regulation, the Internet Way, https://datasmart.ash.harvard.edu/news/article/white-paper-regulation-the-internet-way-660

[27] SSRN, *Regulation 2.0: The Marriage of New Governance and Lex Informatica*, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2746229

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

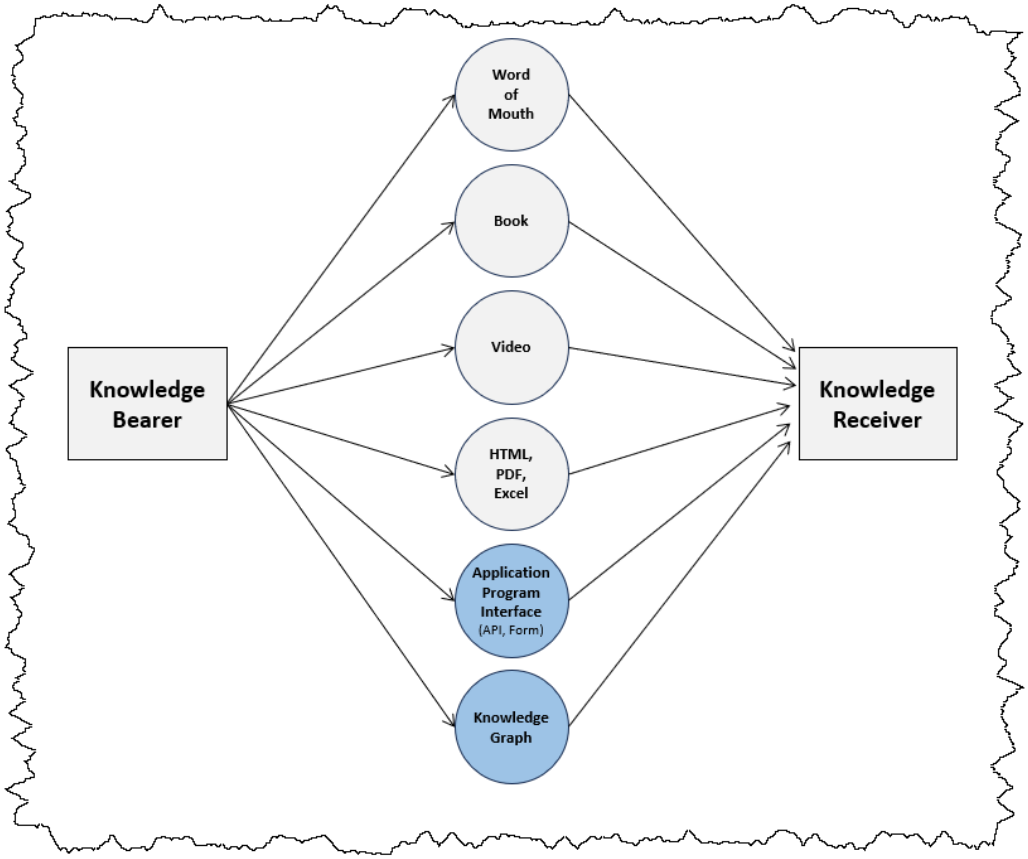
[29] Tim O'Reilly Founder and CEO O'Reilly Media Inc., Open Data and Algorithmic Regulation, https://beyondtransparency.org/chapters/part-5/open-data-and-algorithmic-regulation/

[30] Deloitte, *Finance 2025: Digital transformation in finance*
Our eight predictions about digital technology for CFOs, https://www2.deloitte.com/us/en/pages/finance-transformation/articles/finance-digital-transformation-for-cfos.html

- Robert Kugel of Ventana Research calls it "**Digital Finance**"[31]
- The government of Norway calls this "**Nordic Smart Government and Business**"[32]

1.9. Mediums of Exchange

New mediums of exchange are available. A logical digital twin of a financial report is a knowledge graph that can be used to exchange information reliably and therefore effectively:



1.10. Environment

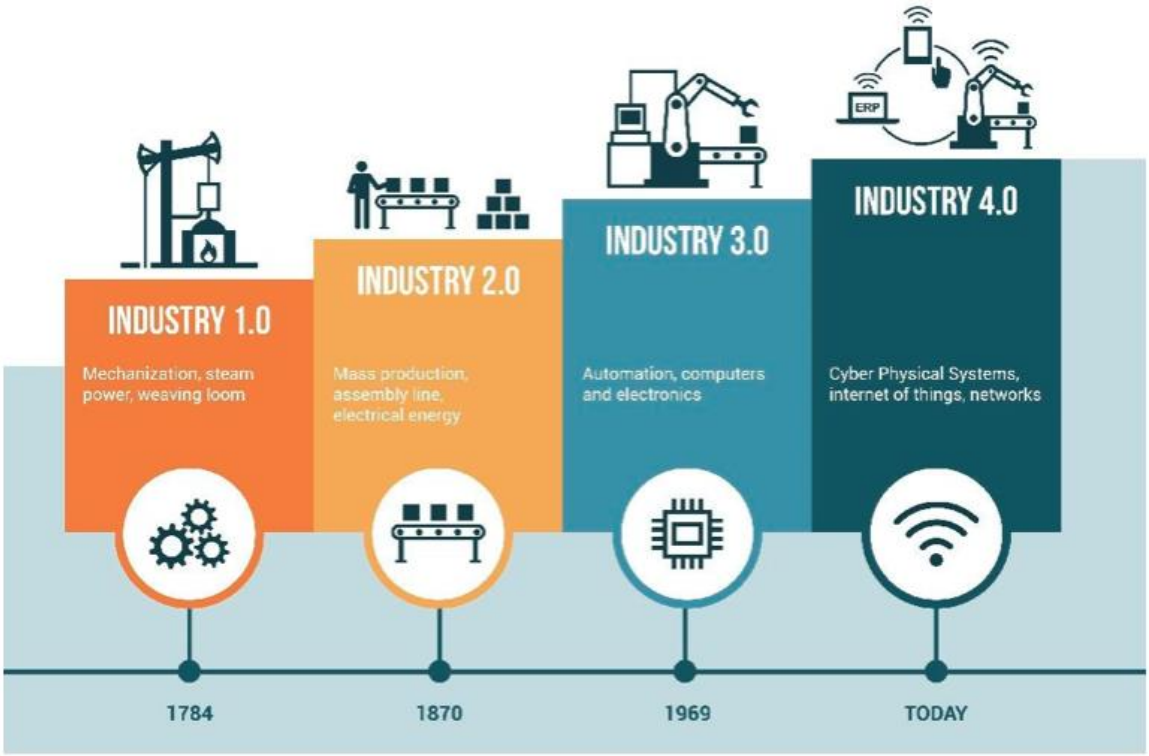
We are changing from an industrial economy to an information economy. Some call this an "algorithmic economy"[33]. This change offers unprecedented opportunities for interacting with customers in completely new ways and new business models that were never possible before. Changes like this, paradigm shifts really, are not new; we have seen them before. In our era we are changing from Industry 3.0 to Industry 4.0.

---

[31] Robert Kugel, *The Rising Expectations for Finance Analytics*, https://www.linkedin.com/pulse/rising-expectations-finance-analytics-robert-kugel/

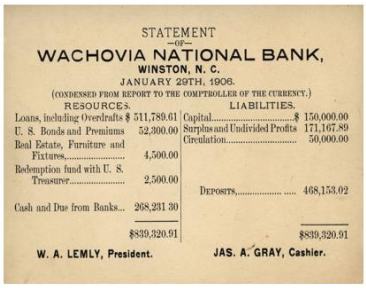
[32] Nordic Smart Government and Business, https://nordicsmartgovernment.org/

[33] The Business of Algorithms: Unveiling Its Global Economic Impact, https://www.linkedin.com/pulse/business-algorithms-unveiling-its-global-economic-impact-ko%C3%AFvogui/



The environment in which we will operate has changed. The three primary foundational changes are:

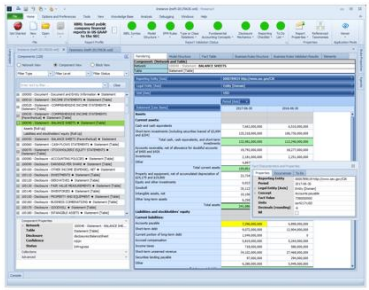
- structured machine-readable information in the form of knowledge graphs,
- artificial intelligence (deductive reasoning, inductive reasoning, abductive reasoning),
- publicly available and private digital distributed ledgers.



STATEMENT
—OF—
WACHOVIA NATIONAL BANK,
WINSTON, N. C.
JANUARY 29TH, 1906.
(CONDENSED FROM REPORT TO THE COMPTROLLER OF THE CURRENCY.)

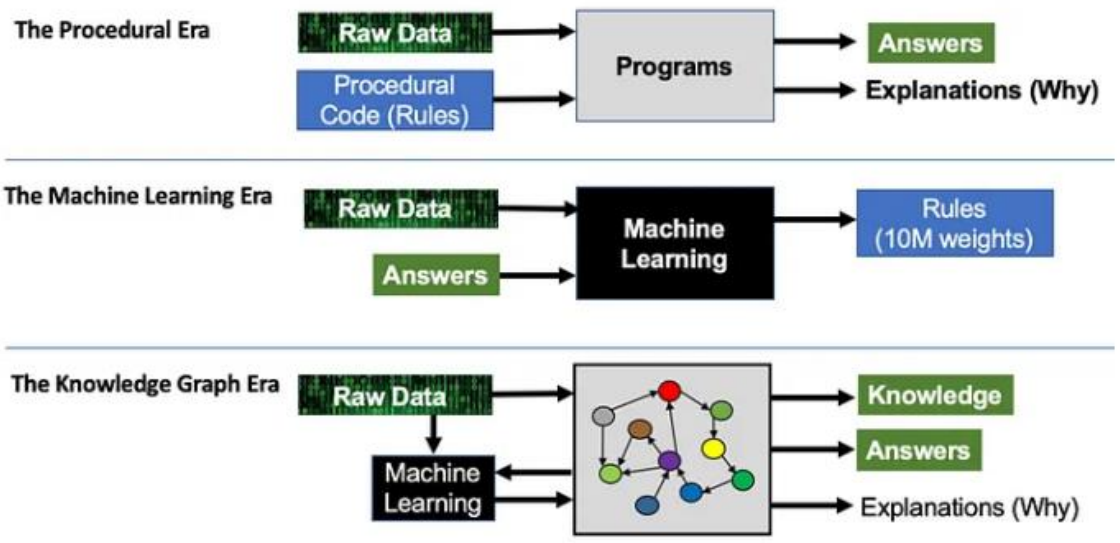
RESOURCES.		LIABILITIES.	
Loans, including Overdrafts	$ 511,789.61	Capital	$ 150,000.00
U. S. Bonds and Premiums	52,300.00	Surplus and Undivided Profits	171,167.89
Real Estate, Furniture and Fixtures	4,500.00	Circulation	50,000.00
Redemption fund with U. S. Treasurer	2,500.00	Deposits	468,153.02
Cash and Due from Banks	268,231.30		
	$839,320.91		$839,320.91

W. A. LEMLY, President. JAS. A. GRAY, Cashier.



As explained in the article, *Knowledge Graphs: The Third Era of Computing*[34], the era of the knowledge graph. This change from the procedural programming era to the era of knowledge graphs will have profound implications on how software is created and what software can be made to do.

[34] Dan McCreary, *Knowledge Graphs: The Third Era of Computing*, https://dmccreary.medium.com/knowledge-graphs-the-third-era-of-computing-a8106f343450



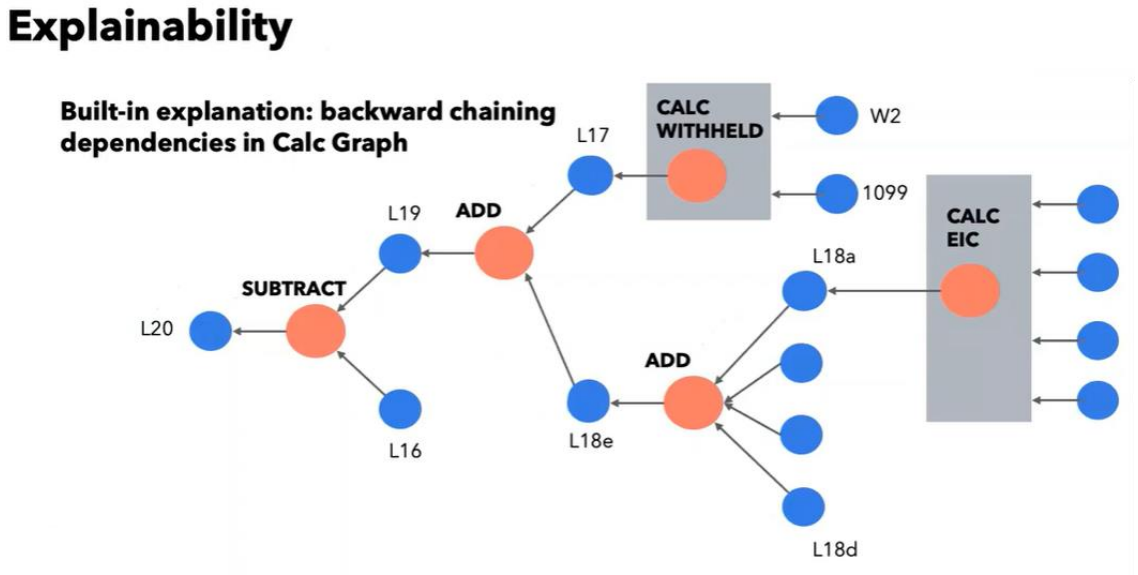
This change will move the logic followed by computer software from the software code written by software developers in the form of mainly "IF…THEN" statements effectively into what amounts to declarative rules[35] that are separated from software code and easier for business professionals to create and maintain.

	PROCEDURAL	KNOWLEDGE GRAPH
Logic	Procedural	Declarative
Execution	Sequential	Dependency-driven
Modularity	By best practice	By design
Explainability	Blackbox	Built-in
Personalization	Hard-coded, limited	Dynamic, automatic
Automation	Not manageable	Manageable
Testability	Harder	Easier
Optimized for	Exceptional complex logic	Most tax logic
Target Developer	Engineers	Domain experts

Knowledge represented in the form of natural language such as the pages in a book or a set of instructions can be hard to detect by humans. Providing this same information in the form of a diagram of this knowledge using simple graphics such as circles and lines can help the knowledge stand out. Further, these knowledge graphs tend to not only be better understandable by humans; the knowledge can also be understandable to computers. If a complete set of knowledge provided in this

[35] Business Rules Group, *The Business Rules Manifesto*, https://www.businessrulesgroup.org/brmanifesto.htm

manner and readable by computer processes, computers can be leveraged to check other information to see if that information is consistent with a provided set of knowledge, a.k.a. automated computer-based verification.



Finally, a new way of processing knowledge is possible when the actual knowledge is separated from the programming code, and "engine" or "rules processing engine" or "logic processing engine" can be created rather than computer programmers writing countless "IF…THEN" statements. Other benefits are opened up as will be explained.

1.11. Computational Professional Services

I refer to all this as *Computational Professional Services*[36]. (There might be a better term, but that is the term I am currently using).

Imagine a set of high-quality knowledge graphs organized into the form of a knowledge portal[37]. Imagine that the knowledge portal is enhanced by blockchain technology. Imagine that the knowledge graphs physical syntax is based on global standards and that the information within those knowledge graphs is also based on standards.

Imagine a system that is simple and elegant to use, rather than a poorly thought-out kludge.

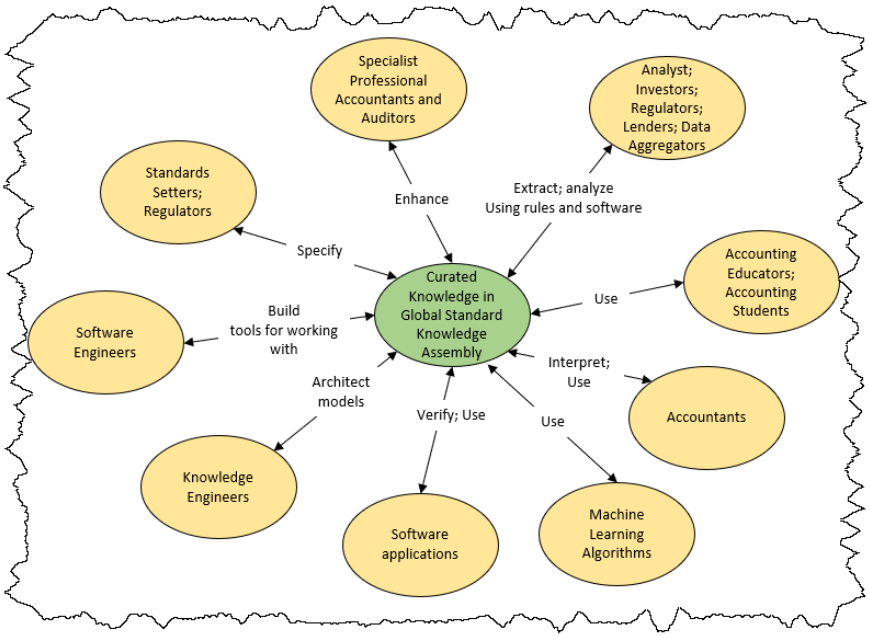
Trying to understand what is going on by trying to plug the changes that you see into the current paradigm of accounting, reporting, auditing, and analysis is like walking around the city of Chicago with a map of New York City to try and find your way. Using the appropriate map would work better. This document helps you update your mental map.

[36] *Computational Professional Services*, http://www.xbrlsite.com/mastering/Part00_Chapter01.A1_ComputationalProfessionalServices.pdf

[37] Data Science Central, Kurt Cagle, *From Knowledge Graphs To Knowledge Portals*, https://www.datasciencecentral.com/from-knowledge-graphs-to-knowledge-portals/

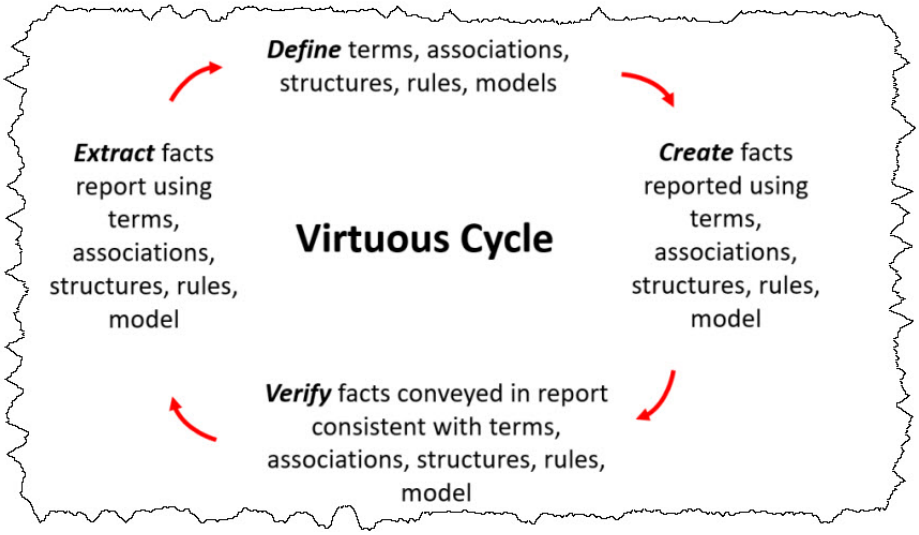
1.12. System Stakeholders

The system is not a collection of separate silos; rather the system is more like a chain or supply chain (stakeholders) of a system; curated machine-readable knowledge represented in standard format for an area of knowledge (stakeholders).



1.13. Virtuous Cycle

Orchestration is necessary. The system stakeholders need to agree on the goal(s) and objective(s) of the logical system. The high level objective of the system is to consciously and deliberately create a virtuous cycle as contrast to a vicious cycle:



1.14. Creation of a New Cottage Industry

Logical digital twins could be, and I predict will be, the foundation of a new cottage industry. That cottage industry will reorganize how financial accounting, reporting, auditing, and analysis work tasks and processes will be performed. Rather than dealing with the symptoms of the problem; logical digital twins will help solve the problem and the symptoms will go away. In this current system; accountants, auditors, and analysts are effectively working as data janitors fixing data quality problems, dealing with the aftermath of the problems. There is far too much "copy", "paste", "adjust" (CPA) involved in the work today. Tomorrow, skilled experienced professionals will work side-by-side with smart computer-based systems.

Think modern semantic spreadsheets[38], semantic accounting working papers and audit schedules[39] rather than presentation-oriented Excel spreadsheets, digital XBRL-based financial reports[40] as contrast to their presentation-oriented counterparts. Think intelligent software agents. Think reliable rules-based expert systems that are then supplemented by machine learning, transformers, LLMs, and copilots[41]. Think digital distributed ledgers. Think logical system.

2. Think Logical System

To understand the new paradigm, you have to change what you see when you think of a financial report and all the accounting working papers and audit schedules that support that report. A financial report is not a piece of paper. A financial report just looks like paper because that is the medium used to exchange the information that is represented on that financial report. Accountants just sprayed information onto that paper because that is the only tool they had. Before the paper accountants chiseled information onto clay tablets. Think not about the exchange medium; think about the information that is being exchanged. Not data; information.

This can be a hard transition, but others have flipped their thinking and so can you. Start by thinking about a financial report as a man-made logical system designed by humans.

2.1. Systems Thinking

Systems thinking[42] is the discipline of seeing wholes. Systems thinking is a framework for seeing logical interrelationships and logical patterns. Systems thinking is based on systems theory and is implemented using systems engineering. Another term for systems thinking is holistic thinking.

2.2. System

A system (a.k.a. formal system) is a cohesive conglomeration of interrelated and interdependent parts that is either natural or designed (man-made). A logical

[38] Modern Spreadsheet, https://digitalfinancialreporting.blogspot.com/2023/05/modern-spreadsheet.html
[39] Semantic Accounting and Auditing Working Papers, https://digitalfinancialreporting.blogspot.com/2023/05/semantic-accounting-and-auditing.html
[40] General Purpose Financial Reporting Support for XBRL, https://digitalfinancialreporting.blogspot.com/2023/02/general-purpose-financial-reporting.html
[41] Useful Generative AI Coming to Accounting and Reporting, https://digitalfinancialreporting.blogspot.com/2023/11/useful-generative-ai-coming-to.html
[42] *Systems Thinking*, https://digitalfinancialreporting.blogspot.com/2023/09/systems-thinking.html

system is a type of system. Systems are organized and have patterns. A system has a describable patterns of logical behavior. The notion of a system is explained by systems theory[43].

The solar system is a type of natural system. A general-purpose financial report is a designed system created by man to serve a very specific purpose.

Logical Systems for Business Professionals[44] helps accountants, auditors, and analysts think about financial report knowledge graphs and the general-purpose financial report logical system.

2.3. Logic

Logic[45] is a formal system/framework that defines the principles and rules of correct reasoning. Logical reasoning is about arriving at a conclusion in a rigorous way. There two broad categories of logical reasoning: deductive and non-deductive.

Deductive reasoning provides a result that is guaranteed to be certain, therefore the result can be relied upon without doubt and humans need not be involved in a process because of the certainty of deductive reasoning. Non-deductive reasoning, on the other hand, is not certain, meaning it could be correct but it could also be incorrect. Non-deductive reasoning is based on probability. And so non-deductive reasoning approaches must have a human in the loop to deal with that uncertainty. There are three types of non-deductive reasoning: inductive, abductive, and analogy.

Computational logic is a branch of logic and computer science that relates to getting computers to perform correct reasoning. Logic programming languages express facts using machine-readable logical statements, then use software to make sure all the facts are consistent and then draw inferences from these facts. To perform proper logical reasoning, one needs a complete set of logical statements.

That set of logical statements, preferably a complete set of logical statements, is a theory. Theories should be "precise" meaning that they describe an area of knowledge correctly, "consistent" meaning that there are no logical inconsistencies in the theory, and "complete" meaning that they fully represent the area of knowledge given the goals and objectives of the theory.

The tools of logic which provide the foundation for mathematics are leveraged by computers to mimic tools previously available only to humans, opening up the possibility of machines literally mimicking an understanding of knowledge. These tools can perform deductive reasoning, inductive reasoning, and abductive reasoning. Deductive, inductive, and abductive reasoning are different tools that bring different capabilities to the table. Hybrid tools which combine the capabilities of the three different types of logic can be created to maximize the utility of each tool within one combined system.

---

[43] *A Theory of a System for Educators and Managers*, https://youtu.be/2MJ3lGJ4OFo

[44] *Logical Systems for Business Professionals*, https://digitalfinancialreporting.blogspot.com/2023/09/logical-systems-for-business.html

[45] Wikipedia, *Logic*, https://en.wikipedia.org/wiki/Logic

2.4. Knowledge

Knowledge[46] is a form of familiarity with information from some specific area. Knowledge is often understood to be awareness of facts, having learned skills, or having gained experience using the things and the state of affairs (situations) within some area of knowledge. Professional accountants, auditors, and analysts understand all this for their area of knowledge, their industry specialty, for the financial reporting schemes with which they work.

Knowledge of facts is distinct from opinion or guesswork by virtue of justification or proof. Knowledge is objective. Opinions and guesswork are subjective. In our case we are talking about certain specific knowledge, the facts that make up that knowledge, being able to create a proof to show the logic of a knowledge graph system is complete, consistent, and precise; and all of this logic being put into a form readable by a machine and reach a conclusion.

Effectively, a machine can read that logic within a knowledge graph and mimic understanding of that knowledge represented in that logical knowledge graph and the information available to both a human reader and a machine reader would be the same and therefore the human and machine should reach the same conclusion. Regardless of the technical syntax used to represent that logical knowledge graph; the logic of the knowledge representation MUST always be the same no matter what technical format is used to represent the knowledge graph.

Philosophy is a formal discipline which provides tools and techniques for the systematic study of specific things including knowledge and reasoning.

2.5. Area of Knowledge

An **area of knowledge** is a highly organized socially constructed aggregation of shared knowledge for a distinct subject matter. An area of knowledge has a specialized insider vocabulary, underlying assumptions (axioms, theorems, constraints), and persistent open questions that have not necessarily been resolved (i.e. flexibility is necessary).

Accounting is an area of knowledge. You can explain aspects of the accounting area of knowledge, such as the nature of a financial report, using a logical theory which explains a logical model of that financial report. A logical theory can be tested and proven by providing a proof. When all the details are worked out, you have a best practice based proven method.

Knowledge can be represented in human-readable form, in machine-readable form, or in a machine-readable form that can be effectively and consistently converted into human-readable form.

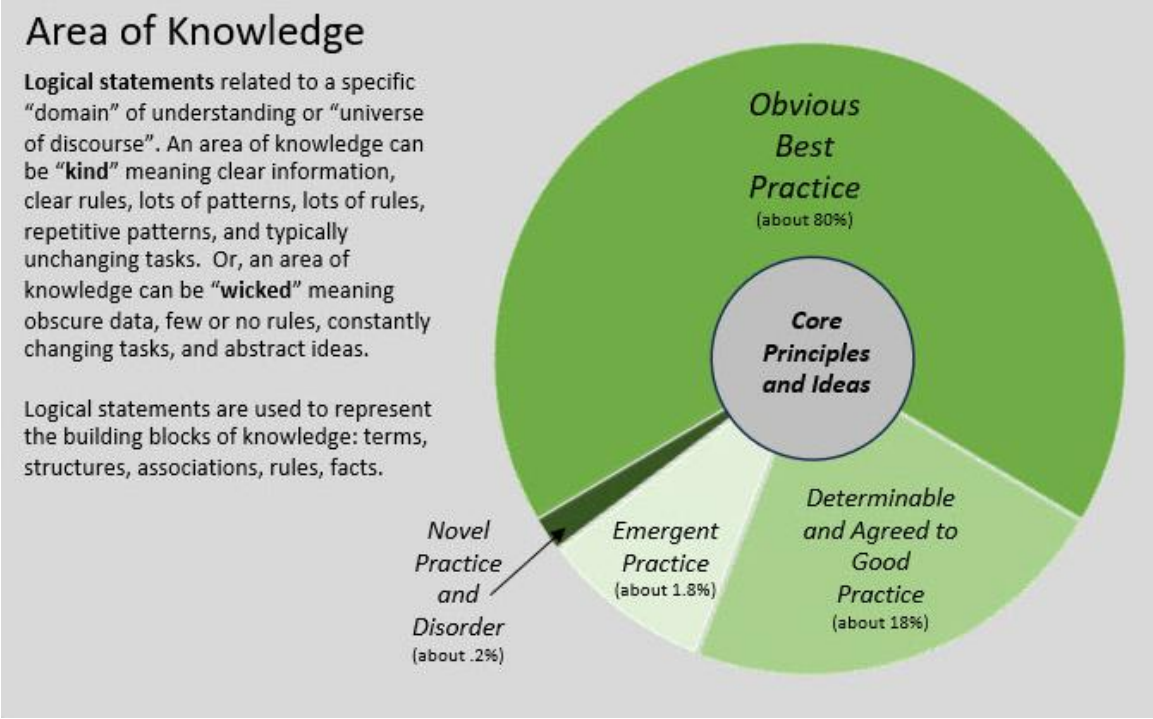
You can think about an area of knowledge as being characterized in a spectrum with two extremes:

- **Kind area of knowledge**: clear rules, lots of patterns, lots of rules, repetitive patterns, and unchanging tasks.
- **Wicked area of knowledge**: obscure data, few or no rules, constant change, and abstract ideas.

[46] *All About Knowledge*, https://digitalfinancialreporting.blogspot.com/2023/06/knowledge.html

An area of knowledge can have aspects of both extremes, but tends to lean toward one side of the spectrum or the other. Accounting tends to lean more toward the "kind" end.

Another term for area of knowledge is a knowledge domain, universe of discourse, or simply domain.



Accountants, auditors, and analysts need to make choices about how to organize their area of knowledge and sometimes they confuse what is driving the choice they make or why they have to make that choice. Accountants, auditors, and analysts need to be able to effectively differentiate:

- **Judgement**; not every choice relates to "professional judgement"; picking between permitted alternatives using their skills and experience
- **Ambiguity**; unintended ambiguity exists in accounting standards, just as ambiguity tends to exist in pretty much everything humans create; this is one reason why the tax code grows, and grows, and grows...
- **Skills and experience**; not every accountant has the same level of skill and experience in every practice area of financial accounting or reporting; most accountants are average; not every disclosure created in a financial report is an "art project"; less than 10% tends to be the art, 90% tends to be straight forward, mechanical, rote.

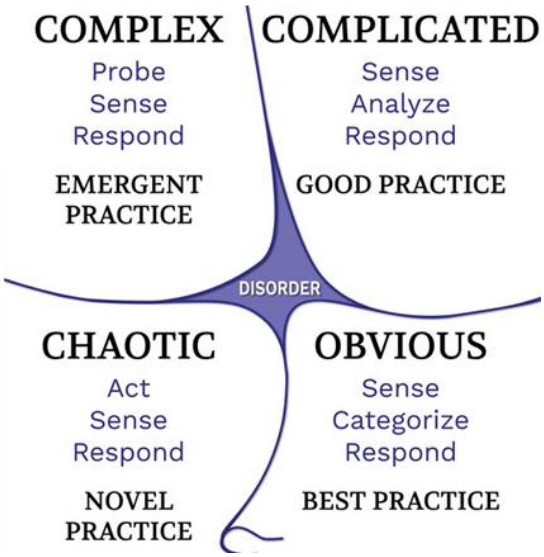
A system is collection of interacting components. An undisputed core of an area of knowledge should serve as a foundation upon which to then build. Only the subject matter experts of an area of knowledge can determine what is important and what is an unimportant trivial detail.

2.6. Complexity and Order

Difference systems have different levels of complexity. Systems can also be ordered or disordered. The *Cynefin Framework*[47] is a conceptual framework that helps you understand the dynamics that are at work within different types of systems. The framework was created in 1999 by David Snowden of IBM Global Services to help IBM to manage intellectual capital.

The following graphic helps one understand the different levels of complexity: simple, complicated, complex, and chaotic. The graphic also helps one understand the difference between disorder and order.

The video *Using Cynefin to Prioritize and Analyze Features, User Stories, and Functional Requirements*[48] provides an excellent walk through of these ideas.



Different skill sets are necessary to be able to create simple, complicated, and complex systems that work effectively.

Sensemaking is the process of determining the deeper meaning or significance or essence of the collective experience for those within an area of knowledge. The Cynefin Framework provides a tool for understanding knowledge. The Cynefin framework categorizes knowledge into the following groupings:

- **Best practices**: things that tend to be obvious even to people outside an area of knowledge. There tends to be only one way to do something which makes sense.
- **Good practices**: things that are a bit more complicated but the subject matter experts within an area of knowledge that have skills and experience tend to agree on these practices. Different groups can use different preferred good practice approaches as a matter of policies.

[47] Cynefin Framework, http://xbrl.squarespace.com/journal/2021/3/21/cynefin-framework.html

[48] Using Cynefin to Prioritize and Analyze Features, User Stories, and Functional Requirements, https://www.youtube.com/watch?v=L5fnxahydXM

- **Emergent practices**: things that are even more complex and subject matter experts within an area of knowledge tend to disagree with one another as to what the good practices are which leads to multiple different views, each which is reasonable based on the principles of an area of knowledge and the logical patterns of the situation. There tends to be tight, identifiable clusters of answers. (For example, if accounting standards have ambiguity and accountants apply fundamental principles to figuring out a situation and say each of the Big 4 CPA firms; PWC, Deloitte, EY, KPMG; each come up with a view on how to handle that situation; each view could be correct)
- **Novel practice**: this is similar to emergent practices except that there are no identifiable logical patterns of the situation and no identifiable principles that can be applied; but logical answers can be figured out but the clustering of answers is more spread out, not as tight.

Disorder of an area of knowledge is information not able to be ordered in any meaningful way.

2.7. Symbolic Systems

Stanford University has a unique undergraduate or graduate major offering called the *Symbolic Systems Program*[49].

So, what is a symbolic system? Per the associate director of the program when interviewed by The Stanford Daily[50]:

> "[The major is] a combination of studying the human mind … and the intelligence of machines and of the design interaction that happens between them, [as well as] how those things can inform each other," said symbolic systems associate director Todd Davies '84 M.S. '85 Ph.D. '95 in an interview with The Daily."

A symbol is something that represents something else. Symbols can be arranged into structures such as lists, hierarchies, or networks and these structures show how symbols relate to each other.

A **symbolic system** is essentially a system built with symbols such as natural language, programming languages, mathematics, or formal logic. Symbolic artificial intelligence[51] uses symbolic systems and rules to restrict systems to operate within permitted boundaries and to reason against the system.

An interesting thing is that symbolic systems are understandable by both humans and by computers.

You can get a more detailed understanding of symbolic systems from the Stanford Bulletin[52] which describes the course. Cognitive science[53] is somewhat similar to symbolic systems. Computational linguistics[54] is also somewhat similar.

---

[49] Stanford University, *Symbolic Systems Program*, https://symsys.stanford.edu/about/span-dig-deep-solve-complex-problems
[50] Stanford University, *The Stanford Daily*, https://www.stanforddaily.com/2019/01/23/unique-to-stanford-symbolic-systems/
[51] Wikipedia, *Symbolic Artificial Intelligence*, https://en.wikipedia.org/wiki/Symbolic_artificial_intelligence
[52] Stanford University, *Stanford Bulletin*, https://exploredegrees.stanford.edu/schoolofhumanitiesandsciences/symbolicsystems/
[53] Wikipedia, *Cognitive Science*, https://en.wikipedia.org/wiki/Cognitive_science

Why is this important?

2.8. Rearranging Abstract Symbols

In his book *Saving Capitalism*[55], Robert Reich describes three categories that all modern work/jobs fit into:

- **Routine production services** which entail 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."

Think Computational Law[56] and Computational Audit[57]. Many tasks in accounting, reporting, auditing, and analysis are related to symbolic-analytic services and rearranging abstract symbols. As I pointed out a while back, the "Learn to code" is a hysteria and is misguided. If you want to understand things like how artificial intelligence actually works and how it will impact accounting, reporting, auditing, and analysis; study symbolic systems and logic.

A logical system is a type of symbolic system. A logical theory can describe a logical system; for example, the *Logical Theory Describing Financial Report*[58] describes the financial report logical system. Not only is a financial report a type of logical system; that logical system can be readable and understandable by both humans and by machines.

2.9. Harnessing the Power of Computers

Important to understanding how to get computers to do what you want is understanding how computers actually work. The strengths of computers and the

---

[54] Wikipedia, *Computational Linguistics*, https://en.wikipedia.org/wiki/Computational_linguistics

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

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

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

[58] *Logical Theory Describing Financial Report*, http://www.xbrlsite.com/mastering/Part02_Chapter05.B_LogicalTheoryDescribingFinancialReport.pdf

obstacles that get in the way of using computers were summarized well by Andrew D. Spear[59]; here is his list with some modifications made by me:

Fundamental strengths/capabilities of computers:

- **store information** reliably and efficiently (tremendous amounts)
- **retrieve information** reliably and efficiently
- **process stored information** reliably and efficiently, mechanically repeating the same process over and over
- **instantly accessible information**, made available to individuals and more importantly other machine-based processes anywhere on the planet in real time

Major obstacles to harnessing the power of computers:

- **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 and store, retrieve, and process exactly the same information
- **inconsistent domain understanding** of and technology's limitations in expressing interconnections within a domain of knowledge
- **computers are dumb beasts**; computers don't understand themselves, the programs they run, or the information that they store, retrieve, process, or provide access to

Keep in mind that the information business professionals are trying to store, retrieve, process, access, and make use of is becoming more complex than what they have been storing, retrieving, processing, and accessing in relational databases or spreadsheets for the past 50 years. For example, a financial report is complex information that is very challenging to store in a relational database and then query across millions of such reports efficiently.

2.10. Knowledge Based Systems (Artificial Intelligence)

The document *Expert System for Creating Financial Reports*[60] outlines the vision of a rules-based expert system for creating financial reports that leverages explainable artificial intelligence (XAI)[61]. This vision is based on working software applications, working proof of concepts, reverse engineering of XBRL-based financial reports submitted to the SEC.

---

[59] Andrew D. Spear in his document, *Ontology for the Twenty First Century: An Introduction with Recommendations*, page 4

[60] Charles Hoffman, CPA, *Expert System for Creating Financial Reports*, http://xbrlsite.azurewebsites.net/2022/Library/ExpertSystemForCreatingFinancialReports.pdf

[61] *Effective Automated Information Exchange and Explainable AI (XAI)*, http://xbrl.squarespace.com/journal/2022/3/21/effective-automated-information-exchange-and-explainable-ai.html

There are two approaches to **artificial intelligence** and, as I have said before, the right approach should be used for the given job[62]. The two approaches are:

- **Rules-based systems** (symbolic systems, good at reasoning, expert systems, three basic types)
  - **Classification or diagnosis type**: helps users of the system select from a set of given alternatives.
  - **Construction type**: helps users of the system assemble something from given primitive components.
  - **Simulation type**: helps users of the system understand how some model reacts to certain inputs.
- **Patterns-based systems** (non-symbolic systems, good at learning, machine learning which can be supervised or unsupervised, five basic types[63])
  - **Clustering algorithms**: categorize or group things
  - **Explanatory algorithms**: explain the relationships between variables
  - **Ensemble learning algorithms**: use multiple models
  - **Similarity algorithms**: compute the similarity of pairs of things
  - **Dimensionality reduction algorithms**: reduces variables in a dataset

Another helpful breakdown of pattern-based systems is provided by the article *Top Machine Learning Algorithms for Prediction: A Short Summary*[64].

Name	Type	Description	Advantages	Disadvantages
Linear Regression		-The best fit line through all data points	-Easy to understand -you can clearly see what the biggest drivers odf the model are.	-sometimes to simple to capture cpmöex relationships between variables, -Tendency für the model to overfit.
Logistic Regression		-The adoption for linear regression to problembs of classification	-Easy to understand	-sometimes to simple to capture cpmöex relationships between variables, -Tendency für the model to overfit.
Decision Tree		-A graph that uses branching method to match all possible outcomes of a decision	-Easy to understand and implement.	-Not often use of ist own for prediction because it's also often too simple and not powerful enough for complex data.
Random Forest		- Takes the average of many decision trees. Each tree is weaker than the full decision tree, but combining them we get better overall performance.	-A sort of „wisdom of the crowd", Tend to result in very high quality results. -Fast to train	-Can be slow to output predictions relative to other algorithms. -Not easy to understand predictions.
Gradient Boosting		-Uses even weaker decision trees that increasingly focused on „hard examples"	-High-performing	-A small change in the future set or training set can create radical changes in the model. -Not easy to understand predictions.
Neural Networks		-Mimics the behaviour of the brain. NNs are interconnected Neurons that pass messages to each other. Deep Learning uses severak layers of NNs to put one after the other.	-Can handle extremely complex tasks. No other alsgorithm comes close in image recognition.	-very very slow to train. Because they have so many layers. Require a lot of power. -Almost impossible to understand predictions.

[62] *Use the Right Artificial Intelligence Approach for the Job*, http://xbrl.squarespace.com/journal/2019/7/12/use-the-right-artificial-intelligence-approach-for-the-job.html

[63] EDUCBA, Machine Learning Models, https://www.educba.com/machine-learning-models/

[64] Medium, *Top Machine Learning Algorithms for Prediction: A Short Summary*, https://medium.com/@webadmin_46735/top-machine-learning-algorithms-for-predictions-a-short-overview-5ed1ff6942ff

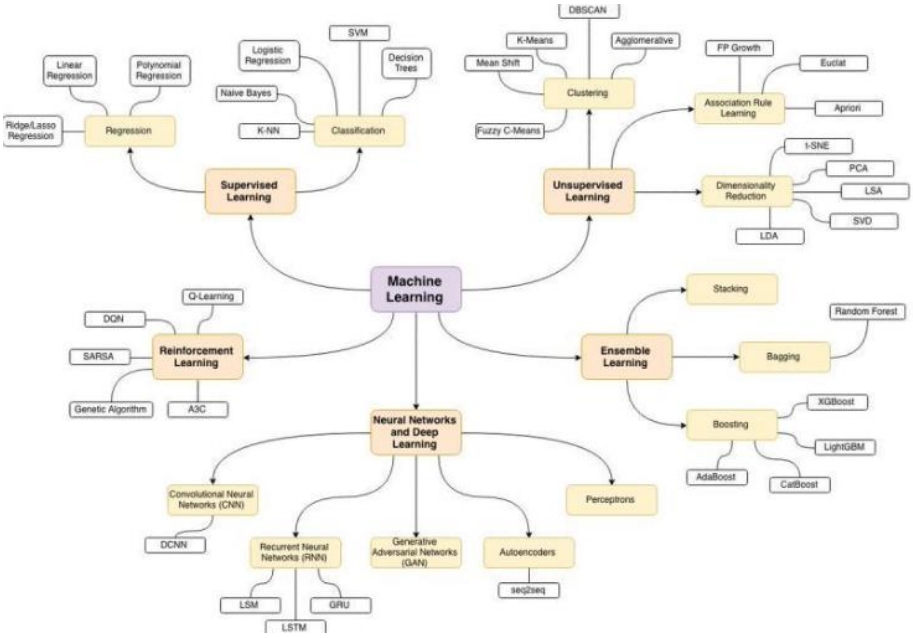
Currently, the expert systems working proof of concepts and commercial software that I have created and experimented with only leverage rules-based systems. But it is pretty clear that machine learning offers additional capabilities that will be inevitably leveraged.

Pacioli[65] is a logic/rules/reasoning/knowledge/insights engine. Frankly, I don't know exactly what to call Pacioli. At first, I called Pacioli a logic/rules/reasoning engine. Then I referred to it as a knowledge engine[66]. Now I am contemplating that Pacioli is an insights engine. For now, I consider Pacioli a logic/reasoning/rules/knowledge/insights engine in order to be complete, I guess.

Pacioli has three roles. The first role of Pacioli is to enforce the financial report logical schema and return information about where a financial report might be violating that logical schema. This is helpful in the process of creating XBRL-based financial reports.

The second role of Pacioli is to leverage the machine-readable logical schema to work with the information that makes up the financial report knowledge graph[67].

The third role of Pacioli, which similarly leverages the machine-readable logical schema and financial report knowledge graph is effectively extracting information for financial analysis.



Many of these ideas are incorporated into Auditchain Luca[68] which is, as far as I can tell, the world's first expert system for creating financial reports using global standards.

---

[65] Auditchain, *Pacioli Logic and Rules Engine*, https://docs.auditchain.finance/auditchain-protocol/pacioli-logic-and-rules-engine

[66] *Pacioli: an XBRL Knowledge Engine*, http://xbrl.squarespace.com/journal/2022/2/19/pacioli-an-xbrl-knowledge-engine.html

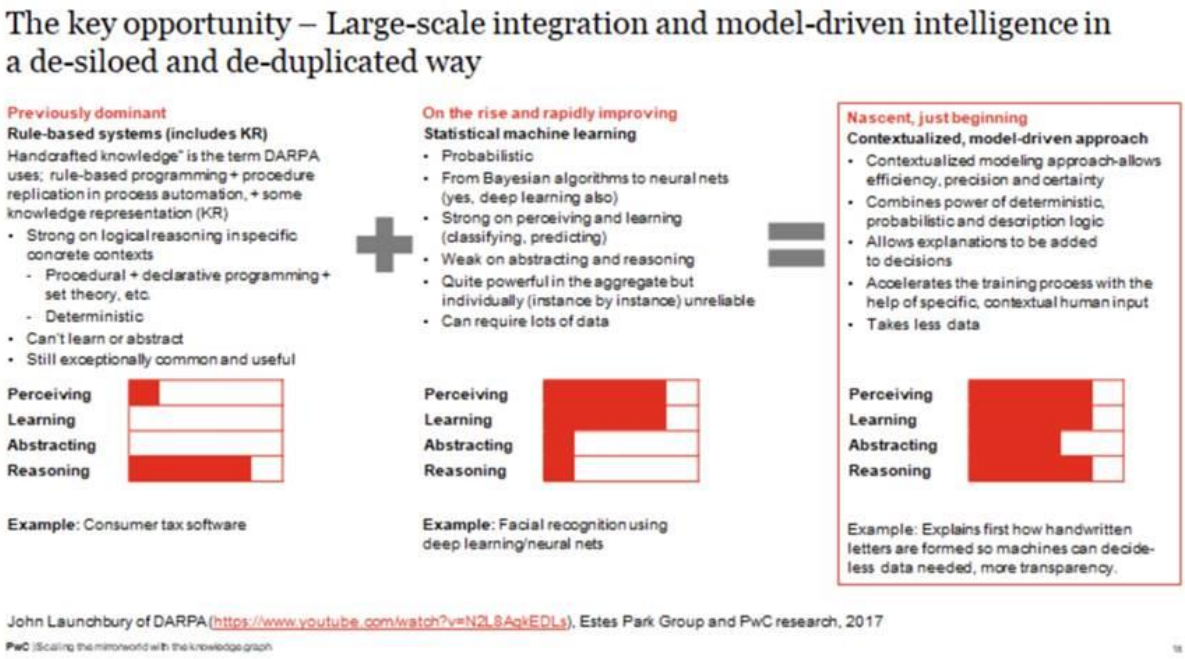
[67] Charles Hoffman, CPA, *Financial Report Knowledge Graphs*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf

[68] *Getting Started with Auditchain Luca*, https://digitalfinancialreporting.blogspot.com/2024/01/getting-started-with-auditchain-luca.html

2.11. Programmed Ability to Process Information

Both symbolic systems and non-symbolic systems offer useful capabilities[69]. Be sure to understand the capabilities of these different approaches and use the right tool for the job.

DARPA and PWC do an excellent job of explaining the capabilities of artificial intelligence. This presentation[70], video[71] and this article[72] provide a summary worth reading.



The graphic above shows three approaches to implementing artificial intelligence or the programmed ability to process information. Auditchain's Pacioli is a "rule-based system" that uses "handcrafted knowledge" that I created (the LEFT SIDE). Statistical machine learning will be added (the MIDDLE) to Pacioli over time most likely. Ultimately, where Auditchain and others will end up is on the RIGHT side which is the combined capabilities of rules-based systems and statistical machine learning. I suspect that this will occur within three to five years.

Here is another explanation that was inspired from a DARPA presentation[73]. Artificial intelligence is defined as a "**programmed ability to process information**". Sure, software like, Excel and Word can process "stuff"; but they don't really understand anything about accounting, reporting, auditing, or analysis.

---

[69] Richa Bhatia, *Understanding the difference between Symbolic AI & Non Symbolic AI*, https://analyticsindiamag.com/understanding-difference-symbolic-ai-non-symbolic-ai/

[70] DARPA, *A DARPA Perspective on Artificial Intelligence*, https://www.darpa.mil/attachments/AIFull.pdf

[71] YouTube.com, *A DARPA Perspective on Artificial Intelligence*, https://youtu.be/-O01G3tSYpU

[72] Quora, Alan Morrison, *What is the relation between Semantic Web and AI?*, https://www.quora.com/What-is-the-relation-between-Semantic-Web-and-AI/answer/Alan-Morrison?ch=2&oid=180785119&srid=Mru&target_type=answer

[73] DARPA, *A DARPA Perspective on Artificial Intelligence*, https://www.darpa.mil/attachments/AIFull.pdf

By "understand", this is what is meant, paraphrasing form the DARPA explanations of these terms:

- **Perceive**: ability to perceive rich, complex, subtle information.
- **Learn**: ability to learn about that information within a specific environment such as within a financial report.
- **Abstract**: ability to create new meaning from existing information.
- **Reason**: ability to plan, ability to decide, ability to verify.

The following graph shows where most software currently is today in terms of helping accountants get their jobs done in terms of accounting knowledge (i.e. basically, you can think of this is the manual tasks and processes that is being performed now)

But this is where something like Auditchain's Pacioli is in terms of perceiving, learning, abstracting, reasoning when it comes to the information within a financial report: (Pacioli can do this NOW using my "handcrafted knowledge" a.k.a. standard machine-readable rules represented using XBRL)

The next graphic shows where Pacioli might ultimately go when statistical machine learning is combined with the rules-based system. The rules-based system's handcrafted rules act as a starter set of the rules needed to make the statistical machine learning work. Basically, the rules-based system focuses on what it can do best and the statistical machine learning focuses on what it can do best and the result is a combination of the two approaches: (Auditchain may be here in three to five years in my view)

The final graphic is what full automation looks like. To understand what "full automation" is one would need to understand the specific details of what is being automated. But also keep in mind that while automation might be possible for some things, for other things it really is not realistic or even necessary to automate to derive value from automation. Making a process 50% more efficient or even 25% more efficient or only 10% more efficient is still quite valuable.

To summarize, there are three very important points to remember here. First, the right artificial intelligence should be used for each automation task. Second, handcrafted metadata is necessary to make automation actually work. Finally, while it is easy to discuss automation in general high-level terms; when you actually automate something you need to work with the details of what is being automated.

2.12. Important to Use the Right Tool for the Job

Machine learning or deep learning systems work best if the system you are using them to model has a high tolerance to error. These types of systems work best for:

- capturing associations or discovering regularities within a set of patterns;
- where the volume, number of variables or diversity of the data is very great;
- relationships between variables are vaguely understood; or,
- relationships are difficult to describe adequately with conventional approaches.

Machine learning basically uses probability and statistics, correlations[74]. This is not to say that machine learning is a bad thing. It is not, machine learning is a tool. Any craftsman knows that you need to use the right tool for the job. Using the wrong tool will leave you unsatisfied[75]. Ultimately, what you create will either work or it will not work to achieve your objectives.

2.13. Knowledge Acquisition

There are no short cuts. No one really disputes the need for models and a "thick layer of metadata" to get a computer to perform work. What is disputed is the best way to get that thick layer of metadata and those models. Machine learning works best if you already have a thick layer of metadata, that is the training data that machine learning needs to work.

Fundamentally, there are three approaches to acquiring knowledge:

- **Humans create the knowledge**: this tends to be expensive and time consuming.
- **Machines create the knowledge**: this tends to be significantly cheaper, but the quality is significantly lower.
- **Hybrid approach using both humans and machines**: humans and machines working together leverages the benefits of each approach.

2.14. Models

Without a model you have an undifferentiated mass of data where there is no way which data can or should interact with other data. A model has little or no significance without the data to describe that model. But put the data and the model together and you get a dynamic web of information that seems almost magical.

---

[74] Kalev Leetaru, *Our Entire AI Revolution Is Built On A Correlation House Of Cards*, https://www.forbes.com/sites/kalevleetaru/2019/04/20/our-entire-ai-revolution-is-built-on-a-correlation-house-of-cards/

[75] Gil Press, *This Week In AI Stats: Up To 50% Failure Rate In 25% Of Enterprises Deploying AI*, Forbes, https://www.forbes.com/sites/gilpress/2019/07/19/this-week-in-ai-stats-up-to-50-failure-rate-in-25-of-enterprises-deploying-ai/

Any large group trying to understand a complex phenomenon will find that process challenging. Models helps areas of knowledge make sense of unorganized information.

A model is an abstract description of something that hides certain unimportant details and illuminates other important details. Models help a group trying to describe an area of knowledge to do so effectively. Models do achieve this in the following important ways:

1. Models help groups of people in an area of knowledge to communicate.
2. Models describe, explain, and make predictions about that area of knowledge.
3. Models describe primitive phenomenon. Those primitive descriptions can then be used to describe complex phenomenon that is based on the primitive descriptions.
4. Models provide a way to orchestrate and mediate when you have multiple viewpoints. It is hard enough to get any two people to agree completely agree about something, let alone a significantly larger group. Models help groups represent commonalities while also understanding and exploring their differences.

Formal models are used in many different areas of knowledge when precision and objectivity are necessary. Models are not "black-and-white". Models organize human thought. Good models offer flexibility where flexibility is necessary.

A model is effectively a logical system. Let's walk you through a simple explanation of a logical system and how you can use a theory to describe such a system.

2.15. Metamodel

A metamodel is a model of a model. A financial reporting example will explain. Suppose that Microsoft represented its financial report in the form of a model. Now, suppose that Apple represented its financial report in the form of a model also. But what happens if Microsoft and Apple used different models to represent each of their financial reports?

That is where a metamodel comes in.

A regulator like the U.S. Securities and Exchange Commission (SEC) or the European Single Market Authority (ESMA) define a metamodel that the report models of companies submitting reports to the regulators must follow. That metamodel helps to make sure that the report model of each company report is consistent.

Could the SEC and ESMA use different metamodels? Sure. But also, could the SEC and ESMA use the same metamodel? Sure, that is possible too. Effectively, what the *Seattle Method* and *Standard Business Report Model* (SBRM) do is specify a metamodel that is consistent for both the SEC and ESMA XBRL-based digital financial reports.

2.16. SBRM and Seattle Method

The Object Management Group (OMG) is taking XBRL-based business reports to a new level, leveraging what has been learned from creating XBRL-based financial

reports over the past 10 years. The *Standard Business Report Model* (SBRM)[76] is described as follows:

> "SBRM formally documents a logical conceptualization of a business report in both human-readable and machine-readable models."

SBRM goes on to explain that through the use of standard models, business experts can define the structure and content of their reports and extensions using high-level logical business report objects, possibly presented in the form of semantic spreadsheets and pivot tables rather than with lower level technical syntax.

While XBRL has mainly been employed for financial reporting, leveraging the nature of financial accounting rules[77]; digital business reporting will benefit from the capabilities pioneered by XBRL-based financial reporting. Further, business reporting will not be limited to only one syntax but rather the arbitrary preferred syntax of can be used and systems can still be consistent with one conceptual model of a business report.

Financial reporting will likewise benefit from SBRM because SBRM helps business professionals and technical professionals constructing systems where flexible reporting is a requirement to effectively control variability and still have high-quality information exchanges.

SBRM is based on the ideas of the *Seattle Method*[78]. The *Seattle Method* is a proven, industrial strength, good practices based, standards-based pragmatic approach to creating provably high quality XBRL-based general purpose financial reports that builds on the Venetian Method of double entry bookkeeping and adapting it for the information age explained in simple terms approachable by business professionals.

2.17. Simple Explanation of a Logical Systems

A system can be explained by a logical theory. A logical theory is an abstract conceptualization[79] of specific important details of some area of knowledge. The logical theory provides a way of thinking about an area of knowledge by means of deductive reasoning to derive logical consequences of the logical theory.

A **logical theory** enables a community of stakeholders trying to achieve a specific goal or objective or a range of goals/objectives to agree on important logical statements used for capturing meaning or representing a shared understanding of and knowledge in some area of knowledge.

A logical theory forms a **logical conceptualization** and is made up of a set of logical *models*, *structures*, *terms*, *associations*, *rules*, and *facts*. In very simple terms,

- **Logical conceptualization**: A *logical conceptualization* is a set of models that are consistent with and permissible per that logical conceptualization.

---

[76] *Object Management Group and Standard Business Report Model (SBRM)*, http://xbrl.squarespace.com/journal/2019/6/25/object-management-group-and-the-standard-business-report-mod.html

[77] Charles Hoffman, *Leveraging the Theoretical and Mathematical Underpinnings of a Financial Report*, http://xbrlsite.azurewebsites.net/2018/Library/TheoreticalAndMathematicalUnderpinningsOfFinancialReport.pdf

[78] Seattle Method, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf

[79] Wikipedia, *Conceptual Model*, https://en.wikipedia.org/wiki/Conceptual_model

- **Model**: A *model*[80] is a set of logical structures that are consistent with and permissible interpretations of that logical conceptualization.
- **Structure**: A *structure* is a set of logical statements which describe the structure.
- **Logical statement**: A *logical statement* is a proposition, claim, assertion, belief, idea, or fact about or related to the area of knowledge to which the logical conceptualization relates. There are five broad categories of logical statements:
    - **Terms**: *Terms* are logical statements that define ideas used by the logical theory such as "assets", "liabilities", "equity", and "balance sheet".
    - **Associations**: *Associations* are logical statements that describe permissible interrelationships between the terms such as "assets is part-of the balance sheet" or "operating expenses is a type-of expense" or "assets = liabilities + equity" or "an asset is a 'debit' and is 'as of' a specific point in time and is always a monetary numeric value".
    - **Rules**: *Rules* are logical statements that describe what tend to be convertible into IF…THEN…ELSE types of relationships such as "IF the economic entity is a not-for-profit THEN net assets = assets - liabilities; ELSE assets = liabilities + equity".
    - **Facts**: *Facts* are logical statements about the numbers and words that are provided by an economic entity within a business report. For example, the financial report, a type of business report, might state "assets for the consolidated legal entity Microsoft as of June 20, 2017 was $241,086,000,000 expressed in US dollars and rounded to the nearest millions of dollars.
    - **Properties**: *Properties* are logical statements about the important qualities and traits of a model, structure, term, association, rule, or fact.

Fundamentally, a logical theory is a set of logical statements. Those logical statements can be represented in human-readable form or they could be expressed in machine-readable form. Once in machine-readable form, those logical statements can be interrogated using software applications. To the extent that this can be performed effectively; software tools can assist professional accountants, financial analysts, and others working with those logical statements.

2.18. Proper Functioning Logical System

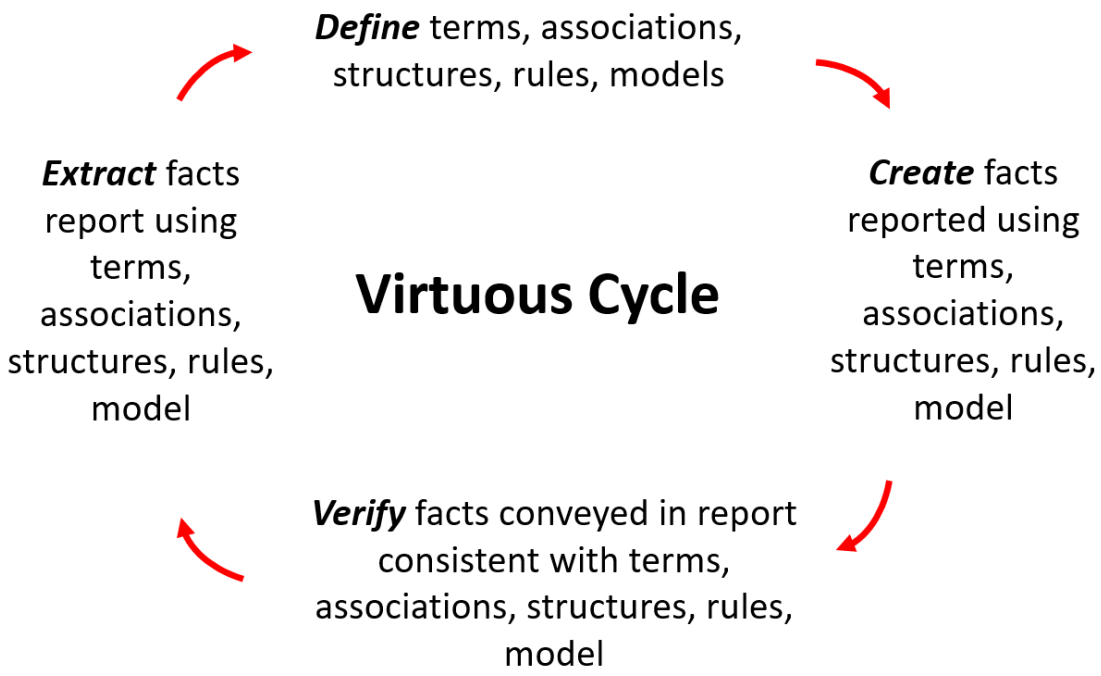
A logical system is said to be **consistent** with a logical theory if there are no contradictions with respect to the logical statements made by the logical theory that describes the logical system.

A logical theory can have high to low **precision** and high to low **coverage** with respect to describing a logical system.

[80] Wikipedia, *Model Theory*, https://en.wikipedia.org/wiki/Model_theory

Precision is a measure of how precisely the information within a logical theory has been represented as contrast to reality of the logical system for the area of knowledge. *Coverage* is a measure of how completely information in a logical theory has been represented relative to the reality of the logical system for the area of knowledge.

When a logical system is consistent and it has high precision and high coverage the logical system can be considered a **properly functioning logical system**. When a logical system is properly functioning, it creates a virtuous cycle[81].



A logical theory conveys knowledge and that knowledge can be represented within a knowledge graph.

2.19. Very Simple Example of Logical System

A very simple example of a logical system is the accounting equation[82]. Here is a description of the accounting equation logical system in both human-readable terms and machine-readable terms using XBRL[83]:

Terms: Three simple terms are defined: Assets, Liabilities, Equity. One complex term is defined, balance sheet.

Structure: One structure is defined, the balance sheet, and identified using the term balance sheet.

Associations: The three terms Assets, Liabilities, and Equity are associated in that they are all PART-OF the structure balance sheet.

---

[81] Charles Hoffman, CPA, *Virtuous Cycle*, http://xbrl.squarespace.com/journal/2020/4/29/virtuous-cycle.html

[82] Wikipedia, *Accounting Equation*, https://en.wikipedia.org/wiki/Accounting_equation

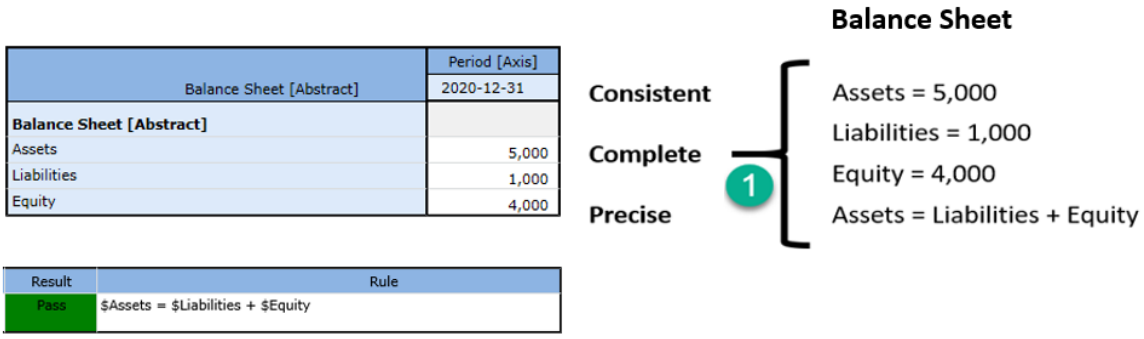
[83] Charles Hoffman, *Accounting Equation*, http://xbrlsite.azurewebsites.net/2020/master/ae/

Rules: A mathematical assertion is made that "Assets = Liabilities + Equity".

Facts: Instances of three facts are established to exercise the model: Assets of $5,000; Liabilities of $1,000; Equity of $4,000.

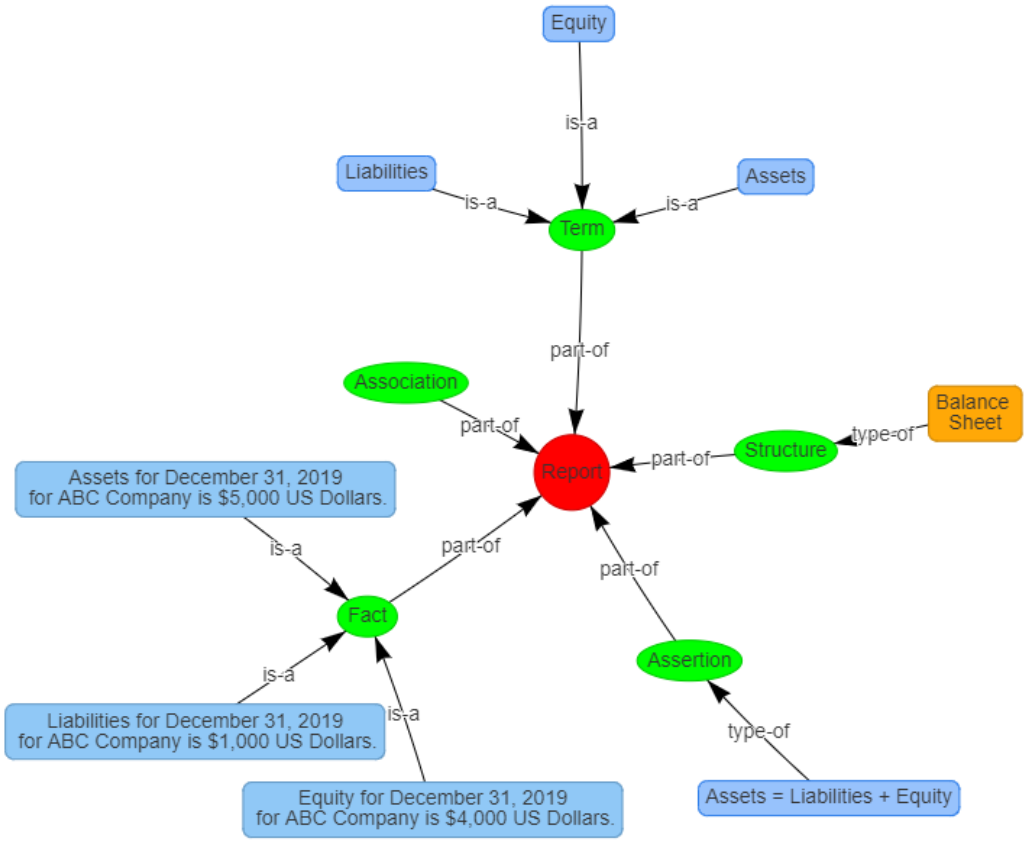
Model: All of the terms, associations, assertions, structures, and facts describe the model. We created only one model, or permissible interpretation, of the logical theory. (As accountants know, if you reverse the equation using the rules of math to "Equity = Assets - Liabilities" and change the term "Equity" to "Net Assets"; then you get another permissible interpretation or model.)

Because this is a very simple example with only a few statements it is easy to get your head around this system and see that it is consistent, complete, and precise. As expected, you see three facts described by three terms which are related to one structure and the one rule is consistent with expectation:



As the size of the logical system increases it becomes increasingly more challenging to verify that the logical system is properly function using manual processes. But, covering the impediments to a properly functioning logical system are beyond our scope here[84]. Essentially, the models, terms, structures, rules, and facts form a directed acyclic graph such as:

[84] Charles Hoffman, CPA, *Impediments to Creating Properly Functioning XBRL-based Reports*, http://xbrlsite.azurewebsites.net/2020/core/master-ae/Documentation.pdf



While a typical financial report is significantly larger (i.e. the Microsoft 2017 10-K is made up of 194 structures) every financial report works the same as this very simple example but just has more pieces[85].

2.20. Levels of Financial Report

Not every machine representation of a report is equal. This section explains and contrasts the different levels[86] of a digital financial report starting with level 0 which is a physical report.

1. **Level 0 (Information provided physically)**: Not machine readable. An example of Level 0 is a clay tablet, papyrus, or paper as the report medium. (Example, imagine that this example is on paper.)
2. **Level 1 (Information provided digitally)**: Machine readable, structured for presentation of the information, information metadata is nonstandard. (Examples: PDF, HTML, JPEG, and other forms of e-paper.
3. **Level 2 (Information provided digitally, structured for meaning, nonstandard)**: Machine readable, structured for meaning, nonstandard, no taxonomy, no rules, no report model. (Example, an XML document; no model explaining report information.

[85] *Analysis of 6,751 XBRL-based Public Company 10-Ks Submitted to SEC*, http://www.xbrlsite.com/mastering/Part05_Chapter08.F_AnalysisOf675110Ks.pdf
[86] Levels of Financial Report, https://digitalfinancialreporting.blogspot.com/2024/01/levels-of-report.html

4. **Level 3 (Global standard report model and report syntax structured for meaning)**: Machine readable, global standard syntax, structured for meaning, with report model which explains some of the report (e.g. incomplete), and the terms used are not from a published common financial reporting scheme. (Example, XBRL-based report model and report but the report defines all report information locally in the report.)
5. **Level 4 (Common dictionary of terms, structures, associations, rules)**: Machine readable, global standard syntax, structured for meaning, with report model, complete set of rules provided, incomplete high-level report model. (Example, an XBRL-based report with a XBRL taxonomy schema, with XBRL relations and resources, and with XBRL Formulas that completely describes the report.)
6. **LOGICAL TWIN - Level 5 (Complete set of logical statements)**: Machine readable, global standard syntax, structured for meaning, with taxonomy, complete set of rules provided, complete global standard high-level report model; yields verifiably proven properly functioning system and understandable report information. (Example, an XBRL-based report with all the characteristics of Level 4, plus consistency cross checks, type-subtype relations, consistent and logical modeling of XBRL presentation relations, information that describes the correct representation of every disclosure within the report, and a reporting checklist that describes all required disclosures; **verified to be a properly functioning logical system**.)
7. **Level 6 (Trust report logic not manipulated)**: All of Level 5 PLUS blockchain-anchored XBRL to increase trust. An XBRL-based report with all the characteristics of Level 5, plus information in the form of a Merkle hash within a digital distributed ledger that assures no one has tampered with the report. (No example to show; here is somewhat of a prototype)
8. **Level 7 (Trust report transactions provenance)**: All of Level 6 PLUS blockchain-anchored accounting transactions and events. An XBRL-based report with all the characteristics of Level 6, plus information in the form of a Merkle hash that indicates that assures no one has tampered with transactions. (No example to show)

2.21. Semantic Glue

Effectively, what a logical digital twin provides is the "semantic glue[87]" that hooks the pieces of information within the logical digital twin together and also lets you reference that logical information. Rather than referencing a position in a spreadsheet, you reference information based on the information logic of that information.

2.22. Complexity

A kluge is a term from the engineering and computer science world that refers to something that is convoluted and messy but gets the job done. Elegance is the quality of being pleasingly ingenious, simple, neat. Elegance is about beating down

[87] *Understanding and Leveraging the "Semantic Glue" of XBRL-based Financial Reports*, http://xbrlsite.com/2024/Library/UnderstandingAndLeveragingSemanticGlue.pdf

complexity. Creating something complex is easy. Creating something simple and elegant is hard work.

The *Law of Conservation of Complexity* points out,

> "Every application has an inherent amount of irreducible complexity. The question is who will have to deal with that complexity: the application user, the application developer, or the platform developer that the application runs on."

Complexity cannot be removed from a system; but it can be moved. The terms "simplistic" and "simple" describe different things. Simplistic is dumbing down a problem in order to make the problem easier to solve. Simple is something that is not complicated, that is easy to understand or do. Simple means without complications.

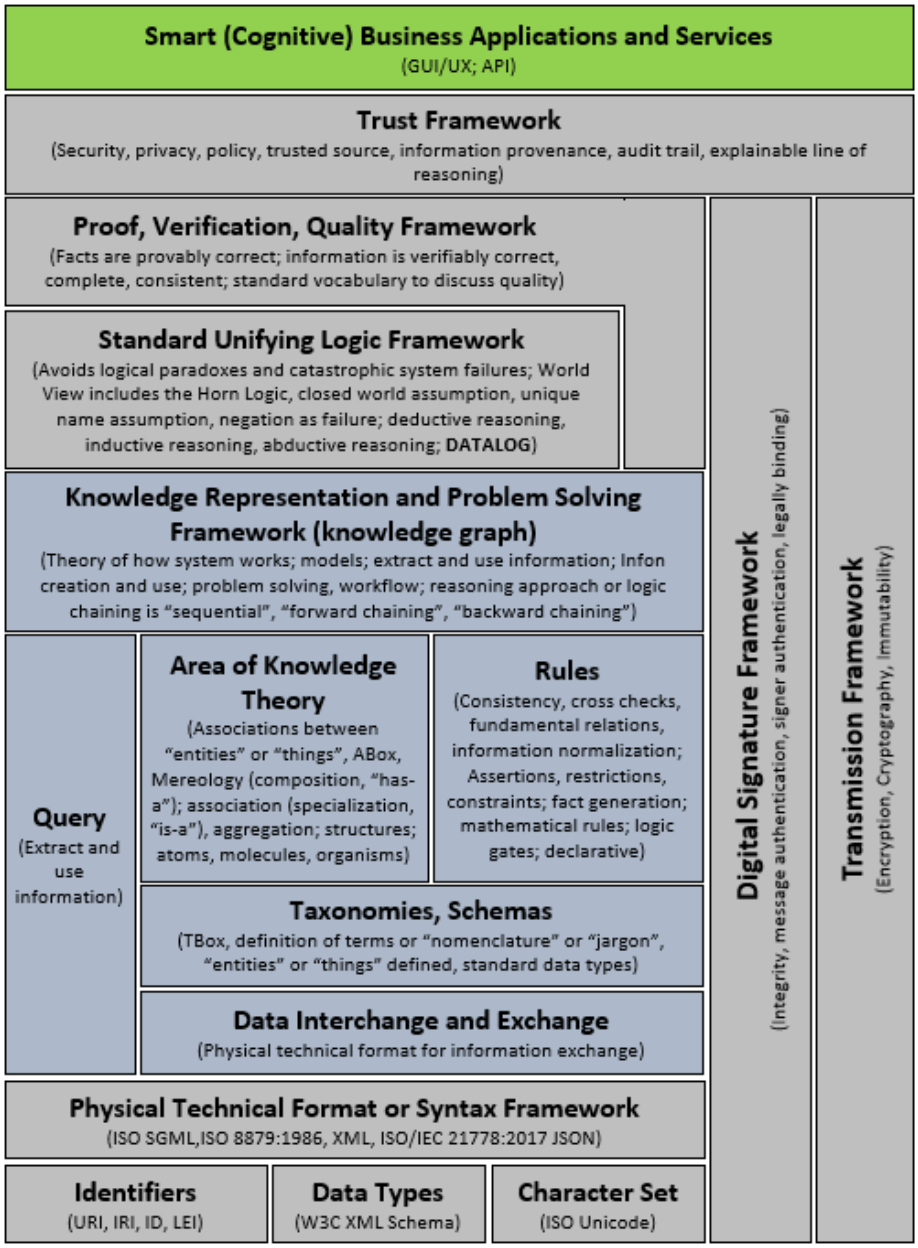
2.23. Irreducible Complexity

Irreducible complexity[88] is a term used to describe a characteristic of complex systems whereby the complex system needs all of its individual component systems in order to effectively function. In other words, it is impossible to reduce the complexity of a system (or to further simplify a system) by removing any of its component parts and still maintain its functionality objective.

The semantic web stack or "layer cake"[89] helps one understand all the pieces that are necessary to implement a compete problem solving system. This graphic generalizes those pieces, trying to separate the problem from the implementation of technology to solve the problem:

[88] *Smart (Cognitive) Business Applications and Services*, https://digitalfinancialreporting.blogspot.com/2024/02/smart-cognitive-business-applications.html

[89] *Semantic Web Stack or Layer Cake*, https://medium.com/openlink-software-blog/semantic-web-layer-cake-tweak-explained-6ba5c6ac3fab

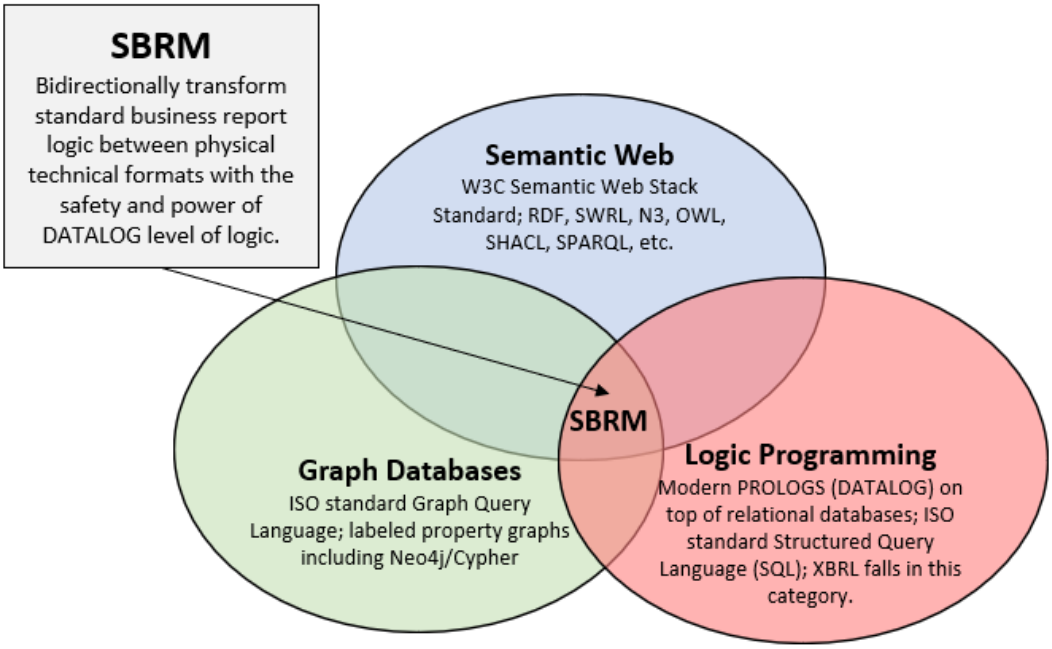


2.24. Implementing Logical Systems

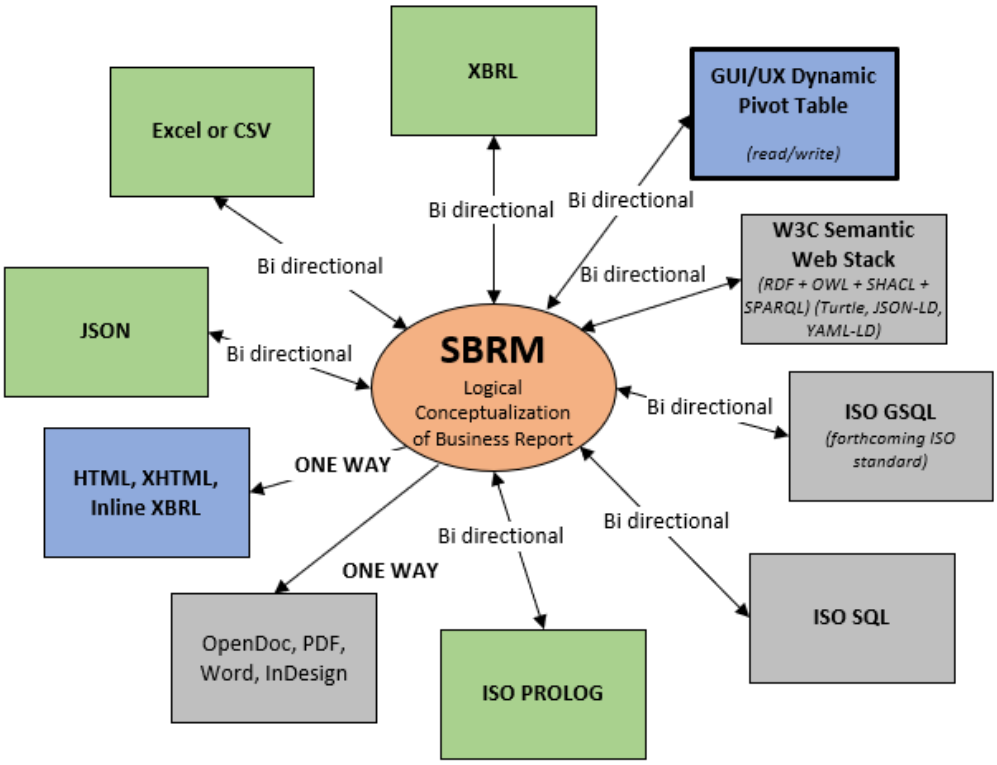
There tends to be three primary logical system (a.k.a. problem solving system) implementation approaches:

- Semantic web stack
- Graph database
- Logic programming

This graphic below shows the three approaches and makes an important point. The Standard Business Report Model (SBRM) logical conceptualization is an approach to enabling the conversion between these three technical implementation approaches.

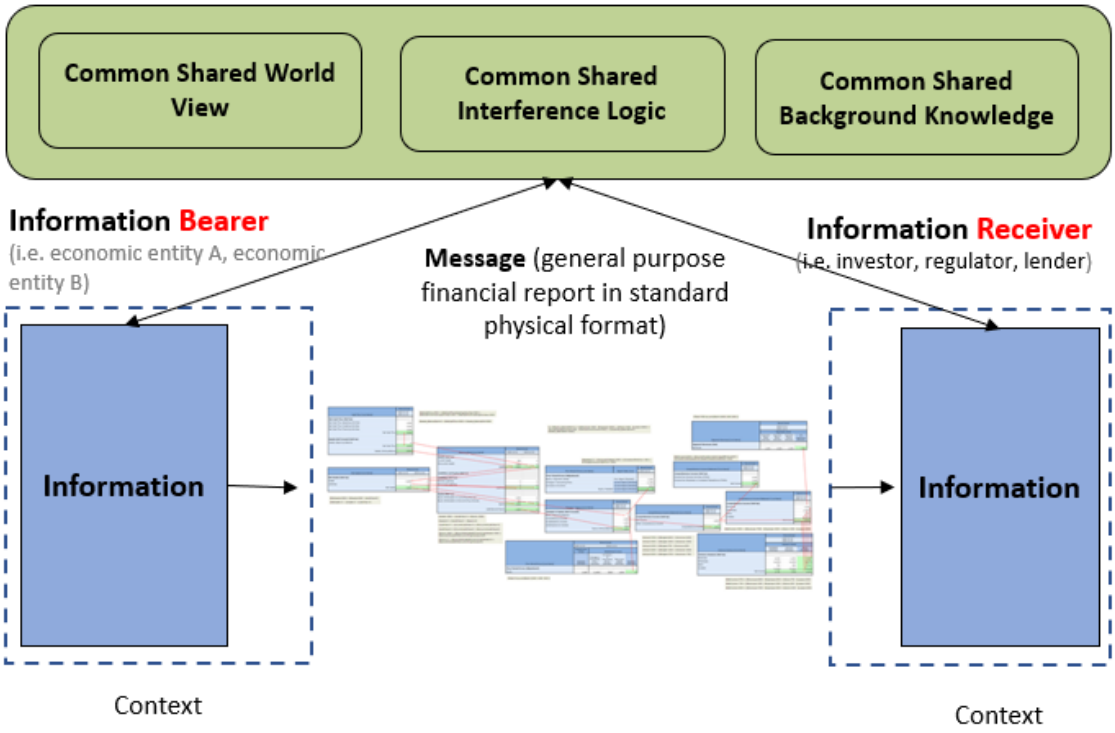


And so, the Standard Business Report Model (SBRM) enables the conversion of the logic of a business report (such as a financial report) between different technical implementations.



2.25. Effective Information Exchange Between Logical Systems

The following graphic shows what is necessary in order to exchange information effectively between two different logical systems.



3. Important Elements of Logic

This blog post was inspired by the article, *Elements of Logic*[90]. For more background details, please see this blog post *Elements of Logic for Accountants*[91]. An area of knowledge can describe the important logic of that system using these building blocks.

There tends to be several different sources of explanations of the elements of logic: philosophy, ontology and knowledge engineers, and computer scientists. These explanations tend to be inconsistent, many are incomplete, and most are either too high level to be useful or provide too much details which becomes overwhelming for business professionals and accountants.

This explanation of the elements of logic is tuned specifically for accountants and is intended to help them understand how to think about XBRL-based digital financial reports.

[90] OpenAI, Elements of Logic, https://chat.openai.com/share/b2d1dc31-4afd-49e1-8fd2-33b0d651890a

[91] Elements of Logic for Accountants, https://digitalfinancialreporting.blogspot.com/2023/10/elements-of-logic-for-accountants.html

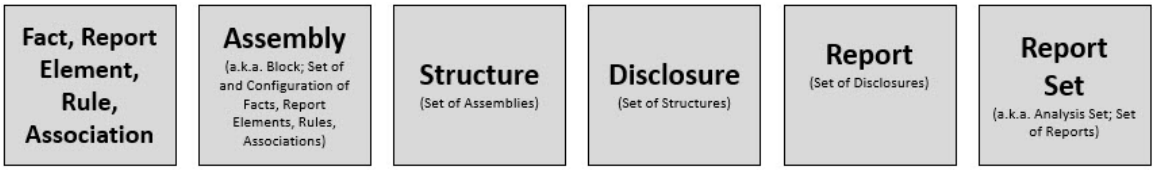
3.1. Atomic Design Methodology

Atomic Design Methodology[92] is an approach that can be used to make software applications easier to use. This is done by burying complexity within software rather than exposing that complexity to software users and forcing them to deal with that complexity.

"Atomic design is a methodology composed of five distinct stages working together to create interface design systems in a more deliberate and hierarchical manner."

- **Atoms** are the basic building blocks.
- **Molecules** are combinations of two or more atoms. These combinations of atoms take on their own unique properties, and become more tangible and operational than atoms.
- **Organisms** are assemblies of molecules functioning together as a unit. Organisms are more complex and sophisticated than molecules.

Atomic design methodology provides the high level building blocks that can be used to hide system complexity from system users. Here is an example of the high level building blocks of a financial report:



To better understand the logical objects in XBRL-based financial reports, please see *Understanding Logical Objects of XBRL-based Digital Financial Reports*[93].

3.2. Logical System Explained in More Detail

The **elements of logic** are the fundamental **building blocks** of logical theories that describe the logical conceptualization of some natural or man-made logical system.

- **Logical statement**: A logical statement is a proposition, claim, assertion, belief, idea, or fact about or related to the area of knowledge to which the logical conceptualization relates. A logical statement is a declarative sentence. Not all sentences are statements; for example, a question such as "What is your name?", or a command such as "Stop!", are not statements. There are five broad categories of logical statements:
    - **Terms**: Terms are important logical statements that define ideas or "things" used by a logical conceptualization. For example, "assets", "liabilities", "equity", and "balance sheet" are things or ideas used in a logical conceptualization.
    - **Associations**: Associations are important logical statements that describe permissible interrelationships between the terms such as "assets is part-of the balance sheet" or "operating expenses is a type-

[92] Atomic Design Methodology, https://digitalfinancialreporting.blogspot.com/2023/12/atomic-design-methodology.html

[93] *Understanding Logical Objects of XBRL-based Digital Financial Reports*, http://xbrlsite.com/2024/Library/UnderstandingLogicalObjects.pdf

of expense" or "assets = liabilities + equity" or "an asset is a 'debit' and is 'as of' a specific point in time and is always a monetary numeric value". Associations can be grouped into two broad groups:

- "**Is-a**" (a.k.a. general-special, association, type-subtype, class-subclass, equivalent-class)
- "**Has-a**" (a.k.a. part-of, has-part, part-whole, composition, aggregation)

- **Rules**: Rules (a.k.a. assertions, restrictions, constraints) are important logical statements that describe what tend to be convertible into IF…THEN…ELSE types of relationships such as "IF the economic entity is a not-for-profit THEN net assets = assets - liabilities; ELSE assets = liabilities + equity". One rule can be connected to another rule using logic gates[94] (AND, OR, NOR, NAND, XOR, XNOR, NOT) to form complex logical statements. Rules can assert mathematical relationships or derive mathematical relationships to form new facts.
- **Facts**: Facts are important logical statements that are known to be true. In the context of databases and knowledge representation, facts are often used to represent known information. Facts are logical statements about the numbers and words that are provided by an economic entity within a financial report. For example, the financial report might state "assets for the consolidated legal entity Microsoft as of June 20, 2017 was $241,086,000,000 expressed in US dollars and rounded to the nearest millions of dollars.
- **Properties:** Properties are important logical statements about the important qualities and traits of a model, structure, term, association, rule, and fact.

- **Axioms:** Axioms are foundational logical statements that are fundamentally accepted as being true per some logical system.
- **Theorems**: Theorems are logical statements that are determined to be true per logical steps that can be taken to arrive at a conclusion using axioms, other theorems, of facts.
- **Assertion**: Something that you hold to be true.
- **Restriction**: Restrictions are a special type of axiom or theorem that is imposed by some authority which chooses to restrict, constrain, limit, or otherwise impose some range on some logical artifact.
- **Classification**: Classification is the grouping of logical artifacts into sets.
- **State or state of affairs**: The current state of a system.
- **Situation**: An event, circumstance, or phenomenon that impacts state or changes to current state of affairs.
- **Infon**: A unit of information.
- **Logical structure**: A logical structure is as set of logical statements which describe the structure. An "infon" which is defined by Situation Theory is a

[94] *Logic Gates: The Building Blocks of Digital Circuits*, https://www.linkedin.com/pulse/logic-gates-building-blocks-digital-circuits-deekshith-kumar/

unit of information[95]. In infon is a type of logical structure. An infon is a useful, convenient unit or "set" of information.

- **Logical model**: A logical model is a set of specific structures that are consistent with and permissible interpretations of that model. Models add flexibility to logical conceptualizations.
- **Logical conceptualization**: A logical conceptualization is a set of models that are consistent with and permissible per that logical conceptualization. A logical conceptualization is made up of a set of models, structures, terms, associations, rules, and facts.
- **Logical theory**: A logical theory is described per some logical conceptualization forms a logical theory that explains what is permitted and what is not permitted per a logical conceptualization which is made up of a set of logical models, structures, terms, associations, rules, and facts.
- **Logical system**: A logical system can be explained by a logical theory. A logical theory is an abstract conceptualization of specific important details of some area of knowledge. The logical theory provides a way of thinking about an area of knowledge by means of deductive reasoning to derive logical consequences of the logical theory.

A logical system described by a logical theory and described by a logical conceptualization enables a community of stakeholders trying to achieve a specific goal or objective or a range of goals/objectives for an area of knowledge to agree on important logical statements used for capturing meaning or representing a shared understanding of and knowledge in some area of knowledge.

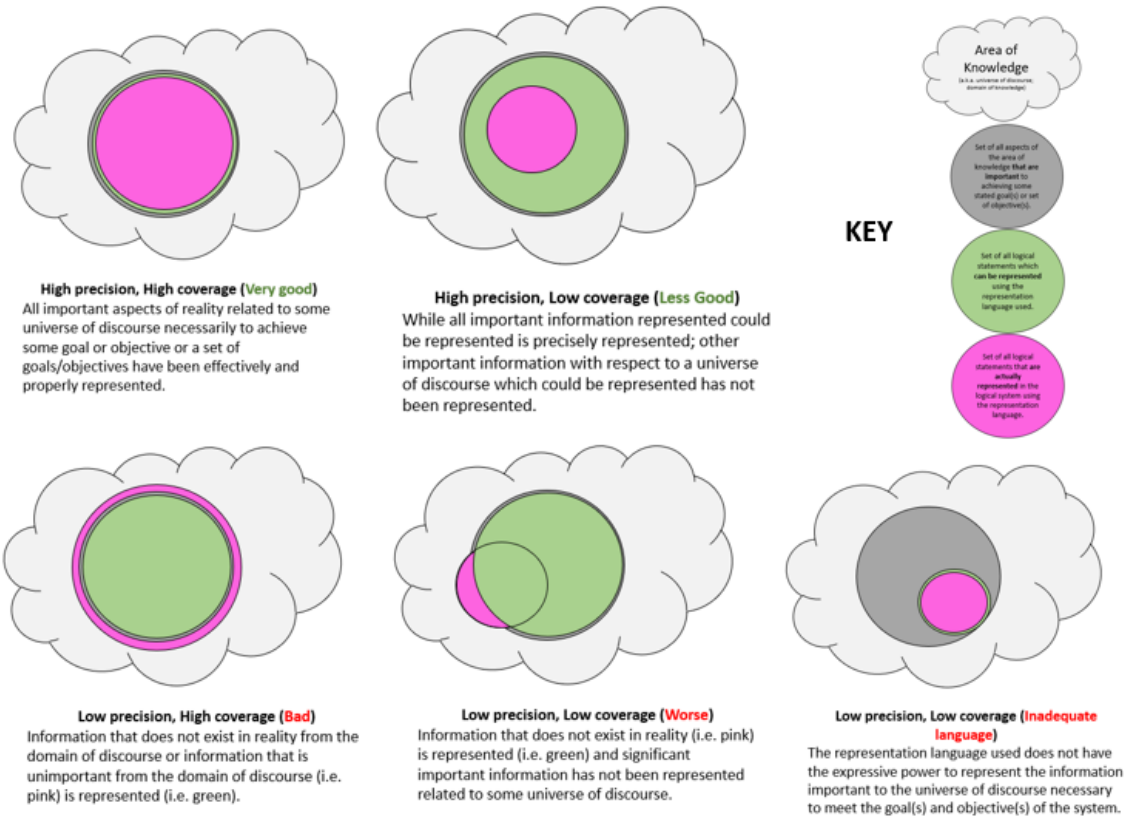
A logical conceptualization must be **consistent** (as opposed to inconsistent, making contradictory statements), **complete** (as opposed to incomplete, leaving a piece out), and **precise** (as opposed to imprecise, describing an area of knowledge incorrectly).

3.3. Quality

Quality matters in financial reporting. When representing information in the form of a logical digital twin, things can go wrong and one needs to be couscous about understand what can go wrong and mitigating those situations. The graphic below is inspired by C. Maria Keet, *An Introduction to Ontology Engineering*[96]:

[95] *Describing Situation Semantics Using Situation Theory*, https://digitalfinancialreporting.blogspot.com/2023/10/describing-situation-semantics-using.html

[96] C. Maria Keet, An Introduction to Ontology Engineering, pages 8-9, https://people.cs.uct.ac.za/~mkeet/files/OEbook.pdf#page=23



3.4. Control

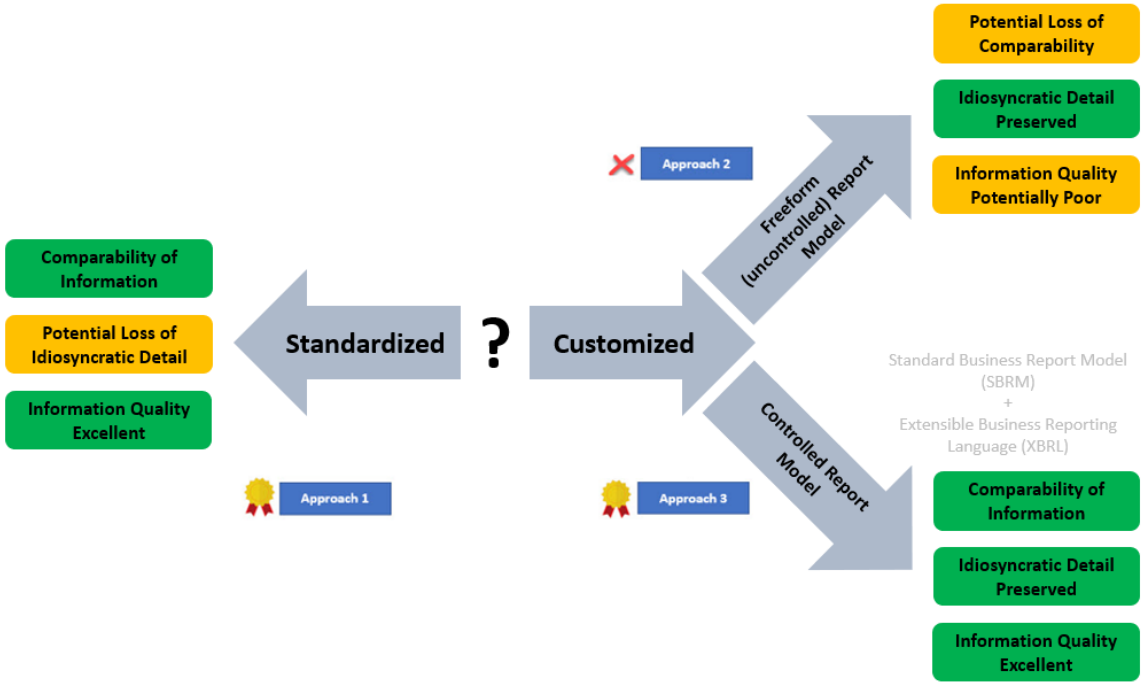
Because the creators of financial reports are permitted to modify a report model; the system needs to be able to control those modifications to make sure the modifications stay within permitted boundaries. The documents *Financial Report Knowledge Graphs* and *Essence of Accounting*[97] help you understand the dynamics of a financial report.

US GAAP and IFRS based financial reports are not forms. The *Seattle Method*[98] is an approach to managing flexibility. The *Seattle Method* is a proven, industrial strength, good practices, standards-based pragmatic approach to creating provably high quality XBRL-based general purpose digital financial reports when reporting entities are permitted to modify the report model.

The focus of the *Seattle Method* is financial reporting using financial reporting schemes such as US GAAP, IFRS, UK GAAP, and other schemes where the preparer of a financial report is permitted to modify the report model. Because modification of the report model is allowed, those modifications must be controlled to keep the modifications within permitted boundaries.

Without control, there can be no automation, no repeatable processes. Rules provide control. Control leads to high quality. High quality leads to effective automation. Accountants manage the rules.

[97] Essence of Accounting, https://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf
[98] *Seattle Method*, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf



Because financial reports are customized systems; those customizations need to be controlled. The Seattle Method and SBRM help enable that control.

Machine readable rules are used to control systems. In addition, the rules describe the system and are available for software applications to use in order to provide functionality to using software to interact with machine readable financial reports. Rules do the following:

- Elimination of "**wild behavior**" by accountants when report model can be modified
- **Description** of report (specification of what is permitted); created by standards setter or regulator or anyone else specifying a report
  - Machine ***readable form***
  - Machine readable form converted to ***human readable form***
- **Create** report based on description (assisted by software utilizing machine readable description)
- **Verify** that report has been created per description (assisted by software utilizing machine readable description)
- **Extract** information from report per report description (assisted by software utilizing machine readable description)

A financial statement tells a story. That financial report is an information product which sends a signal. Those creating reports want to make sure that the signal communicated is the signal that they intended to be communicated.

3.5. Flexibility

Flexibility in-and-of-itself is not a good or bad thing. Flexibility in the right areas is a feature. Undesirable flexibility is a bug. The trick is to provide flexibility in the precise areas that the flexibility is needed and to control that flexibility to guide logical system users where you desire them to go.

To achieve the desired level of flexibility in exactly the correct areas it is critically important to understand the moving puzzle pieces of the logical system. To do that, you need to correctly understand the goal(s) and objective(s) of all of the stakeholders of the logical system.

3.6. Classification

It was the Greek philosopher Aristotle (384-322 B.C.) that first came up with the idea of classifying plants and animals by type[99], essentially creating the notion of a hierarchy or taxonomy. The idea was to group types of plants and animals according to their similarities thus forming something that looked like a "tree" with which most people are familiar.

People tend to be less familiar with the notion of a "graph[100]". A tree, or hierarchy, is actually a type of graph. Trees/hierarchies tend to be easier to get your head around. But the real world can be more complicated than the rather simple relations that can be represented by trees/hierarchies. So, graphs[101] are used.

Classification provides three things. First, you can **describe** the model of something. Second, you can use that description of the model to **verify** an instance of the model of something against that provided description. To the extent that you have machine-readable rules, that verification process can be automated. Third, you **explain** or spell out or tell a software application (software algorithm, AI) knowledge about the *state* of where you are in your *agenda* of *tasks* necessary to meet some *goal*. To the extent that you have machine-readable rules, software can assist human users of the software in completing the *tasks* in their *agenda* and achieving that *goal*.

In his book, *Everything is Miscellaneous*[102], David Wenberger points out that every classification system has problems. Every classification scheme ever devised inherently reflects the biases of those that constructed the classification system. The role metadata plays in allowing you to create your own custom classification system so you can have the view of something that you want. The best thing to do about this is to create a flexible enough classification system to let people classify things how they might want to classify them, usually in ways unanticipated by the creators of the classification system.

Wenberger also points out the power of metadata and the three orders of order. First order of order, second order of order, and third order of order.

---

[99] Lois Tilton, *From Aristotle to Linnaeus: the History of Taxonomy*, https://davesgarden.com/guides/articles/view/2051

[100] Wikipedia, *Graph Theory*, https://en.wikipedia.org/wiki/Graph_theory

[101] Maël Fabien, *Introduction to Graphs (Part 1)*, https://towardsdatascience.com/introduction-to-graphs-part-1-2de6cda8c5a5

[102] See, http://xbrl.squarespace.com/journal/2011/1/31/us-gaap-taxonomy-build-it-to-allow-reoganization.html

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

In fact, the third-order practices that make a company's existing assets more profitable, increase customer loyalty, and seriously reduce costs are the Trojan horse of the information age. As we all get used to them, third-order practices undermine some of our most deeply ingrained ways of thinking about the world and our knowledge of it.

3.7. Knowledge Commitment

Ontological commitment is a concept used in philosophy, artificial intelligence, and information systems. It refers to the agreement to use a shared vocabulary, associations, and rules in a coherent and consistent manner within a specific context.

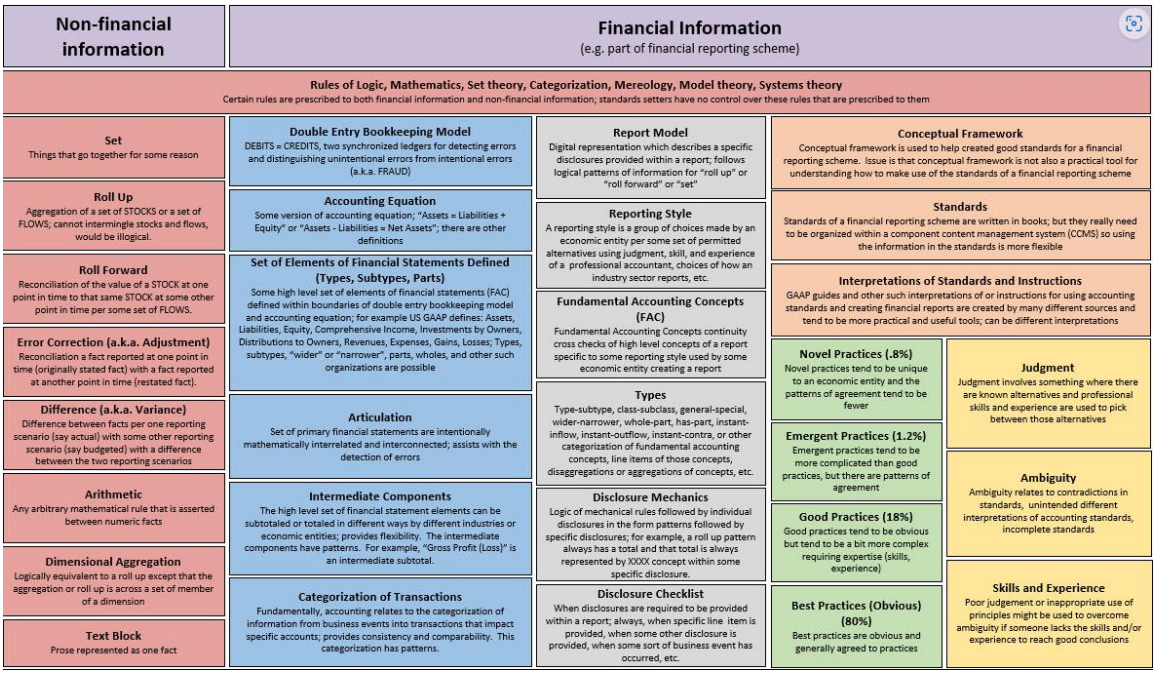
In simple terms, when you make an ontological commitment, you're essentially saying "I agree that these things exist in the way we have defined them in our shared understanding and I will use these definitions, associations and rules consistently when we talk about those things."

Knowledge commitment[103] is the same idea as ontological commitment.

3.8. Fundamental Puzzle Pieces of Digital Financial Reporting

The details of the fundamental puzzle pieces of digital financial reporting[104] are explained by the *Essence of Accounting*[105], *Financial Report Knowledge Graph*[106], and *Logical Theory Describing Financial Report*[107]. That information is summarized in this graphic below:

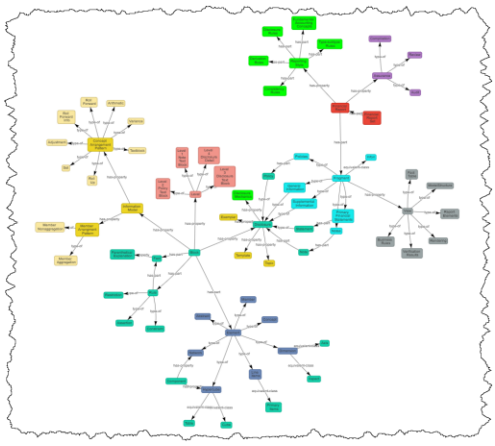
[103] Knowledge Commitment, https://digitalfinancialreporting.blogspot.com/2023/12/knowledge-graph-commitment.html
[104] Puzzle Pieces of Digital Financial Reporting, https://digitalfinancialreporting.blogspot.com/2023/11/puzzle-pieces-of-digital-financial.html
[105] Charles Hoffman, CPA, *Essence of Accounting*, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf
[106] Charles Hoffman, CPA, *Financial Report Knowledge Graph*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf
[107] Charles Hoffman, CPA, *Logical Theory Describing Financial Report*, http://xbrlsite.com/seattlemethod/LogicalTheoryDescribingFinancialReport_Terse.pdf



3.9. Logical Theory Describing Financial Report

To make working within an area of knowledge such as financial reporting approachable to business professionals it is essential to create a high level logical model of that system and software that leverages that high level logical model to turn a general purpose system that is very flexible but too hard for business professionals to make us of into a specialized system that is approachable by business professionals. This approach can be used to hide the information technology complexity from system users by burying that complexity within the software platform and software application; exposing software system users to the logic of their area of knowledge with they understand.

The *Logical Theory Describing Financial Report*[108] provides that high level model for financial reporting.



[108] Charles Hoffman, CPA, *Logical Theory Describing Financial Report*, http://xbrlsite.com/seattlemethod/LogicalTheoryDescribingFinancialReport_Terse.pdf

4. Representing Knowledge

This section provides important information about representing knowledge.

4.1. Sensemaking

Sensemaking[109] is the process of determining the deeper meaning or significance or essence of the collective experience for those within an area of knowledge. Systems evolve over time[110].

Some professional accountants need to develop the skills to represent things digitally[111]. A Wardly Map is a sensemaking tool.

A Wardley Map is a sketch. Usually of a business, market, or any other kind of work system. A Wardly Map is a design, like a blueprint. The Wardly Map might be right, it might be wrong; but it is something that can discussed and refined together on paper to make sure everything we do in reality is as purposeful as it can be. These designs will inevitably be put to the test. No one is exempt; not even governments or non-profits. So, to help us cope, we acknowledge this fact up front by arranging our blueprint by evolutionally stage; from the uncharted where things are uncertain, high failure, and a gamble; to the industrialized where things are known, reliable, and standard practice. Awareness of these qualities helps us approach each part of the system deliberately. No "one size fits all", only careful specific intention. But the map is only the beginning. Behind Wardley Mapping is a deeper strategic thinking process. There are many patterns to learn, principles to practice, and moves to make. Take it one step at a time for even the smallest insight can change everything

4.2. Formalism

A formalism is the practice of using strict and complete methods to define and specify the important essence of a model. Every term in the formalism is given precise a definition. Every variable, parameter, rule, and factor are given a precise name and definition. A formalism stives to be complete and precise definition of the model and its mechanisms which are being described, ideally with no redundancy or gaps.

Logic and knowledge graphs (a.k.a. semantic networks[112]) are formalisms for describing information.

4.3. Information is Data in Context

There are specific differences between data, information, knowledge, and wisdom[113]: (note that I have added insight to match with the graphic which is shown next)

- **Data**: The basic compound for intelligence is data. Data are measures, observations, symbols, phenomenon, utterances, and other such

[109] Sensemaking, http://xbrl.squarespace.com/journal/2021/11/18/sensemaking.html

[110] *Evolution of a System*, http://xbrlsite.com/2023/Library/EvolutionOfSystem.pdf

[111] Skills to Represent Things Digitally, https://digitalfinancialreporting.blogspot.com/2023/02/skill-to-represent-things-digitally.html

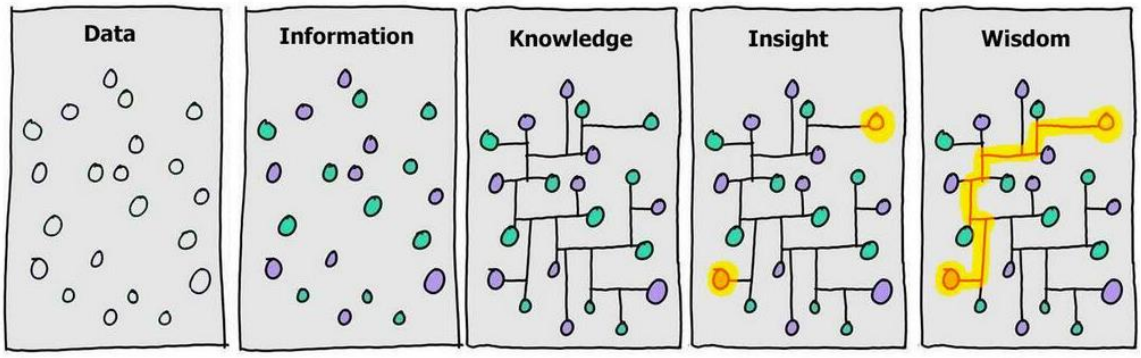
[112] Wikipedia, *Semantic Network*, https://en.wikipedia.org/wiki/Semantic_network

[113] Wikipedia, *DIKW Pyramid*, https://en.wikipedia.org/wiki/DIKW_Pyramid

representations of the world around us presented as external signals and picked up by various sensory instruments and organs. *Simplified: data is raw facts and numbers*.

- **Information**: Information is produced by assigning relevant meaning related to the context of the data to the data. *Simplified: information is data in context*.
- **Knowledge**: Knowledge is the understanding or interpretation, a justifiable true belief, of information and approach to act upon the information in the mind of the perceiver. *Simplified: knowledge is the interpretation of information*.
- **Insight**: Insight is the first step in putting information and knowledge to work for you.
- **Wisdom**: 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*.

The following graphic perhaps provides the best visual explanation as to the difference between data, information, knowledge, insight, and wisdom[114] that I have run across:



Decision = Data + Knowledge

Ontology = Terms + Associations

Knowledge = Ontology + Rules

Algorithm = Logic + Control

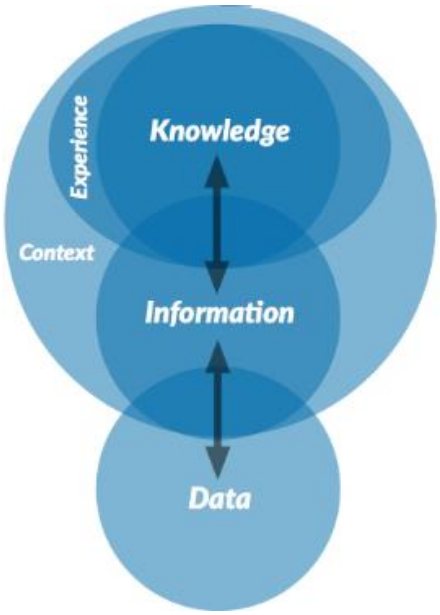
The difference between *data* and *information* is that data is the raw numbers and words where information is data in context. This is important to understand as most problems faced by accountants are an information problem, rather than a data problem. Getting data is easy. Knowing what that data represents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

[114] Tumblr, *Information isn't Power*, https://random-blather.com/2014/04/28/information-isnt-power/

Insight and wisdom are related to putting information and knowledge to work for you.

The following graph created by Shawn Riley shows the important to understand differences between data, information, and knowledge[115].



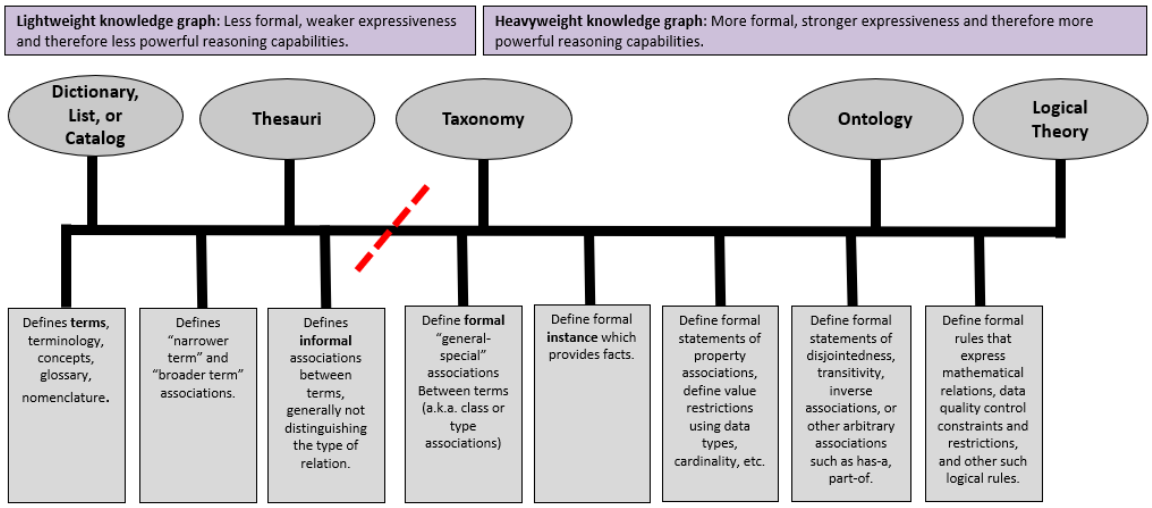
The important point to understand here is that it takes the skill and experience of human professionals to create information and knowledge and to put that knowledge into the proper context.

Our focus here is information, not data.

4.4. Tools for Representing Knowledge

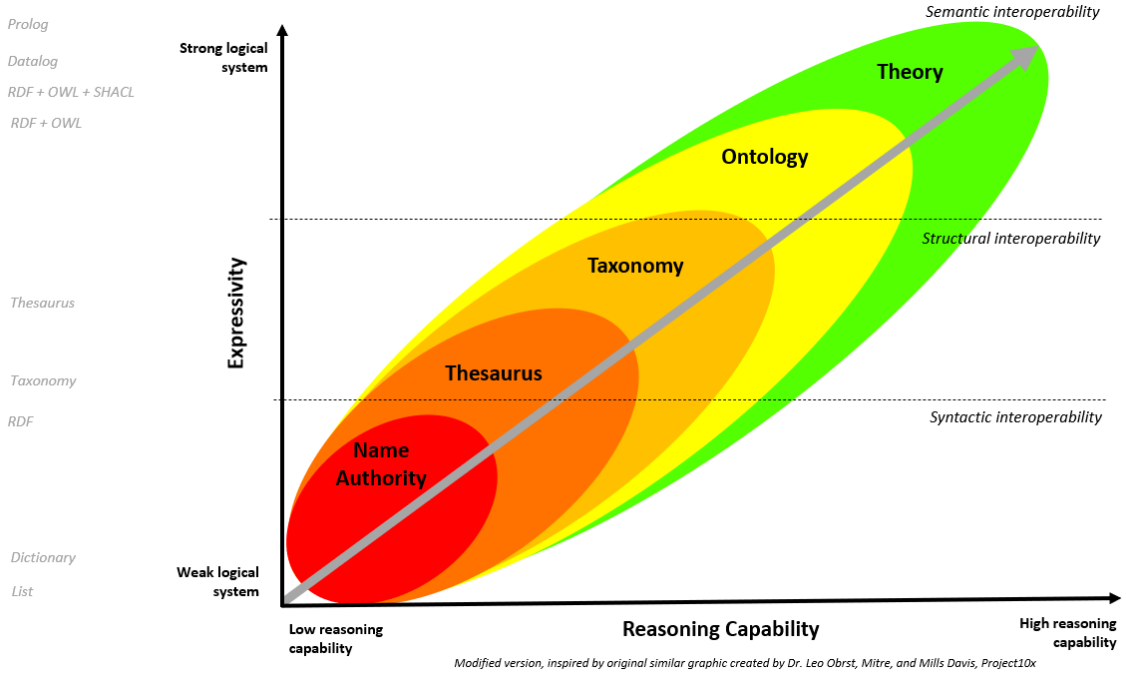
There are a number of different tools that can be used to effectively represent knowledge in the form of a knowledge graph. Below you see a spectrum of such tools with the least powerful tools on the left and increasing in power to the right:

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



Inspired primarily by Deborah L. McGuinness, *Ontologies for the Modern Age*, Slide 4, https://www.slideshare.net/deborahmcguinness/ontologies-for-the-modern-age-mcguinness-keynote-at-iswc-2017

Another version of the information above is this graphic:



Modified version, inspired by original similar graphic created by Dr. Leo Obrst, Mitre, and Mills Davis, Project10x
Mills Davis, *AI - Externalization of Mind*, slide 60, https://www.slideshare.net/Mills/ai-externalization-of-mind-81319464

When representing knowledge, the right tool should be used for the job. A logical theory is the easiest way to enable business professionals to understand a logical system because business professionals have an innate understanding of logic.

4.5. Knowledge Graph

We communicate using knowledge graphs. When you go to a whiteboard and draw circles and squares and connect them with lines with arrows you are drawing a graph and communicating knowledge. Those circles, squares, lines, and arrows are intuitively understandable and very expressive. These informal knowledge graphs

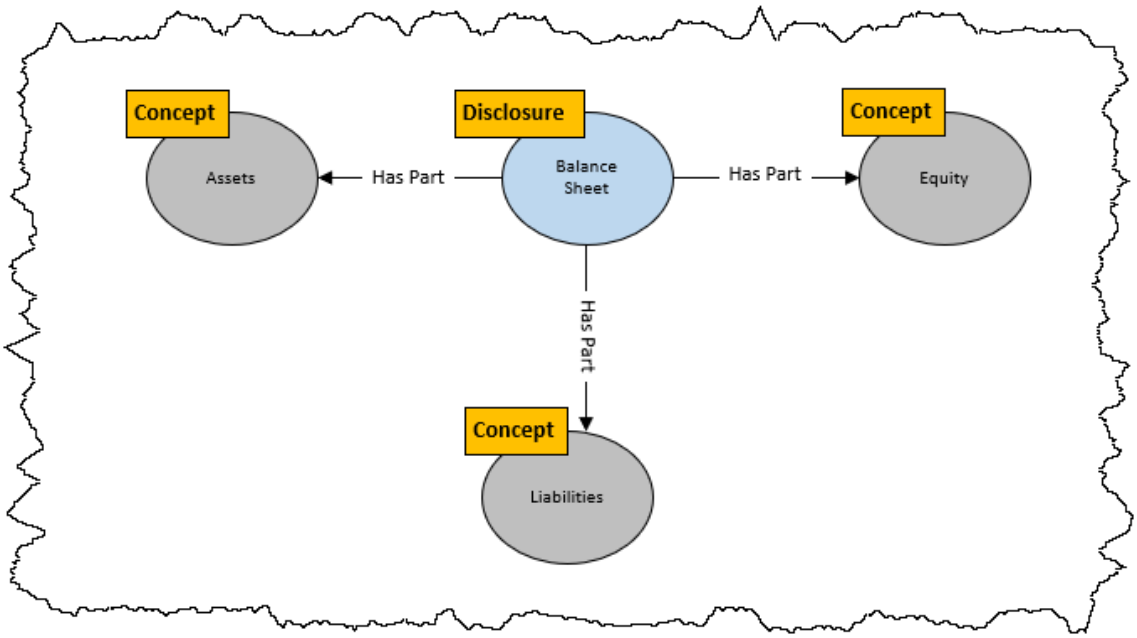
like this have been used by humans to communicate information for quite some time. The earliest documented use of knowledge graphs (a.k.a. semantic networks) was the third century CE.

Knowledge is the understanding or interpretation of information. Knowledge relates to terms, structures, associations, rules, facts, and skills acquired by a person through experience or education that relates to the theoretical or practical understanding of something.

A **graph**, in formal terms, is a set of vertices and edges. In less intimidating language, a graph is a set of nodes and the relationships that connect the nodes together. Graphs represent things as nodes and the ways in which those things relate to one another and rest of the world as relationships.

A graph is a general-purpose communications tool that allows us to model all sorts of scenarios in terms that are innately understandable to humans. One thing that can be represented in the form of a graph is knowledge.

This is a simple graph of knowledge, or a **knowledge graph**:



A knowledge graph[116], also known as a semantic network, represents a network of real-world things (entities)—i.e. objects, events, situations, or concepts—and illustrates the relationship between them. This information can be visualized as a graph structure.

4.5.1.Graphs

When I use the term graph, I am referring to the term in the context of graph theory[117] which is a discipline of mathematics. Wikipedia's definition of graph theory and graph is:

[116] IBM, *What is a Knowledge Graph?*, https://www.ibm.com/topics/knowledge-graph
[117] Wikipedia, *Graph Theory*, https://en.wikipedia.org/wiki/Graph_theory

> In mathematics, graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects.

This is a very simple graph:

Just like most other things, graphs have a jargon. In formal graph jargon, the circles are referred to as an edge and the line is referred to as a vertex.

Others in other areas use different terminology to refer to exactly the same idea. Here are synonyms for the notions of *edge* and *vertex*:

Edge	Entity	Node	Point	Report element
Vertex	Relationship	Line	Path	Association

A graph can have can have one or more paths between points; paths can have loops or cycles, circuits, as well as can have self-loops, and paths can go in one direction or both directions.

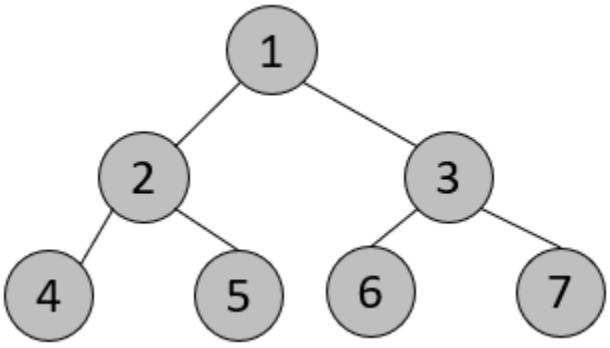
To better understand graphs, let's look at some subtle but very important differences between some different types of graphs.

If we take the time to consciously formalize the rules related to graphs and understand those rules these communications tools become more effective and they can even be understood by computer software applications.

4.5.2. Trees

A tree is a special type of graph. Most people are more familiar with trees than graphs. A tree is what is called an undirected graph because the items in a tree are connected by exactly one path. This is important to understand because it means that trees are safer than other types of graphs which can contain cycles. But trees have a limitation in that an edge can appear only once in a tree and a tree always has exactly one root edge. Also, because trees are undirected, they provide less information and so they are less powerful in terms of expressiveness.

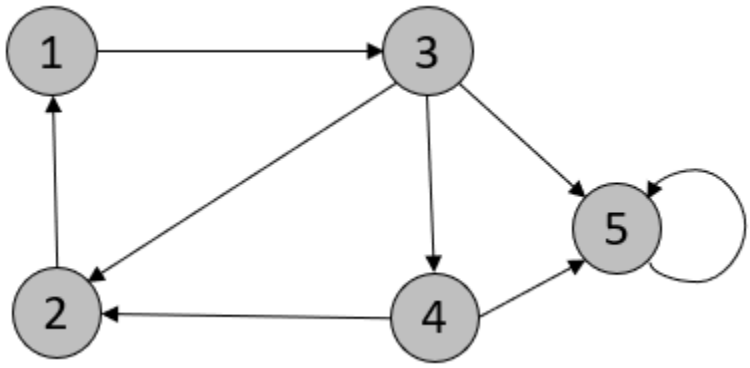
The following is an example of a graph that is also a tree:



Notice the root, node number 1 and that ever other node that appears is unique. Notice also that there is no direction associated with the lines that appear between the nodes.

4.5.3.Directed Graph

A directed graph is a special type of graph that provides a direction on each vertex. For example, below you see a directed graph: (notice the arrows)

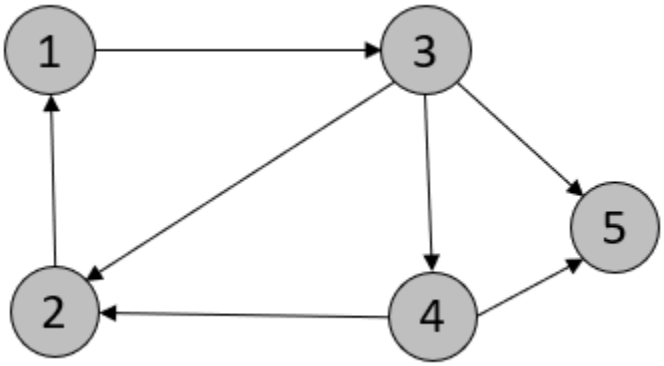


Note that each vertex (line) has an arrow that points in a specific direction; that is what makes the graph a directed graph. Note edge (node) number 5 which has an arrow that points to itself; that is a cycle. Cycles like that can cause issues such as causing an infinite loop. Those sorts of issues can be solved by using a directed acyclic graph that does not allow such cycles which we will cover next.

Directed graphs are more powerful than trees but because of the possibility of a cycles, they can be unsafe for certain things.

4.5.4.Directed Acyclic Graph

A **directed acyclic graph** (DAG) is an even more special type of graph that provides a direction on each vertex and you are guaranteed not to have any cycles in the graph. This makes the graph very save as there is not a possibility of creating infinite loops that can break software applications. For example, below you see a directed acyclic graph:



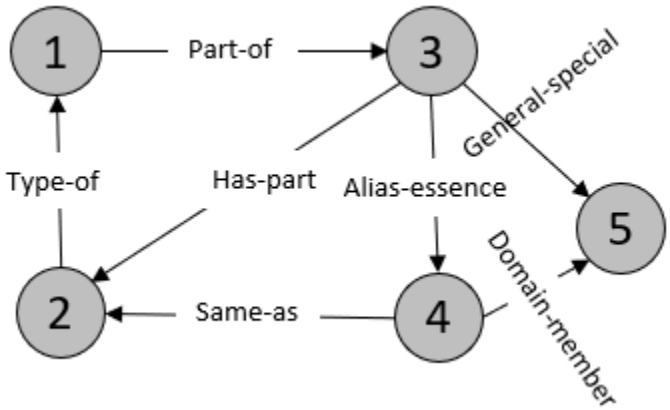
Note again that each vertex has an arrow which specifies a direction and that there are no cycles making this a directed acyclic graph. Note that there is not one specific edge (node) that can be considered the root of the graph.

But note that you don't have any information about the nature of the vertices (lines). What if information was provided about the relationships in the graph as communicated by the vertices (lines)?

4.5.5.Labelled Directed Acyclic Property Graph

A labeled directed acyclic property graph specifies a type of vertex for each association between any two edges. Specifying that feature, the nature of the relationship, provides additional information that is useful in working with a graph.

For example, below you see a labeled directed acyclic graph: (notice the labels on the arrows)



Note the labels that explain each vertex in the graph. You can, for example, query a graph for those relationship types. Labeled directed acyclic graphs have the most power in terms of expressiveness but are also very safe to use because they are guaranteed not to contain any cycles which can lead to catastrophic failure when read by a machine-based process.

4.5.6.Typed Directed Acyclic Property Graph

Now I am getting over my head, but this seems to have a profoundly important impact on functionality and query speed. There seems to be a difference between a "labeled property graph" and a "typed property graph". Also, there seems to be a critically important difference between RDF graphs and graph databases. Seems that RDF graphs are typed, but you cannot add properties. Seems that labeled property graphs are more flexibly, but that flexibility might not be needed and it impacts functionality and query speed.

This is maddeningly difficult for a business professional to understand. But, reading this article *Labeled vs Typed Property Graphs — All Graph Databases are not the same*[118] and understanding TypeDB[119] are important. Strongly typed graph

[118] Medium, *Labeled vs Typed Property Graphs — All Graph Databases are not the same*, https://medium.com/geekculture/labeled-vs-typed-property-graphs-all-graph-databases-are-not-the-same-efdbc782f099

[119] Vaticle, Strongly Typed Database, https://vaticle.com/

databases seem very compelling. "TypeDB provides a strong type system for developers to break down complex problems into meaningful and logical systems. Through TypeQL, TypeDB provides powerful abstractions over low-level and complex data patterns."

This seems like incredibly important stuff but I don't really understand it as well as I would like to. And frankly, most software engineers don't seem to understand it well either which makes this problematic. Finally, how does something like PROLOG fit into this comparison.

4.5.7.Knowledge Graph with a Logical Schema

Suppose that you have a strongly typed acyclic labelled property graph of semantic knowledge; but the knowledge in that knowledge graph is incomplete, it does not accurately or precisely represent the logic of an area of knowledge, and the logic within that knowledge graph was inconsistent, contradicting itself. How useful is that?

And that is why it is important to provide a logical schema to verify that the information contained within a knowledge graph is consistent with expectation, is complete, and is precise. This is why you want a professional quality knowledge graph.

4.6. Professional Knowledge Graphs

Strongly typed directed acyclic labelled property graph of semantic knowledge which includes a logical schema are the knowledge representation tool of professional master craftsmen[120].

This describes each of the carefully used words that I have selected and used above. Notice that there is nothing about which syntax is used to physically represent that type of knowledge graph.

- "**Graph**" in knowledge graphs as contrast to the notion of a "tree" which is a type of graph which is not as flexible. So a knowledge GRAPH as contrast to a knowledge TREE is more flexible should you need that flexibility and many business use cases need that flexibility. (This explains the difference between a graph and a tree.)
- "**Directed**" as contrast to "undirected"; directed graphs can specify a direction of a relationship whereas undirected graphs do not. (This explains the difference between directed and undirected graphs.)
- "**Acyclic**" graphs do not contain cycles as contrast to a "cyclic" graph that does contain cycles. Cycles can be problematic and cause catastrophic processing failures such as an infinite loop. Graphs with cycles can cause logical paradoxes as I understand it and therefore cycles in graphs are to be avoided. (This explains the difference between acyclic and cyclic graphs.)
- "**Property**" graphs allow you to formally express properties on a relationship (a.k.a. edges) as contrast to only being able to provide properties for "entities" or "things" (a.k.a. nodes). Since the properties expressed provide

[120] Professional Knowledge Graphs, https://digitalfinancialreporting.blogspot.com/2023/12/professional-knowledge-graphs.html

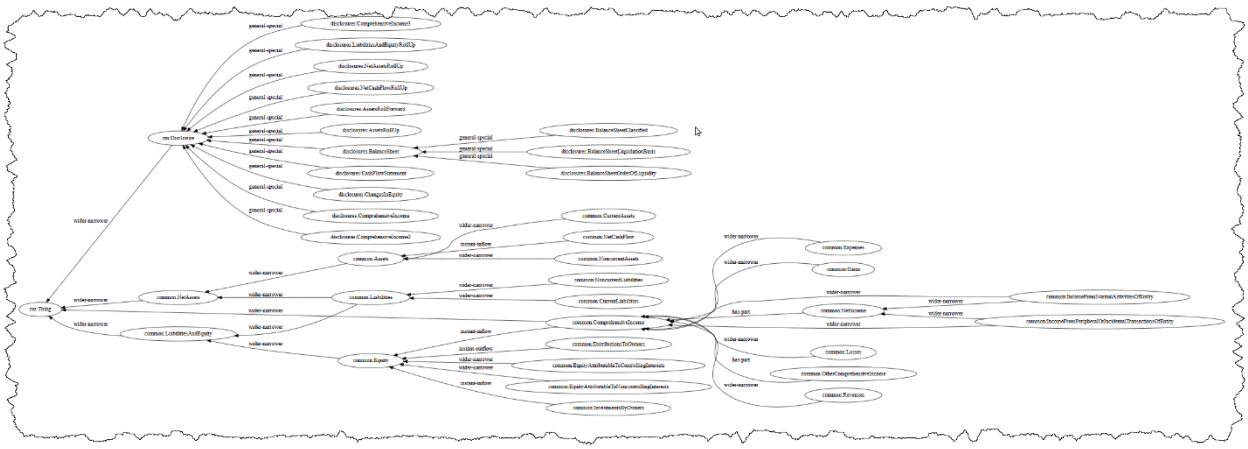
more information, property graphs can be more powerful in terms of processing capabilities. (This explains property graphs.)

- "**Typed**" graph systems provide a formal system of typing properties and subtyping. This is as contrast to untyped graphs which do not. Typed graph theory explains the details. Another term for this is the notion seems to be strong typing. And yet another term seems to be static typing versus dynamic typing. The graph database TypeDB explains why typing is important.
- "**Labeled**" graphs have explicit, distinct identifiers while unlabled graphs do not. See this explanation of labeled graphs and unlabled graphs. It is unclear to me whether a graph can be both labeled and typed. This seems to imply that you get one or the other.
- "**Semantic**" graphs are can provide models of meaning as contrast to other knowledge graphs which are for something different such as the presentation of information within sheets, rows, columns, and cells of a spreadsheet. When I am talking about knowledge graphs it relates to business semantics or business logic, not presentation. For more information see, Semantic Graphs - an Introduction; All About Knowledge;
- "**Logical schema**" is similar to the schema of a database but far, far richer in terms of logical expression.

In addition to being a strongly typed directed acyclic labeled property graph of semantic knowledge and having a logical schema; one needs a rules engine to enforce the logic at the DATALOG level of logic. TerminusDB[121] seems to fit this bill.

4.7. Basic Example of Knowledge Graph

The following is an example knowledge graph that explains accounting information related to the four statement financial statement model[122]:

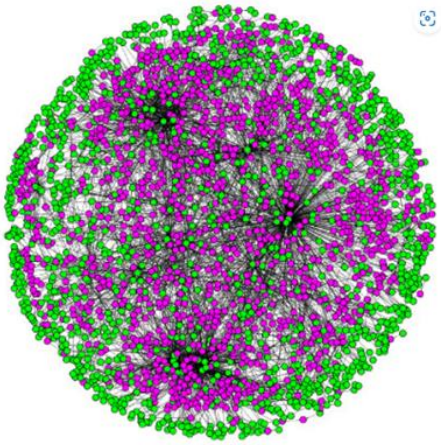


[121] TerminusDB, Datalog Explanation, https://terminusdb.com/docs/datalog-explanation/

[122] Auditchain, *Four Statement Model Knowledge Graph*, https://auditchain.infura-ipfs.io/ipfs/QmTtcz3rcEmsSYjkev3Xo9qHH2YUrnkGimyt9Wq2YV3Lr2/typeSubTypeGraph.html

4.8. Graph Hairball (Generalized Knowledge Graph)

When viewed in general terms; a knowledge graph system logic, the "things" and "relations between things" that graph theory calls "vertices" (a.k.a. nodes, points, entities, things) and "edges" (a.k.a. links, lines, relations, associations), looks like a big graph hairball as some people call it an example of which you can see here in the graphic below. This is because in order to work with any specific knowledge graph system of logic we need to look at that is common between the knowledge graph system which is the things and the types of relations between things.



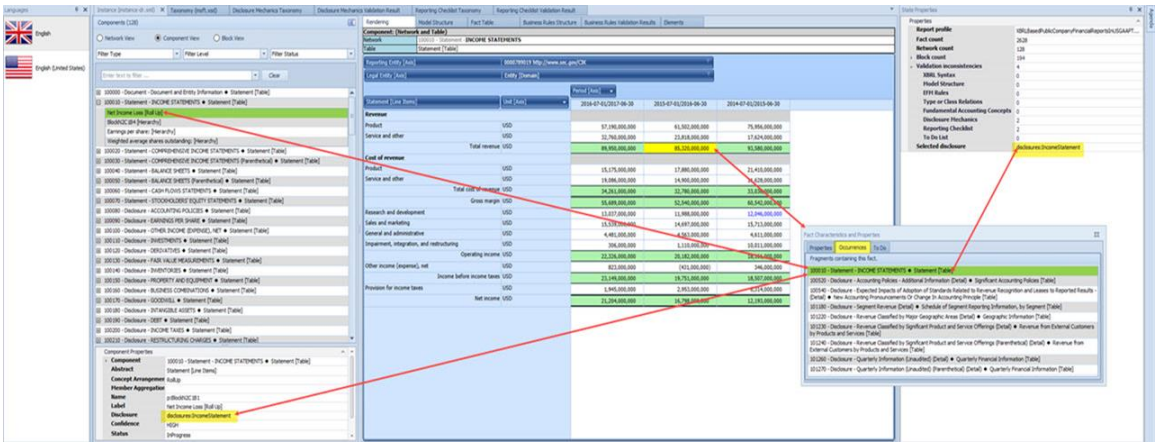
This view of the information in a knowledge graph can be useful for some things, but it is less useful for other things. In order to get a more useful view of the logic contained in a knowledge graph; we specialize the view to the specific application of the graph of knowledge that we are using.

By leveraging the specific high level logical mode of a specific type of knowledge graph system we can get a view of the knowledge graph system that is more useful to that specific area of knowledge.

4.9. Financial Report Knowledge Graph (Specialized knowledge graph with a high level logical model)

As explained in the document, *Financial Report Knowledge Graph*[123]; here is a "specialized" or specific view of one type of knowledge graph, a financial report, that is specific to the area of knowledge of financial reporting:

[123] Charles Hoffman, CPA, *Financial Report Knowledge Graph*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf



Here is an example of an XBRL-based digital financial report that has been converted into an HTML human readable rendering of that machine readable knowledge graph[124]:

Component Perspective | Overview Perspective

All Components (7)

01-Balance Sheet | Balance Sheet [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | **Combined**

02-Net Assets | Net Assets [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | **Combined**

03-Comprehensive Income | Comprehensive Income Statement [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | **Combined**

04-Comprehensive Income 2 | Comprehensive Income Statement [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | **Combined**

05-Comprehensive Income 3 | Comprehensive Income Statement [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | **Combined**

06-Cash Flow | Cash Flow [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | **Combined**

07-Changes in Equity | Changes in Equity [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | **Combined**

Rendering

Component: (Network and Table)	
Network	**01-Balance Sheet** (http://www.xbrlsite.com/report/role/BalanceSheet)
Table	Balance Sheet [Hypercube]

Slicers (applies to each fact value in each table cell)

Reporting Entity [Axis]	GH259400TOMPUOLS65II (http://standards.iso.org/iso/17442)

Balance Sheet [Line Items]	Period [Axis] 2020-12-31	2019-12-31
Assets [Roll Up]		
Current Assets	500	0
Noncurrent Assets	3,000	0
Assets	3,500	0
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities	0	0
Noncurrent Liabilities	0	0
Liabilities	0	0
Equity [Roll Up]		
Equity Attributable To Controlling Interests	3,000	0
Equity Attributable to Noncontrolling Interests	500	0
Equity	3,500	0
Liabilities and Equity	3,500	0

4.10. Knowledge Assembly

I ran across a new term, knowledge assembly[125], which is used to explain some important ideas related to the idea of a knowledge graph or knowledge graph system. Similar terms are knowledge fabric[126] or data fabric or data mesh or information mesh. Allow me to explain.

---

[124] *HTML rendering of financial report knowledge graph*, http://www.xbrlsite.com/seattlemethod/golden/common2/reference-implementation/evidence-package/

[125] Knowledge Assembly, https://digitalfinancialreporting.blogspot.com/2023/08/knowledge-assembly.html

[126] Knowledge Fabric, https://youtu.be/yNsE02FAR3w?si=_VbcDy40u_nN9HUb

Hetionet[127] provides an example of what they refer to as a knowledge assembly. Hetionet is a demonstration of connecting multiple databases of information (an "assembly") that contain data and meta data related to biomedicine.

A similar sort of knowledge assembly for a financial reporting scheme which is a subset of finance. There is knowledge that is common to all financial reporting schemes such as the double entry bookkeeping model and the accounting equation. The logical model of a financial report is part of that assembly. The conceptual framework of the financial reporting scheme is a part of the knowledge assembly. The different reporting styles permitted by that financial reporting scheme is part of the knowledge assembly. How to compute financial ratios used to analyze information reported per that scheme are part of the knowledge assembly. A specific economic entity's report model is another part of the assembly. The entire SEC EDGAR system database of reports could be part of the knowledge assembly, or similarly the ESMA database of reported financial information.is another part of the assembly.

4.11. Knowledge Graph Systems

This article published by the IEEE, *EHR-Oriented Knowledge Graph System: Toward Efficient Utilization of Non-Used Information Buried in Routine Clinical Practice*[128], uses the term "knowledge graph system" and others have used the term "knowledge graph approach" to describe what I am trying to achieve using the *Seattle Method*.

The vision provided by that article for electronic health records is similar to what I am trying to achieve for financial reporting.

4.12. Knowledge Graph Systems for Financial Reporting

Think of applying the notion of a knowledge graph system to financial reporting. The graphic that I show below is my view of how a knowledge graph system for financial reporting might work and more importantly why I think it will work that way.

First, you have to understand that a financial report is, and always has been, a knowledge graph[129]. In the past, that financial report knowledge graph was only readable by humans. Going forward, that financial report knowledge graph will be still be readable by humans; but it will also be readable by a machine in the form of a software application.

This transformation is less about artificial intelligence (AI); it is more about a transformation caused by HUMANS effectively harnessing the power of AI[130]. Transformation = AI + HI. The transformation will be a lot of work, most people will get it wrong, but a few will get it right and then everyone will copy those who got it right and then improve things even more. Where things end up will be a

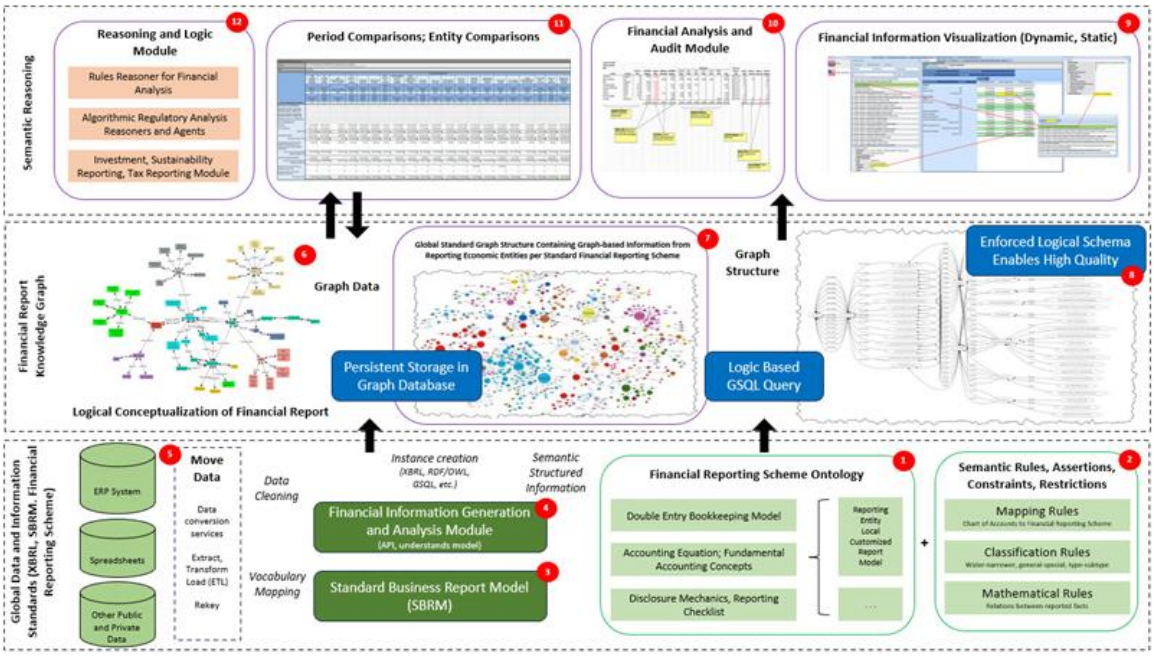
[127] Hetionet, https://het.io/

[128] IEEE, *EHR-Oriented Knowledge Graph System: Toward Efficient Utilization of Non-Used Information Buried in Routine Clinical Practice*, https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9444689

[129] *Financial Report Knowledge Graph*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf

[130] *Transformation = AI + HI*, https://digitalfinancialreporting.blogspot.com/2023/04/ai-hi.html

system similar to the graphic below. Here is a graphic that describes my view of the pieces that make up a knowledge graph system for financial reporting[131]:



For more details and examples of the pieces of that system, refer to this blog post[132]. So how do you physically represent that knowledge graph system? Well, you use a knowledge assembly.

4.13. Global Standard Knowledge Assembly

Imagine a global standard knowledge assembly for financial reporting.

I provide examples of what amount to a knowledge assembly in my examples of different general purpose financial report creation schemes[133]. All of my examples use the same framework which I refer to as the Seattle Method. That framework is heavily tested and is proven to work effectively[134].

My framework uses the global standard XBRL to represent the knowledge graphs within a standards-based knowledge graph system using a standards-based framework (Seattle Method, Standard Business Report Model) that make up the knowledge assembly. Other machine readable representation approaches of such a knowledge assembly exist such as the Semantic Web Stack (RDF+OWL+SHACL and then use SPARQL to query information) or a graph database (using GSQL[135]) or

[131] *Knowledge Graph System for Financial Reporting* (Graphic), http://xbrlsite.com/seattlemethod/platinum/KnowledgeGraphSystem_FinancialReporting.jpg
[132] *Knowledge Graph System for Financial Reporting*, https://digitalfinancialreporting.blogspot.com/2023/07/knowledge-graph-system-for-financial.html
[133] General Purpose Financial Reporting, https://digitalfinancialreporting.blogspot.com/2023/02/general-purpose-financial-reporting.html
[134] XBRL + SBRM = Improved BI, https://digitalfinancialreporting.blogspot.com/2023/08/xbrlsbrm-improved-bi.html
[135] Graph Query Language, https://en.wikipedia.org/wiki/Graph_Query_Language

PROLOG (modern ISO standard PROLOG[136]) or other such similar robust standards-based mechanisms.

A knowledge assembly is a set of knowledge graphs. A knowledge graph is a machine-readable structured representation of knowledge (semantics) related to a particular area of interest. So a knowledge assembly is a machine-readable network[137] of things and relations between things. The things and relations are classified or grouped in helpful/useful ways. Semantics is the science of giving meaning to data. Knowledge assemblies are about semantics which is data in context, a.k.a. information. Knowledge = ontology (things and relations between things) + rules (assertions, restrictions, constraints). A knowledge assembly can be explained using a logical theory or logical schema that verifies/validates the knowledge assembly. Knowledge assembly terminology is grounded in the more approachable and innately understandable terminology of logic and philosophy[138], not the technical jargon/terminology of computer science.

What always seems to be necessary to work with some machine-readable knowledge assembly is:

1. Some sort of database to store the knowledge within.
2. A logical model that is used to understand and process the information within that knowledge assembly.
3. Some sort of reasoning engine or semantic reasoner or rules engine that understands the logical model (#2) and processes the information in the database (#1) to give you the answers that you need from the knowledge stored in the knowledge assembly.

This capability, when implemented effectively, is an incredibly useful tool. This can be a general tool, think semantic spreadsheet or a specific tool for financial reporting. But you need to make it work like a well-oiled machine for the capability to be useful.

4.14. Knowledge Portal (Accounting Oracle Machine)

An oracle is a person or agency, like a software application, considered to provide wise, insightful, useful information or counsel or perhaps useful simulations or predictions. For example, a Certified Public Accountant (CPA) provides accounting and business related advice on specific topics; a CPA is an example of an oracle: a trusted business adviser.

An oracle machine can be thought of as a Turing machine connected to an oracle of machine-readable information and rules. The oracle, in this context, is a software application capable of solving some computational problem (logical, mathematical), which for example may be a construction problem, a decision problem, or a function problem. Think rules-based expert system for very precise and accurate information and counsel or maybe a ChatGPT "copilot" type thingy that is more probability based, but helpful.

---

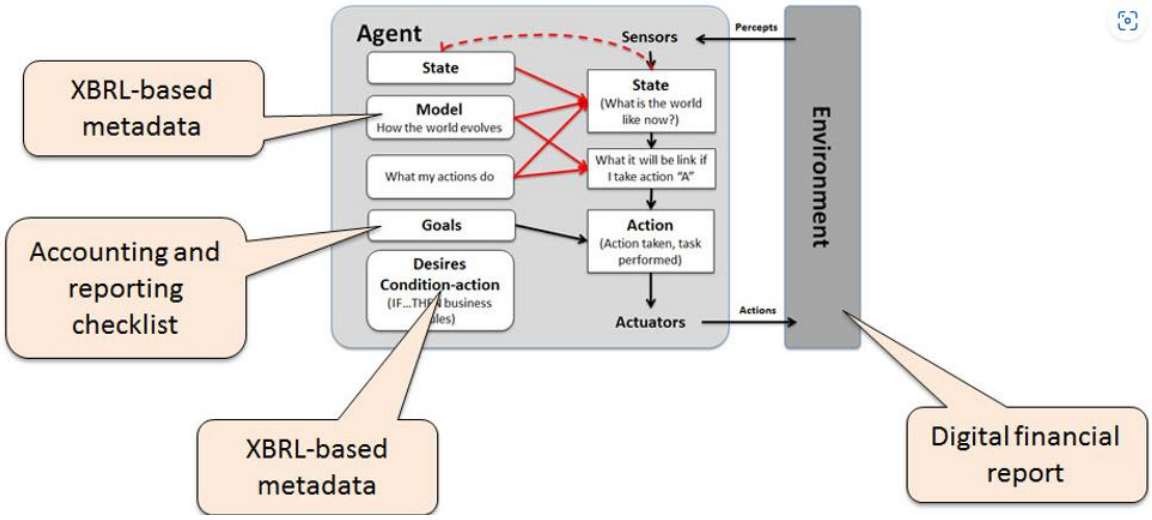
[136] Modern Prolog, https://digitalfinancialreporting.blogspot.com/2023/05/modern-prolog.html

[137] Wikipedia, Network, https://en.wikipedia.org/wiki/Network_theory

[138] Simple Explanation of a Logica System, https://digitalfinancialreporting.blogspot.com/2023/06/simple-explanation-of-logical-systems.html

If you don't understand rules-based expert systems or intelligent agents; it is really time to learn. There is ZERO PROBABILITY that artificial intelligence will have no impact on accounting, reporting, auditing, and analysis. Exactly WHAT impact is still to be determined, but trust me; the impact will not be "nothing".

Here is a diagram that explains what an intelligent software agent might look like and do: (to understand more details, read the information provided above)



4.15. Knowledge Hive (Semantic Hive or Hive Plot)

A sematic hive or hive plot[139] is is a group that has a similar view and have similar "ontological commitment" or "knowledge commitment". Effectively, each "semantic hive" or "hive plot" is mutually exclusive: you belong to one semantic hive or another semantic hive, you cannot belong to both because that would be illogical.

Here is an example. Say you wanted to create a knowledge graph of the semantics of the U.S. Constitution. If you tried to get everyone to agree with one single knowledge graph, this could be quite a challenge. But, if you broke up the group of people trying to do this into "semantic hives" or "hive plots" such as "democrat" and "republican" and/or "conservative" and "liberal" figuring out what goes into that knowledge graph becomes significantly easier. Then, you could use the semantic hive or hive plot created by the group you subscribe to most to perform reasoning.

4.16. Semantic Web

Digital distributed ledgers, smart contracts, nonfungible tokens (NFTs), the Interplanetary File System (IPFS), decentralized autonomous organizations (DAOs), and other such technologies will be used to implement interesting new capabilities.

4.17. Knowledge Products

Now, think less of logical digital twins and try and think more about what you can do with a properly functioning logical digital twin. Knowledge products offer completely

[139] Semantic Hive or Hive Plot, https://digitalfinancialreporting.blogspot.com/2024/01/notion-of-semantic-hive-or-hive-plot.html

new business models[140]. Imagine represented your skills and abilities in machine-readable form so that a computer software application can perform work using your knowledge.

A **data product** is generally a reusable raw and unprocessed data asset, engineered to deliver a trusted dataset to a user for a specific purpose. It integrates data from relevant source systems, processes the data, ensures that it is compliant, and makes the data accessible to those with the right credentials. The focus of a data product is raw, unprocessed data which can be numbers, text, or some sort of measurement. Examples of a data product are stock market data, website traffic lots, social media posts. The data might be delivered in the form of a CSV file, spreadsheets, an API, an XML file, RDF, JSON.

An **information product** tends to provide processed information to the information product's user. Information products are often used to monetize knowledge. Information products are organized and interpreted data which provides context and meaning. Information is generally provided in a way that the information can be consumed such as reports, dashboards of information, articles, and in the future very likely machine-readable knowledge graphs of high-quality information.

A **knowledge product** is refined and actionable information that has been processed, organized, and/or structured in some way or put into practice in some way making the information super-useful. The information is ready to use. The knowledge is derived from expertise, research, lessons learned. Knowledge products allow the user of the knowledge product to make informed decisions or better decisions.

Knowledge products provide insights, best practices, good practices, expertise which was derived from information, skills and experience. Knowledge products tend to answer "why" and "how" types of questions. Knowledge products can be delivered in a wide variety of forms such as training material, templates, guides, checklists, tutorials, artificial intelligence models, spreadsheet models. In the future knowledge products will, I believe, include machine-readable knowledge graphs that, preferably, use a global standard format and are proven to be of high quality per a logical schema.

Knowledge products empower their users, and the user could be a human or a software application such as an expert system for creating financial reports or your tax return. AI-powered software agents or assistants need machine-readable knowledge to operate. Subject matter experts in some area of knowledge will create these knowledge products.

An example of a knowledge product is an accounting oracle machine which I have mentioned before here and here.

Knowledge products are actionable; they should equip the users of the knowledge product to make informed decisions or take effective action in the moment. This could be in the form of recommendations of some sort, knowledge about best practices, or maybe a checklist or step-by-step guide. Knowledge products are driven by the skills and expertise and deep understanding gained within some specific field. Knowledge products do not provide a surface level understanding, they offer true insights and understanding.

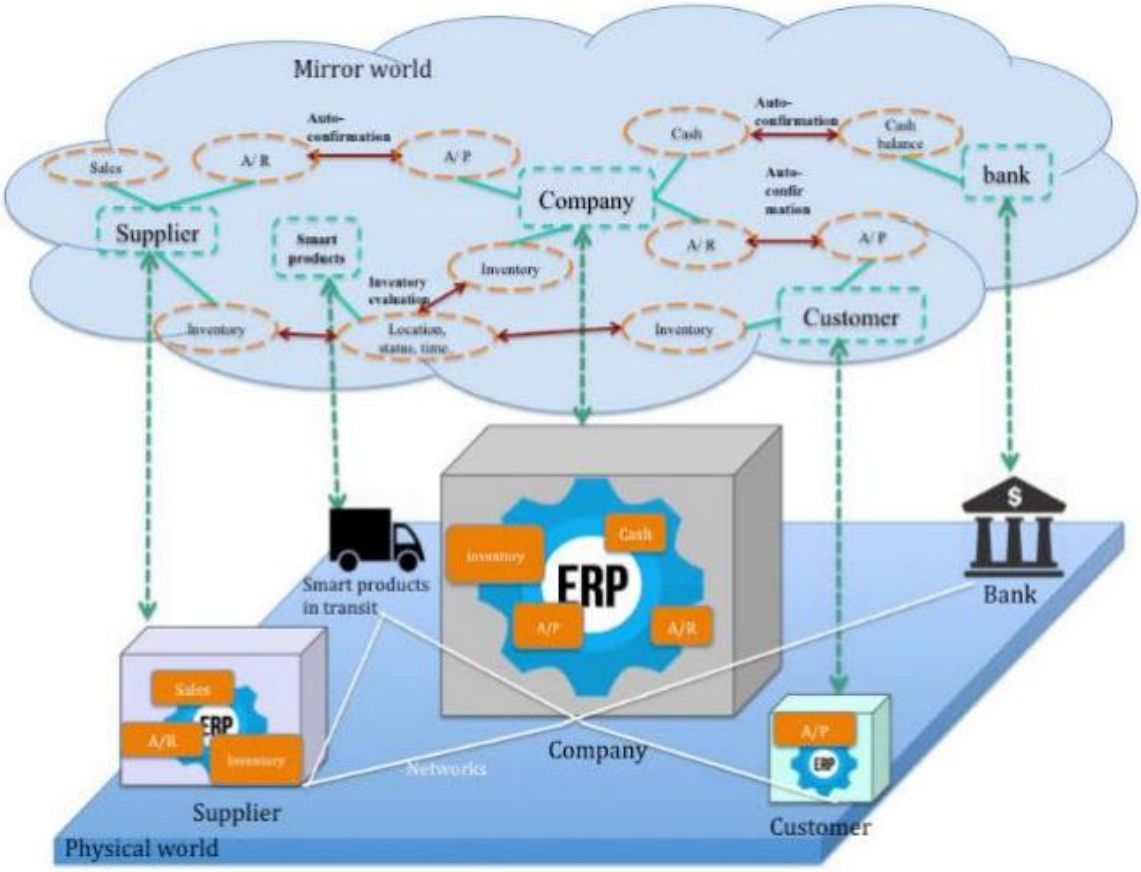
---

[140] *Knowledge Products offer New Business Models*, https://digitalfinancialreporting.blogspot.com/2024/02/a-new-business-model-is-emerging-which.html

Knowledge products tend to be tailored to the audience or a specific group of users and address the needs and challenges of that audience or group. Knowledge products might teach new knowledge. They might provide expert consultations to tap into the expertise of specific individuals or organizations. They could provide personalized recommendations or solutions to very specific problems.

4.18. Digital Twin for Financial Status and Performance of Economic Entity

Now, imagine a digital twin that is created using this global standard knowledge assembly[141].



4.19. Logical Digital Twin for Financial Reporting

And that leads us to the notion of a logical digital twin specifically designed for financial reporting. Such a logical digital twin is a knowledge graph of professional quality that is specific to an area of knowledge, financial reporting in our case, that has a logical schema that enforces the logic represented by the knowledge graph.

This logical digital twin might be represented in a number of different technical formats such as XBRL, RDF+OWL+SHACL, PROLOG, DATALOG, GSQL, SQL or some other technical format.

[141] Digital Twin for Financial Status and Performance of Economic Entity, https://digitalfinancialreporting.blogspot.com/2023/09/digital-twin-for-financial-status-and.html

Using a machine-understandable logical digital twins represented using a global standard technical syntax such as XBRL or RDF; it is possible to describe quantitative and qualitative associations between facts or sets/assemblies of facts within a report, such as a financial report or other type of business report. Such associations can have any degree of complexity. This information can then be reasoned on using a logic engine such as DATALOG which is an implementation of nearly a complete set of first order logic (i.e. some risky capabilities were removed in order to guarantee that catastrophic logical failures caused by logical paradoxes do not occur and therefore the processing is certain and reliable).

5. Common Impediments to a Properly Functioning Logical System

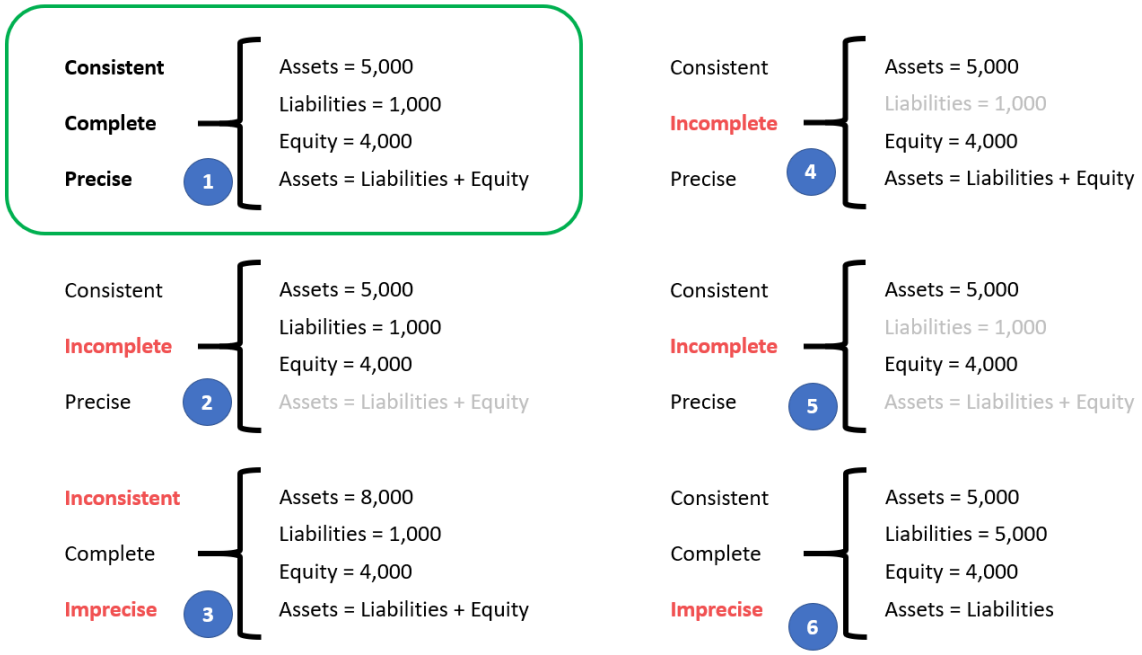
A good way to understand the notion of a "properly functioning logical systems" is to understand the impediments to a properly functioning logical system; what can cause a system to function improperly.

In this section we will start with a very simple logical system, the accounting equation which was described above, show the properly functioning version of that system, and then describe several states where that logical system becomes an improperly functioning logical system.

5.1. Properly and Improperly Functioning Logical Systems

Below you can see examples of each of six possible states of the accounting equation logical system. For example, the logical system can be functioning improperly if a single statement is left out, if one statement contradicts another statement within the logical system, if a statement is imprecise with respect to reality; all of these situations impact (a) the logical system and (b) what information is necessary to include within the logical system.

Here is a graphic depicting the first 6 states including the first which is a properly functioning logical system:



In the following sections I want to make some adjustments to the logical system which make the logical system either inconsistent, incomplete, or imprecise and explain why the system is then not a properly functioning logical system. To the six examples above I will add three additional examples. I made videos that explain each of these impediments to a properly functioning logical system which you can see in this video playlist, Understanding the Financial Report Logical System[142].

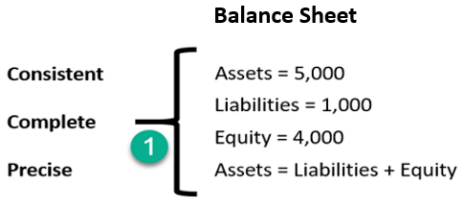
Before we get to the improperly functioning logical systems, let's take one final look at the properly functioning logical system so that you can compare and contrast the properly functioning and improperly functioning logical systems.

5.2. State 1: Properly Functioning Logical System

For completeness, I want to start by mentioning again our properly functioning logical system which is consistent, complete, and precise. It can be helpful to contrast other states to this state to understand the difference between properly functioning logical systems and improperly functioning systems.

Balance Sheet [Abstract]	Period [Axis]
	2020-12-31
Balance Sheet [Abstract]	
Assets	5,000
Liabilities	1,000
Equity	4,000

Result	Rule
Pass	$Assets = $Liabilities + $Equity



Again, this is considered a properly functioning logical system because (a) all the statements within the system are **consistent**; (b) the set of statements that

[142] Understanding the Financial Report Logical System, https://www.youtube.com/playlist?list=PLqMZRUzQ64B7EWamzDP-WaYbS_W0RL9nt

describe the system is **complete**; and (c) the information conveyed by the system is **precise** in its representation of reality. Further, we are formally declaring this "reality"[143] to be our base understanding.

Also, we need to be explicit. We defined three terms "Assets", "Liabilities", and "Equity".

Now, you may know what those three terms are; but a computer does not. You have to define what you work with relative to something that you know. Imagine our system defines four terms, "fac:Assets", "fac:Liabilities", "fac:Equity", and "fac:LiabilitiesAndEquity"[144]. You understand your system but you have to map every external system into your system[145]. Your internal system understands more that the accounting equation system (i.e. you have LiabilitiesAndEquity). You have to be able to compute that value based on some other system's information[146]. It is perfectly reasonable for our system to create a concept LiabilitiesAndEquity and compute that value even though the accounting equation logical system does not have that explicit value.

The point is that different economic entities have different models; but all models of a financial reporting scheme are reconcilable from/to one another in some manner[147].

5.3. State 2: Incomplete Coverage by Rules

The logical system #2 below is intended to show exactly the same information as our #1 properly functioning logical system, except that #2 leaves out the rule "Assets = Liabilities and Equity" which is showed as grayed out (i.e. because it is assumed to be missing from the logical system.

Coverage is a measure of how well you do or can represent a domain of information within a logical system. "Do" is about using the tools you have correctly and effectively. "Can" is about the capabilities of the tools you are using to represent the rule.

For example, if your logical system neglects to include the rule "Assets = Liabilities + Equity" or if your tools don't provide the capabilities to allow you to represent that rule; then there is the possibility that the facts being represented to be represented incorrectly and the system will not detect the inconsistency. As such, that logical system has **incomplete coverage**.

---

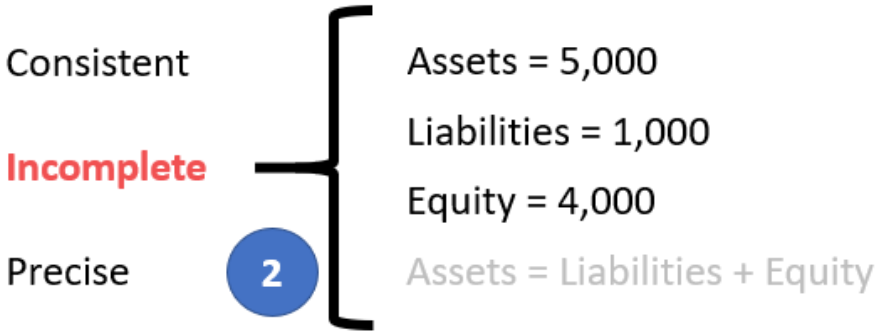
[143] YouTube, *Reality*, https://youtu.be/eq2Jw6waaCI

[144] Fundamental accounting concepts, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac.xsd

[145] Mapping from accounting equation to fundamental accounting concepts in our system, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml

[146] XBRL Formula to derive the value for LiabilitiesAndEquity, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-ImputeRule-LiabilitiesAndEquity-formula.xml

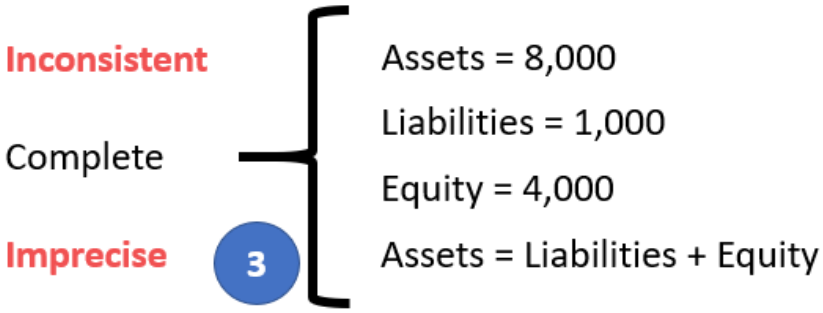
[147] Charles Hoffman, CPA, *Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements*, http://xbrl.squarespace.com/journal/2019/12/30/special-theory-of-machine-based-automated-communication-of-s.html



While this specific state #2 does have the Assets, Liabilities, and Equity facts consistent with the absent rule; the system is still incomplete because the coverage can be improved by adding the missing rule. If that missing rule is added, then the logical system can be considered complete again.

5.4. State 3: Inconsistent and Imprecise

All the statements in the system must be consistent for the logical system to be considered properly functioning. If statements are inconsistent, the logical system is not is not properly functioning. In this system #3, the values for Assets, Liabilities, and Equity are inconsistent with the rule "Assets = Liabilities + Equity". From looking at the information provided, it is impossible to know exactly which of the three facts are incorrect; it is only possible to understand that the statements made within the logical system is inconsistent. It could be the case that the rule is incorrect.



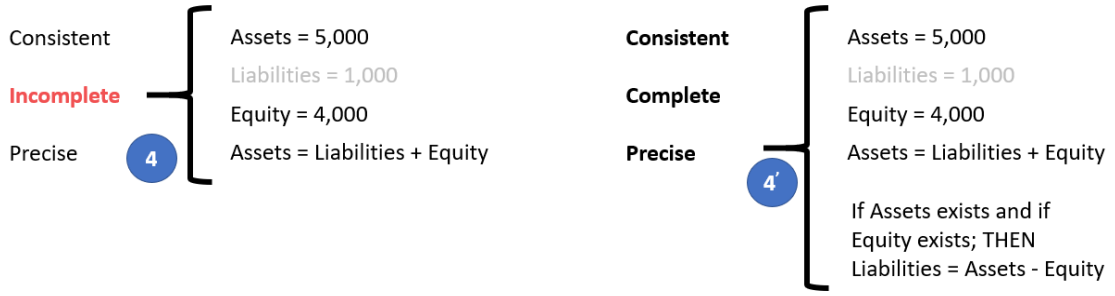
However, given that we know from state #1 that the value for Assets is 5,000 and not 8,000; the facts in this system is imprecise because the fact for Assets does not reflect reality.

5.5. State 4: Unreported Facts

In state #4, the situation is that the economic entity representing information in their report neglected to include the fact for Liabilities. Whether it is the case that a fact can, or cannot, be left unreported is a decision that can be made by the stakeholders of the system.

If it is the case that it is decided that the fact "Liabilities" can be omitted if both Assets and Equity are reported; then you must provide a rule to derive the value of Liabilities when that fact is not reported. Below you see that the system has been

adjusted in state #4' to add the rule "IF Assets exists and if Equity exists; THEN Liabilities = Assets - Equity"[148].



If it were likewise true that either Assets[149] or Equity[150] could also be left unreported, similarly derivation rules could be created for each of those facts. Note that XBRL Formula chaining[151] can be used to physically derive unreported facts if any one of these three facts remain unreported. Note that it is impossible to derive missing information if any two of the facts remain unreported. Adding the derivation rule makes the system complete.

Allowing certain line items of a report to go unreported specifies the need to create rules to derive missing information. Or saying this another way, omitting the possibility of unreported facts negates the need for creating derivation rules.

A second downside of allowing unreported facts is that you lose the parity check or cross check if facts can go unreported. Said another way, it would be considered best practice to not leave important high-level financial report line items to go unreported.

5.6. State 5: Incomplete

Similar to state #4, in state #5 the logical system is incomplete because both (a) the fact Liabilities is unreported and also (b) the consistency rule "Assets = Liabilities + Equity" is missing from the logical system. Because both a fact and the rule are missing from the logical system, it would be impossible to deduce the value of Liabilities in this case. There is not enough information in the logical system to allow Liabilities to be derived. At a minimum, a consistency crosscheck rule[152] plus the derivation rule to impute Liabilities[153] would be necessary.

---

[148] Here is the impute or derivation rule that would be added to the accounting equation logical system for this situation, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-1-Code-BS-Impute-01-formula.xml

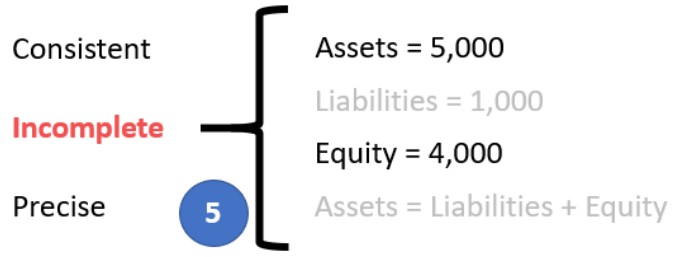
[149] XBRL Formula rule for deriving Assets, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-3-Code-BS-Impute-03-formula.xml

[150] XBRL Formula rule for deriving Equity, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-2-Code-BS-Impute-02-formula.xml

[151] Deriving Facts Using XBRL Formula Chaining (Example), http://xbrl.squarespace.com/journal/2019/4/24/deriving-information-using-xbrl-formula-chaining-example.html

[152] XBRL Formula consistency crosscheck rule Assets = Liabilities + Equity, http://xbrlsite.azurewebsites.net/2020/core/master-ae/Consistency-5-Code-BS01-formula.xml

[153] XBRL Formula derivation rule to impute Liabilities, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-1-Code-BS-Impute-01-formula.xml

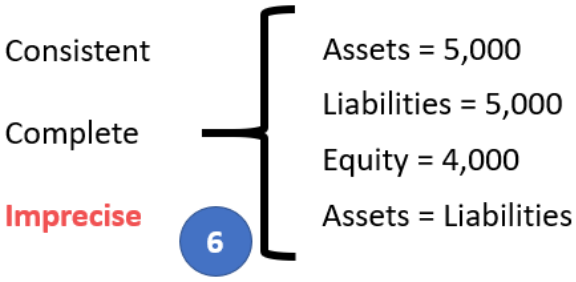


Again, consistent with state #4; Assets and Equity would require similar rules and there is no parity check of reported information.

5.7. State 6: Imprecise

A logical system is a true and fair representation of some agreed upon realism. **Precision** is a measure of how precisely you do or can represent the information of a domain within a logical theory. The reality that we formalized in state #1 indicates that "Assets = Liabilities + Equity". Yet, in the state #6 example, the rule "Assets = Liabilities" was provided. Further, the values of Assets and Liabilities are, in fact, consistent with the rule that has been provided.

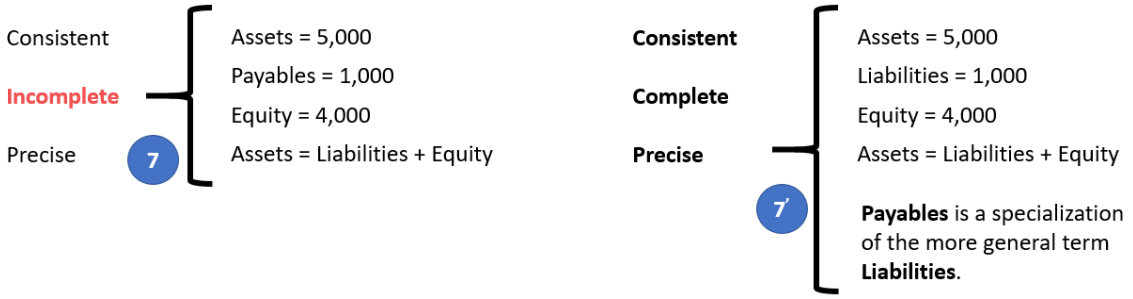
Remember that in state #1 we formalized our truth to be that "Assets = Liabilities + Equity". As such, this logical system can be described as being imprecise. To make this logical system precise, all that needs to be done is to fix the rule.



5.8. State 7: Extension Concept

In state #7 on the left, what we are trying to convey is that the economic entity reported the fact for Liabilities using the extension concept "Payables" that it had created. If a fact is represented using an extension concept created by a reporting entity; then a "general-special" or "wider-narrower" or "class-equivalentClass" association must be created to indicate to software applications of the relationship so that information can be used correctly. State #7' on the right, the rule "Payables is a specialization of the more general term Liabilities" has been added to the logical system which allows the system to operate effectively[154].

[154] XBRL Definition relations showing example of a mapping rule, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml

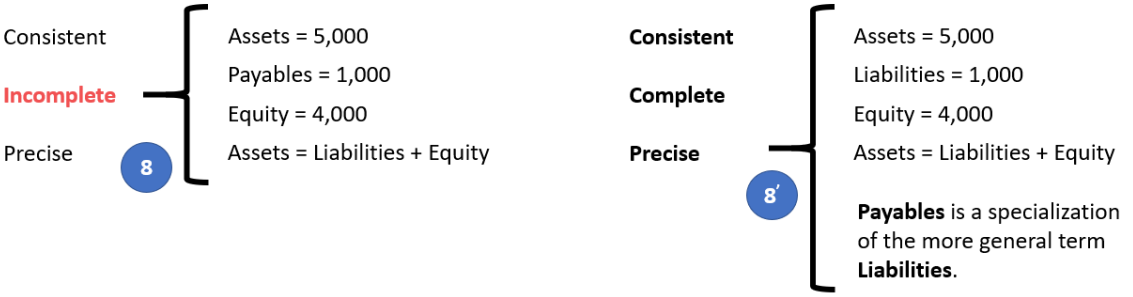


5.9. State 8: Base Taxonomy Wider/Narrower Concept Use

State #8 on the left below is similar to state #7 in that a different concept is used to report a fact; but while state #7 focuses on using an extension concept; state #8 points out that using a wider or narrower base taxonomy concept gives exactly the same result.

Now, our base state #1 does not have the concept "Payables"; but let's assume for a moment that it does have the concept "Payables". Also suppose that there was no information in the base logical system indicating the relationship between "Payables" and any other concept. If a fact is represented using a BASE TAXONOMY CONCEPT by a reporting entity; then a "general-special" or "wider-narrower" or "class-equivalentClass" association must exist in that base taxonomy to indicate that some concept is a permissible alternative for some other concept.

State #8' on the right adds the rule "Payables is a specialization of the more general term Liabilities"[155].



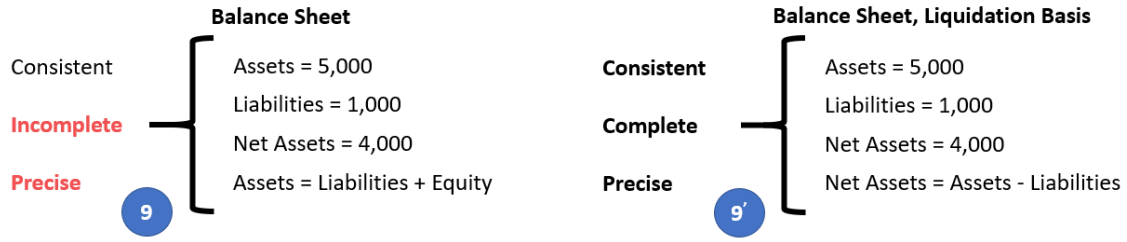
5.10. State 9: Defining a Completely New Structure

State #9 below on the left focuses on the structure as contrast all the prior examples which focused on the terms and rules. If a new structure is created, the new structure must be referenced to the base taxonomy and the new structure needs to be explained using machine-readable rules[156]. Even base taxonomy structures need

[155] XBRL Definition relations showing example of a mapping rule, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml
[156] XBRL Definition relations used to represent structure rules, http://xbrlsite.azurewebsites.net/2020/core/master-ae/dm-1355-rules-def.xml

to be defined in order to be referred to[157]. When you say "Balance Sheet" you know what that means. But a machine does not know.

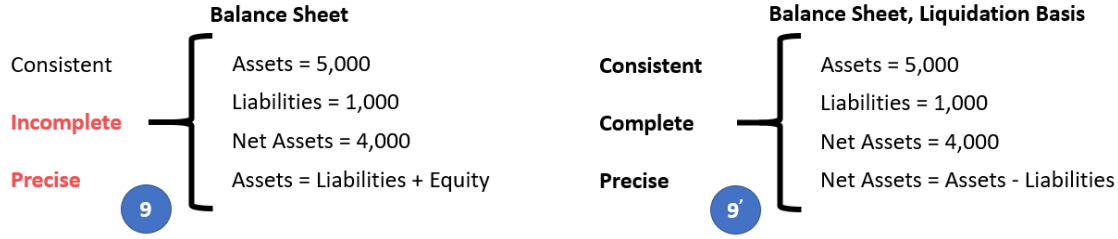
A base taxonomy should (a) provide all necessary structures separately, not intermingle different models in the same set of associations and (b) define what each structure must look like. Remember, computers are like babies and need to be led by the hand in order to understand the details you need them to understand.



Finally, in our case we have only one disclosure, the Balance Sheet. In our case, the Balance Sheet is always required to be reported per this logical system. As such, that rule is stated in a machine-readable reporting checklist[158]. Other logical systems with more disclosures will have more rules relating to when a disclosure is required to be provided in a report.

5.11. State 10: Organizing Disclosures Using Topics

State #10 points out that while the accounting equation logical system has one structure, the balance sheet, ultimately if a complete financial reporting scheme were represented one might have hundreds or even thousands of disclosures. Disclosures can be organized into topics[159]. Then, rather than having one flat list of disclosures, they can be organized into a handy hierarchy[160].



6. Further Reading

The following is additional reading which readers of this document might very likely find useful.

---

[157] XBRL taxonomy schema used to define "Balance Sheet", http://xbrlsite.azurewebsites.net/2020/core/master-ae/disclosures.xsd

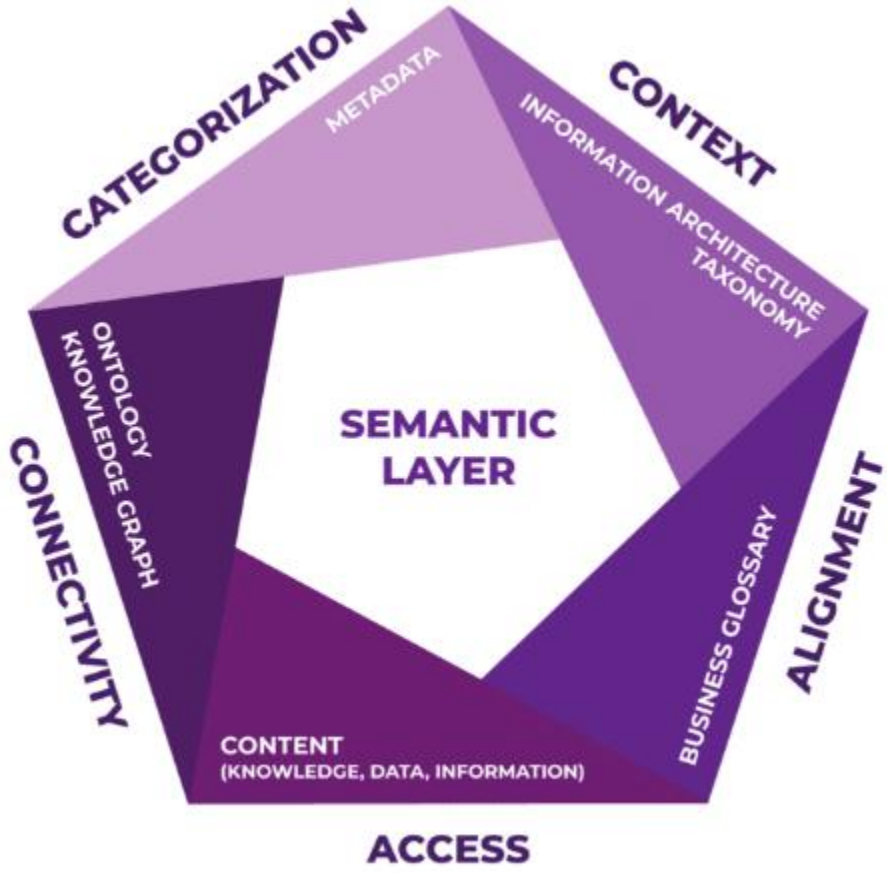
[158] XBRL Definition relations used to represent a reporting checklist or disclosure rules, http://xbrlsite.azurewebsites.net/2020/core/master-ae/reporting-checklist-rules-def.xml

[159] XBRL taxonomy schema used to represent topics, http://xbrlsite.azurewebsites.net/2020/core/master-ae/topics.xsd

[160] XBRL definition relations used to create a hierarchy of disclosures, http://xbrlsite.azurewebsites.net/2020/core/master-ae/disclosures-with-topics-def.xml

6.1. What is a Semantic Layer?

The article *What is a Semantic Layer*[161] explains a semantic layer to be the following. "A semantic layer is a standardized framework that organizes and abstracts organizational data (structured, unstructured, semi-structured) and serves as a data connector for data and knowledge."



6.2. Fast Healthcare Interoperability Resources (FHIR)

Fast Healthcare Interoperability Resources (FHIR)[162], per their website, is a next generation standards framework created by HL7. FHIR combines the best features of HL7's v2 icon, HL7 v3 icon and CDA icon product lines while leveraging the latest web standards and applying a tight focus on implement ability.

HL7 International is a leader in creating effective approaches to information exchange. There are a lot of good ideas to be found on their web site.

---

[161] Enterprise Knowledge, Lulit Tesfaye, *What is a Semantic Layer?*, https://enterprise-knowledge.com/what-is-a-semantic-layer-components-and-enterprise-applications/

[162] *HL7 FHIR is an Example of Leadership in Action*, https://digitalfinancialreporting.blogspot.com/2024/01/hl7-flir-is-example-of-leadership-in.html

Definitions | Formats | UML | XML | JSON | ND-JSON | RDF

2.1.6.0 Resource Definitions

FHIR Infrastructure Work Group	Maturity Level: Normative	Standards Status: Normative

This page documents how the content of the resources are described. In actual exchange, resources can be represented in the following formats: XML, JSON and Turtle. A UML Based Object-Oriented Definition is also provided.

2.1.6.0.1 Resource Definition

The resources are described in several different ways:

- a hierarchical table that presents a logical view of the content
- a UML diagram that summarizes the content graphically
- a pseudo-XML syntax that provides a visual sense of what the end resource instances will look like in XML
- a pseudo-JSON syntax that provides a visual sense of what the end resource instances will look like in JSON
- a pseudo-Turtle syntax that provides a visual sense of what the end resource instances will look like in Turtle

In addition to this descriptive syntax, other definitional forms are available, including W3C schema, Schematron, JSON Schema, and the StructureDefinition syntax defined internally.