CC0 1.0 Universal (CC0 1.0) Public Domain Dedication

Practical Knowledge Graphs in Practice

A nontechnical explanation of knowledge graphs where there is zero tolerance for errors in the context of financial statements for motivated professional accountants, auditors, analysts, and others with a liberal arts degree trying to wrap their heads around artificial intelligence and want to take an AI-first, Digital-First, Graph-First approach

By

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

August 30, 2026 (DRAFT)

https://seattlemethod.blogspot.com/

1. Introduction

Knowledge graphs are essential tools of the modern information age. As the Fourth Industrial Revolution drives our transition from an industrial economy to a digital economy, the tools we use must evolve to meet the demands of our current times.

Specifically, this approach applies to using knowledge graphs in practical, zero-tolerance-for-error environments. Compliance reporting represents one such critical use case; because most business professionals are already familiar with general-purpose financial statements, this common reporting model is used to provide concrete examples.

But what exactly is a knowledge graph[1], and how does it connect to digital transformation and artificial intelligence?

At its foundation, every machine-readable knowledge graph is built upon a machine-readable model. This model is a collection of interlinked logical statements that explicitly define the elements, structures, connections, conditions, observations, and measurements that make up the knowledge graph. By providing both a clear conceptualization to explain the graph and a schema to verify its construction, the knowledge model puts information into context. Through this combination of rich linking and logical metadata, knowledge graphs deliver a robust framework for integrating data, solving complex problems, analyzing information, and seamlessly sharing machine-readable knowledge.

When you try and understand knowledge graphs you often tend to run across explanations in terms of products that are offered. Three excellent resources that explain knowledge graphs in terms of specific product implementations include:

- *The Knowledge Graph Cookbook: Recipes that Work*[2] (uses the W3C standard Semantic Web Stack; this approach uses RDF,

[1] Ontotext, *What is a Knowledge Graph?*, https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/

[2] *The Knowledge Graph Cookbook: Recipes that Work*, http://xbrl.squarespace.com/journal/2021/6/27/the-knowledge-graph-cookbook-recipes-that-work.html

SWRL, OWL, N3, SHACL, SPARQL, and other such W3C standard technologies for representing data, ontologies, and rules[3])

- *Graph Databases*[4] (uses the Neo4j graph database and graph compute engine, which will very likely become an ISO standard graph query language[5])
- *Systematic Introduction to Expert Systems*[6]*: Knowledge Representation and Problem-Solving Methods* (uses programming logic, PROLOG, for which is an ISO standard[7])

A significant portion of the information in this resource comes from those three excellent resources. If you want additional details after reading this document, those three books are worth looking into. Those three resources provide information about implementing knowledge graphs within specific software applications.

Either of these three problem solving paradigms can be employed effectively to represent financial statements. Further, the global standard XBRL[8] technical syntax can be bi-directionally serialized into or out of any of these three paradigms.

Most business professionals understand financial statements. A financial statement is a knowledge graph. I will explain what I mean by that in this document. A financial statement is also a type of logical system and a specialization of a business report. You could interact with financial statement knowledge graphs using general-purpose tools for processing any knowledge graphs.

But there are significant advantages to using special-purpose tools, tuned specifically for financial statement knowledge graphs, when you want to interact with a financial statement knowledge graph. These special-purpose tools are easier for business professionals to use and

---

[3] W3C, *Semantic Web*, https://www.w3.org/2001/sw/wiki/Main_Page

[4] *Graph Databases*, https://neo4j.com/neoassets/graphbooks/Graph_Databases_2e_Neo4j.pdf

[5] *New Query Language for Graph Databases to Become International Standard*, https://neo4j.com/press-releases/query-language-graph-databases-international-standard/

[6] Frank Puppe, *Systematic Introduction to Expert Systems: Knowledge Representation and Problem-Solving Methods*, https://www.google.com/books/edition/_/_kKqCAAAQBAJ

[7] ISO, *ISO/IEC 13211-1:1995 Information technology — Programming languages — Prolog — Part 1: General core*, https://www.iso.org/standard/21413.html

[8] XBRL International, *XBRL Standard*, https://www.xbrl.org/the-standard/

offer all the power of a general-purpose tool such as a graph database and a graph compute engine.

We will also look into the connection between knowledge graphs and artificial intelligence in this resource. Knowledge graphs basically supercharge artificial intelligence.

To create a specialized logical conceptualization of a financial statement from a general logical conceptualization of a knowledge graph we use three steps:

1. Logical theory or logical system
2. Logical conceptualization of a business report which builds on and is a type of logical theory/logical system.
3. Logical conceptualization of a financial statement which builds on but is a type of business report.

The logical conceptualization of a financial statement is knowledge that can be stored in the form of a graph. As such, a financial statement is a knowledge graph.

All this will be explained in this resource. Let's start by breaking down the terms "knowledge" and "graph".

1.1. Big Picture

A knowledge graph, also known as a semantic network, represents a network of real-world entities; such as objects, events, situations, or concepts; and illustrates the relationships between them in a visual graph structure or perhaps a dynamic software application[9] or a static document[10] or maybe a dynamic web page application[11].

In simpler terms, it is a set of "things" (represented as nodes) and the connections or associations between those things (represented as lines). At the core of every machine-readable knowledge graph is a knowledge model, which is a collection of interlinked logical statements describing the elements, structures, connections, conditions, and facts that make

---

[9] Example software application, https://luca.pacioli.ai/luca/view/9175b4014583e9fb8113bc253bf24aa764caa86880eec558111ca1cb0f8b1f67c119db37/index

[10] Example static document, https://xbrlsite.azurewebsites.net/seattlemethod/platinum/ifrs-smes/ref-model/inline.html

[11] Example dynamic web page application, https://xbrlsite.azurewebsites.net/seattlemethod/platinum/ifrs-smes/ref-model/ixbrl-report-viewer.html

up the graph. This model provides a conceptualization to explain the graph and a schema to verify its construction. By linking metadata, knowledge graphs put information into context, providing a solid framework for data integration, problem-solving, and information analysis.

Knowledge graphs essentially supercharge artificial intelligence. For AI to perform to its full potential or succeed, information originally created for human interpretation must be correctly interpretable by machines. Because traditional human-oriented documents can be highly arbitrary, machines struggle to interpret them accurately. Knowledge graphs solve this by providing a coherent, consistent, and structured representation of reality that both humans and machines can interpret in the exact same way.

Furthermore, intelligent software agents can use knowledge graphs as persistent, structured external memory. This acts as an extension of human reasoning and dramatically reduces the cognitive penalty associated with imperfect recall. Ultimately, this framework enables unprecedented human-machine collaboration, allowing machines to assist with the precise steps and tasks they do best while freeing humans to focus on what they do best.

1.2. Simple Overview

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.

Knowledge[12] 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 mathematics, a graph is a structure which is made up of a set of objects which are related in some way[13]. In mathematics and computer science

---

[12] Seattle Method, *Knowledge*, https://seattlemethod.blogspot.com/2025/01/knowledge.html

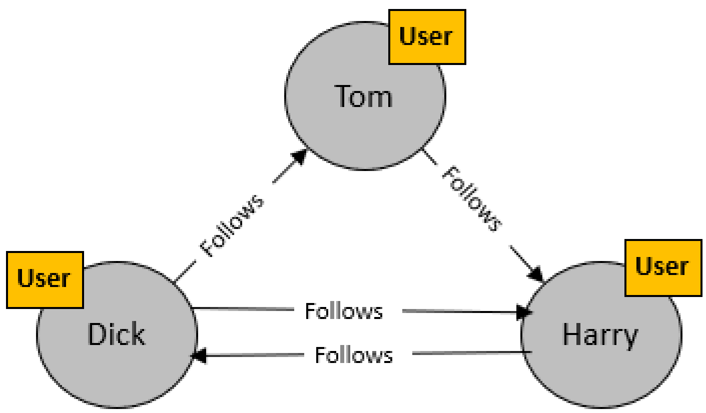
[13] Wikipedia, *Graph (discrete mathematics)*, https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)

a graph is explained by graph theory[14]. These formal explanations of a graph are important to getting graphs right.

In less intimidating language, a graph is a set of things[15] and the connections between those things. A node is used to represent a thing and lines are used to 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 connections (a.k.a. relationships, associations).

A graph is a general-purpose communications tool that allows us to model[16] an area of knowledge[17] 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[18], also known as a semantic network[19], represents a network[20] of real-world things (entities) — e.g. objects, events, situations, or concepts — and illustrates the relationship between them. This information can be visualized as a graph structure.

[14] Wikipedia, *Graph Theory*, https://en.wikipedia.org/wiki/Graph_theory
[15] Digital Financial Reporting, *Things*, https://digitalfinancialreporting.blogspot.com/2024/11/things.html
[16] Seattle Method, *Essence of Modeling*, https://seattlemethod.blogspot.com/2026/08/essence-of-modeling.html
[17] Seattle Method, *Area of Knowledge*, https://seattlemethod.blogspot.com/2025/01/area-of-knowledge.html
[18] IBM, *What is a Knowledge Graph?*, https://www.ibm.com/topics/knowledge-graph
[19] Wikipedia, *Semantic Network*, https://en.wikipedia.org/wiki/Semantic_network
[20] Wikipedia, Network Theory, https://en.wikipedia.org/wiki/Network_theory

Knowledge graphs are rich in terms of expressiveness but still innately interpreted by humans but knowledge graphs can also be read and interpreted by machines such as computers.

The point of all this is to work with the meaning[21] conveyed within the knowledge graph. To do that, we need to create a framework.

1.3. Communication

Communication is hard, be that communication be between two different humans, a human and a machine, or between two machines.

The strengths of computers and the obstacles that get in the way of using computers were summarized well by Andrew D. Spear in his paper *Ontology for the Twenty First Century: An Introduction with Recommendations*[22]; here is his list summarized into bullet points with some modifications made by me:

Fundamental strengths/capabilities of computers:

- store information reliably and efficiently
- 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 anytime and 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

---

[21] Digital Financial Reporting, *Meaning*, https://digitalfinancialreporting.blogspot.com/2026/02/meaning.html

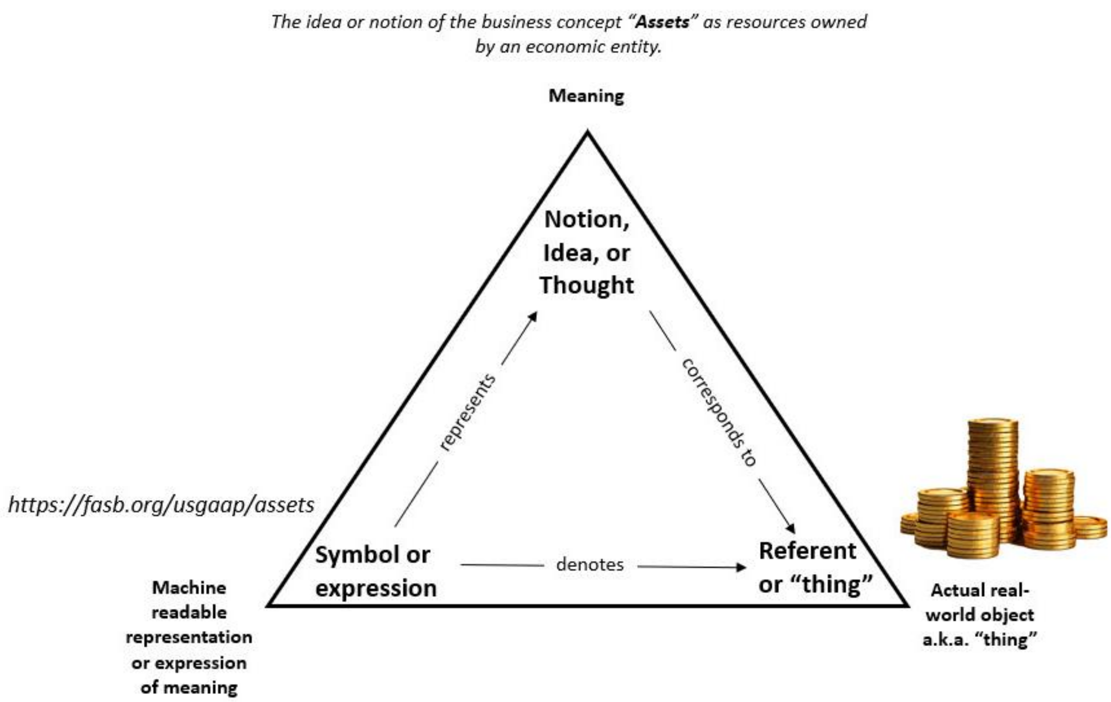
[22] Andrew D. Spear, *Ontology for the Twenty First Century: An Introduction with Recommendations*, https://www.semanticscholar.org/paper/Ontology-for-the-Twenty-First-Century%3A-An-with-Spear/14c2a7436182b88ad490868bb4b0ba75ab91527e

- inconsistent domain understanding of and technology's limitations in expressing interconnections within an area 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

Computers are machines. Computers do not understand, they only interpret. And when computers cannot interpret what they read, then humans need to help the computer out by providing a mapping between works that mean the same thing or to convert information from one technical format to some other technical format.

1.4. Triangle of Meaning

The triangle of meaning[23] is an important idea that helps one trying to understand what a knowledge graph is and how to build one. There is no single version of the triangle of meaning, but all tend to communicate the same fundamental idea. That fundamental idea is that there is a difference between the referent or "thing" being referred to in the real world, the implementation (a.k.a. representation) of that referent/thing, and the conceptualization of the meaning of the referent/thing.



[23] Seattle Method, *Triangle of Meaning*, https://seattlemethod.blogspot.com/2026/03/triangle-of-meaning.html

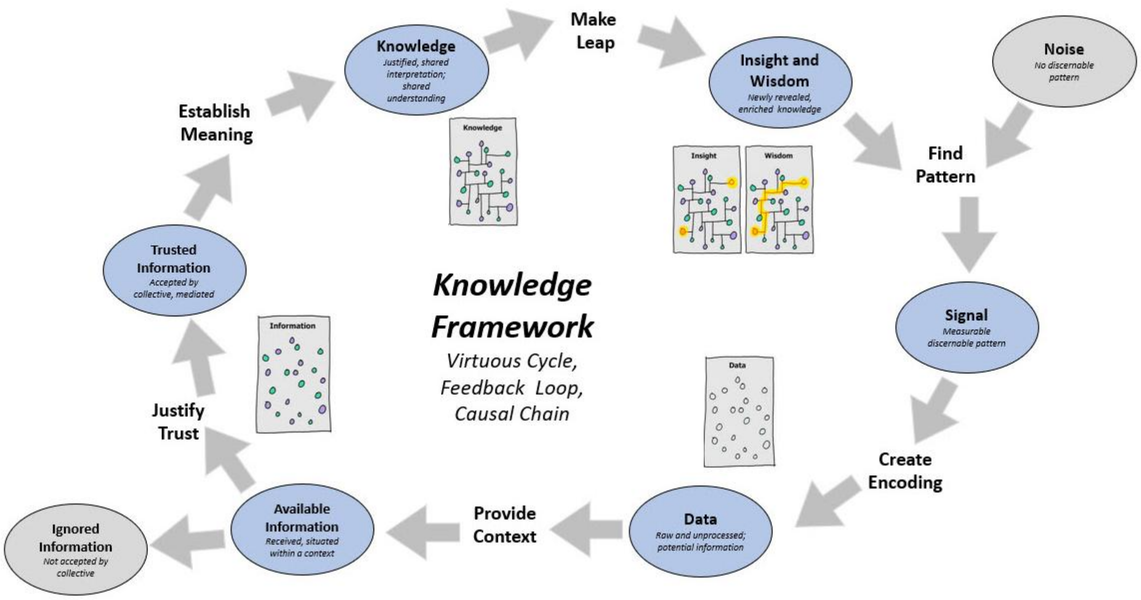
Fundamentally, there is a difference between "information" and the "costume[24]" that information "wears" which might be a document, an electronic spreadsheet, a relational database, a labeled property graph, or something else.

1.5. Frameworks

A framework[25] is stabilizing scaffolding or essential supporting structure which provides leverage and scalability.

A framework is a structured set of guidelines, libraries, conventions, components, best practices, and tools that provides a foundation for building and stabilizing applications or systems. Frameworks are useful because they save time, enforce consistency, and avoid reinventing solutions to common problems.

Think of a framework like the foundation of a house. Instead of starting with raw materials every time, you begin with a solid base and predefined structures that make construction faster and more reliable.



24 Digital Financial Reporting, *Document is a "Costume" a Financial Statement Wears*, https://digitalfinancialreporting.blogspot.com/2026/02/document-is-costume-financial-statement.html

25 Seattle Method, *Framework*, https://seattlemethod.blogspot.com/2025/11/framework.html

1.6. Example of Knowledge

A general-purpose financial statement is an example of knowledge[26]. A general-purpose **financial statement** conveys knowledge about the financial status, financial performance, and liquidity of an economic entity which created the financial statement. For example, here is a fragment of a financial statement:

	Period [Axis]		
Statement [Line Items]	2016-07-01 - 2017-06-30	2015-07-01 - 2016-06-30	2014-07-01 - 2015-06-30
Net income	21,204,000,000 [2,3]	16,798,000,000 [1]	12,193,000,000
Other comprehensive income (loss):			
Net unrealized gains (losses) on derivatives (net of tax effects of $(5), $(12), and $20)	(218,000,000)	(238,000,000)	559,000,000
Net unrealized losses on investments (net of tax effects of $(613), $(121), and $(197))	(1,116,000,000)	(228,000,000)	(362,000,000)
Translation adjustments and other (net of tax effects of $9, $(33), and $16)	228,000,000	(519,000,000)	(1,383,000,000)
Other comprehensive loss	(1,106,000,000)	(985,000,000)	(1,186,000,000)
Comprehensive income	20,098,000,000	15,813,000,000	11,007,000,000

The information contained within a financial statement can also be seen as or represented as a knowledge graph that is readable by both machines and by humans. While the formatting of the information within a financial statement is different than the formal vertices and edges of a graph; I think that it is rather easy to see or perceive that a financial statement is a knowledge graph.

What is useful when representing something that can be quite complex[27] in terms which are easier to understand is to use high level models.

1.7. Atomic Design Methodology

Atomic Design Methodology[28] is an approach to designing something. Originally contemplated for designing web pages, this approach is quite useful in thinking about knowledge graphs. I am not going to go into details here, but the fundamental idea is to abstract away the technical oriented "atoms" by using easier to understand "molecules", "organisms", and "species".

---

[26] Digital Financial Reporting, *Example of Financial Statement Holon*, https://digitalfinancialreporting.blogspot.com/2026/05/example-financial-statement-holon.html

[27] Seattle Method, *Complexity*, https://seattlemethod.blogspot.com/2025/01/complexity.html

[28] Seattle Method, Atomic Design Methodology, https://seattlemethod.blogspot.com/2026/03/atomic-design-methodology.html

Also fitting into this mental model is the notion of a holon[29]. A holon is something that is simultaneously a whole in and of itself, as well as a part of a larger whole. Holons are things that exist as both wholes and parts of larger systems at the same time. Holons have a dual nature in that they function independently by one set of rules while at the same time they contribute to the functioning of the larger system in which they exist. A hierarchy of holons is called a holarchy.

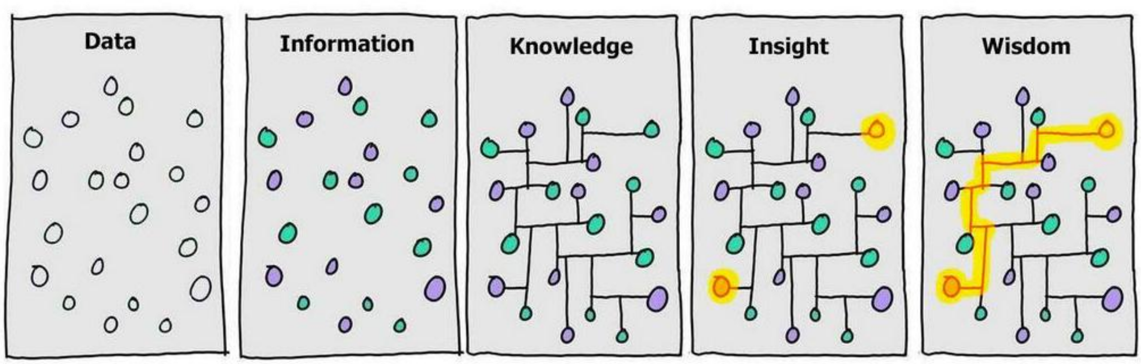
For now, just keep the notions of atoms, molecules, organisms, species, and holons in the back of your mind.

Let's start to expand on our understanding of knowledge graphs by looking a bit closer at the two terms: knowledge and graph.

1.8. Data, Information, Knowledge, Insight, Wisdom

Data, information, and knowledge are not the same things but often the terms are used interchangeably. For our purposes, we need a precise definition of each of these ideas.

The following graphic perhaps provides the best visual explanation as to the difference between data, information, knowledge, insight, and wisdom[30] that I have run across:



There are specific differences between data, information, knowledge, insight, and wisdom[31]:

- **Data**: The basic compound for intelligence is data. Data are measures, observations, symbols, phenomenon, utterances, and other such representations of the world around us presented as

---

[29] Digital Financial Reporting, Holon, https://digitalfinancialreporting.blogspot.com/2026/03/holon.html

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

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

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 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.

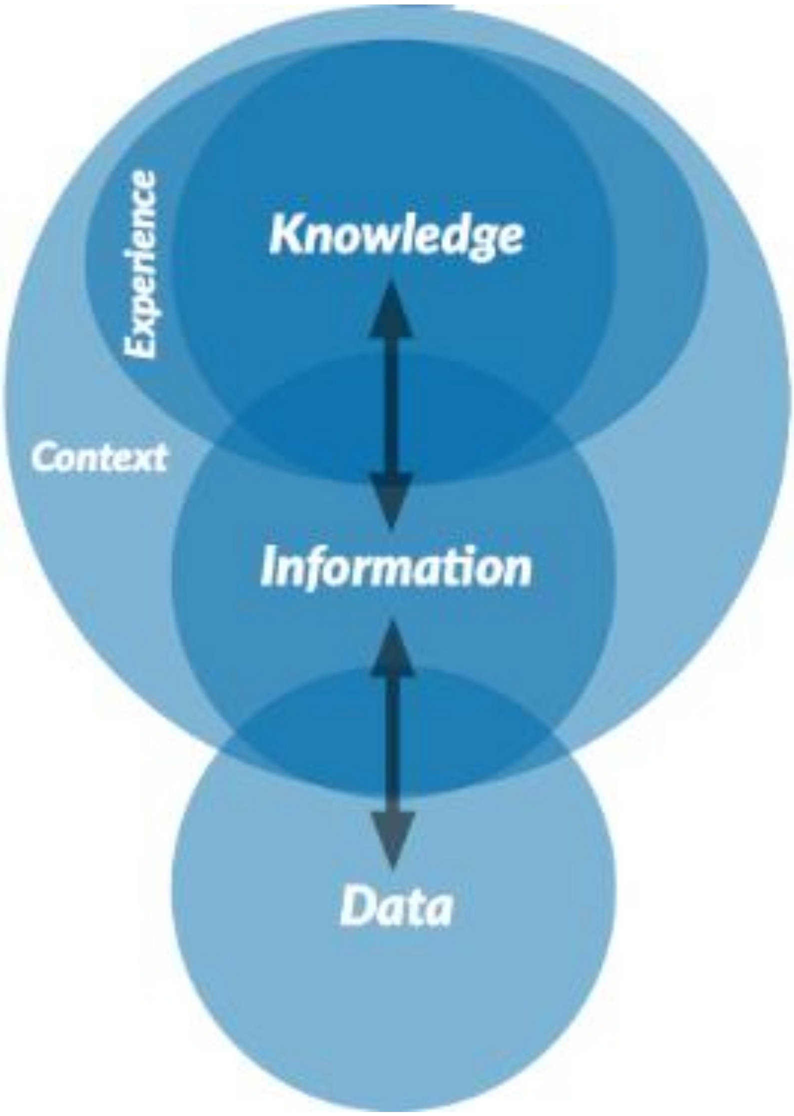
Something else that you need to keep in your mind is that information and knowledge are not static; they change. Processes change information and knowledge[32].

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.

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[33].

[32] Seattle Method, *Industrial Process*, https://seattlemethod.blogspot.com/2026/03/industrial-process.html
[33] Shawn Riley, *Machine Learning versus Machine Understanding*, https://www.linkedin.com/pulse/machine-learning-vs-understanding-shawn-riley/

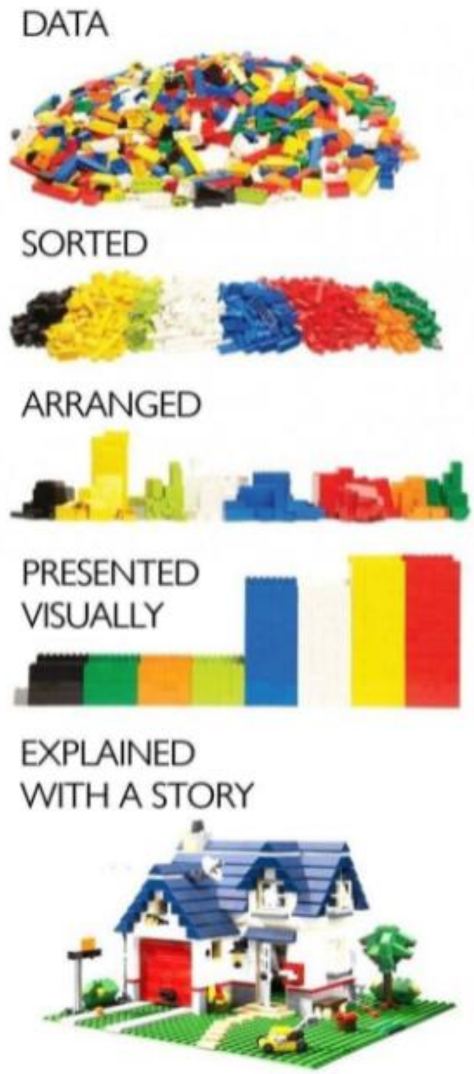


The important point to understand here is that it takes the skill and experience of human professionals to create information and knowledge and put that knowledge into the proper context.

The following graphic posted by Mark Cossey on LinkedIn[34] tries to show the difference between raw data and sorting, arranging, presenting, or putting the information together in the form of a story in order to explain

[34] LinkedIn, Mark Cossey, https://www.linkedin.com/feed/update/urn:li:activity:6839928291433029632/

the information. This helps one see how the tool of classification[35] is helpful:



The point that I am trying to make is that there is a very significant difference between data, information, and knowledge. Or focus is on information and knowledge, not data.

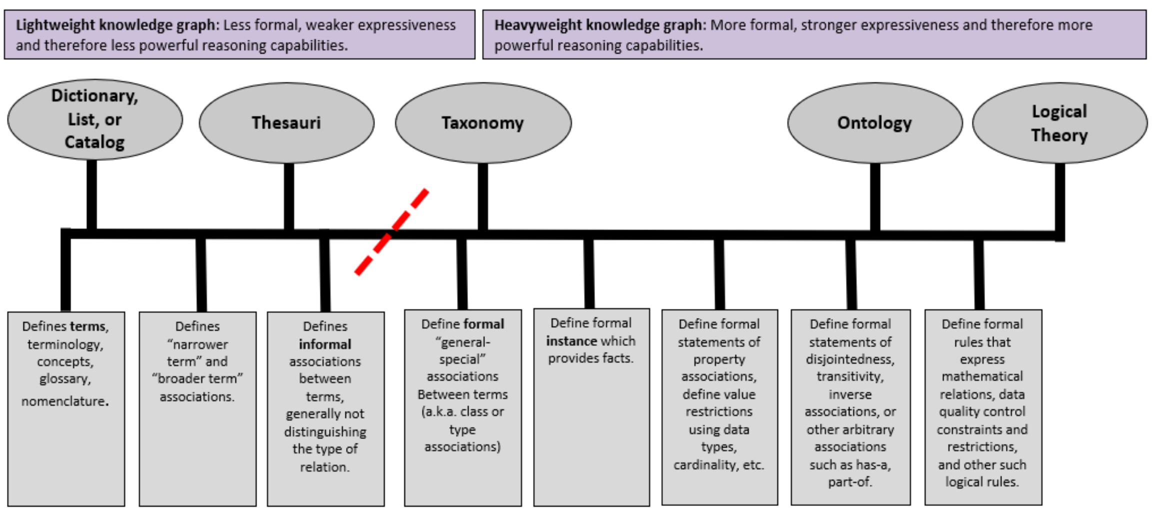
[35] Seattle Method, *Classification*, https://seattlemethod.blogspot.com/2026/06/classification.html

1.9. Conceptualization

A conceptualization[36] is a tool for creating a shared mental or collective understanding of what exists and how what exists hangs together in some area of knowledge (a.k.a. domain). A conceptualization is like a blueprint.

A conceptualization is intended to be understandable by humans (the community of subject domain stakeholders) but can also be represented in machine interpretable form. Conceptualizations are social artifacts and sociotechnical tools.

There are a number of different tools that can be used to effectively describe or represent knowledge in the form of a conceptualization[37]. Below you see a spectrum of such tools with the least powerful tools on the left and increasing in power to the right:



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

When representing knowledge, the right tool should be used for the job. You should look at these tools more in terms of a pipeline[38] than separate alternatives.

[36] Seattle Method, *Conceptualization*, https://seattlemethod.blogspot.com/2025/09/conceptualization.html
[37] Seattle Method, *Knowledge Representation Approaches*, https://seattlemethod.blogspot.com/2025/01/knowledge-representation-approach.html
[38] Digital Financial Reporting, Building out the Ontology Pipeline, https://digitalfinancialreporting.blogspot.com/2025/07/building-out-ontology-pipeline.html

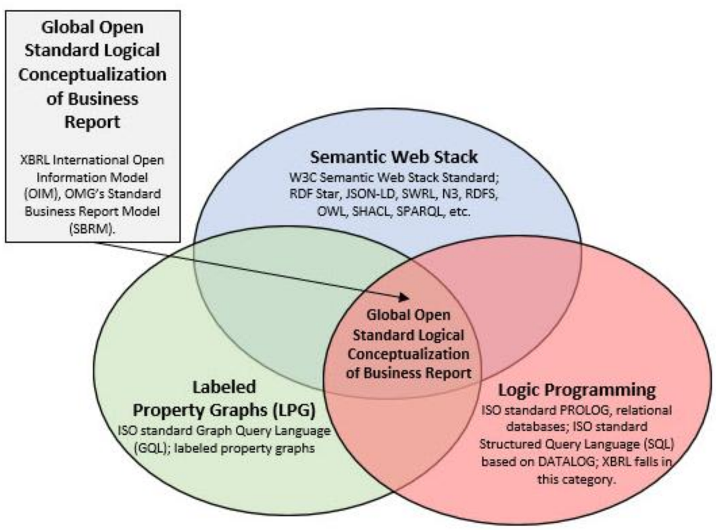
A logical theory[39] is the easiest way to enable business professionals to understand a logical system[40] because business professionals have an innate understanding of logic. When representing the logic of a financial statement, the power of a logical theory is necessary. In fact, many different theories can be combined together to describe a system completely[41].

1.10. Implementation

The approaches to physically implementing[42] a knowledge graph can be grouped into three general buckets. Those buckets are:

- Semantic Web Stack (a.k.a. Semantic Web Layer Cake): These are the standards published by the W3C.
- Labeled Property Graph (LPG): Graph databases that use the ISO standard graph query language (GQL).
- Logic Programming: Includes ISO standard PROLOG, SQL, XBRL.

These different physical implementation approaches can be converted from one physical implementation approach to the other two physical implementation approaches in many cases.



[39] Seattle Method, *Theory*, https://seattlemethod.blogspot.com/2025/12/theory.html
[40] Seattle Method, *System*, https://seattlemethod.blogspot.com/2025/01/system.html
[41] Seattle Method, *Complete*, https://seattlemethod.blogspot.com/2025/07/complete.html
[42] Seattle Method, Implementation, https://seattlemethod.blogspot.com/2026/03/implementation.html

Multiple technology architecture stacks is a fact of life. Different enterprises have different preferences and therefore make different choices as to their desired technical implementation approaches. That said, it is possible to transform between different physical implementation approaches.

1.11. Intelligence

Intelligence[43] is an idea that philosophers and scientists have been debating for thousands of years. To this day, there is no single, universally agreed-upon definition of intelligence. Four our purposes I am going to classify intelligence into two groups:

- Human-intelligence
- Machine-intelligence

I would contend that it is a fool's errand to try and compare machine-intelligence to human-intelligence. They are not even in the same ballpark and they won't be for a long time, if ever. The notion of "superintelligence" in computers is a fantasy. Computers are nowhere close to achieving superintelligence which I would consider as being human-level intelligence or higher levels of intelligence beyond even the capabilities of humans.

For our purposes here, my working definition of intelligence is as follows: Intelligence is the ability to acquire and apply skills and knowledge within a specific area. Skills are recognized patterns of actions and behaviors. Knowledge is facts and information. Crystalized intelligence is memory. Fluid intelligence is about processes and mechanisms for working with intelligence.

Basically, intelligence for our purposes here is defined as human task performance or “work”[44]. While many people are focused on artificial intelligence, they really should be more interested in computability.

1.12. Computability

In order to be on the same page; we want to be crystal clear with regard to several specific terms we are using to describe system complexity and computability.

---

[43] Digital Financial Reporting, *Intelligence*, https://digitalfinancialreporting.blogspot.com/2024/04/intelligence.html

[44] Seattle Method, *Work Tasks*, https://seattlemethod.blogspot.com/2025/01/work-tasks.html

Computability[45] is the ability for a computer to perform work within a system.

A **system**[46] is a set of elements, connections, conditions, boundaries, observations, measurements, algorithms, and interaction patterns between the different parts of the system. A conceptualization of a system can be explained using a theory[47].

Complexity[48] is the degree or level to which a system's elements, structures, conditions, and behavior depends on interactions that may, or may not, be cleanly decomposed into it understandable independent parts.

There tends to be three levels of complexity which have an impact on computability of a system: simple, complicated, and complex.

- **Simple system** (a.k.a. non-complex): Predictable and deterministic. The system is "simple" or "non-complex" and therefore computable; clear and obvious for a pretty much anyone including non-subject matter experts to understand, and the set of elements, connections, types/categories, conditions, boundaries, and interaction patterns are fully understandable and understood. Easy to understand control techniques can be used to eliminate all risk errors from the system.
- **Complicated system**: (a.k.a. not simple, but still non-complex) Predictable and deterministic. The system is "non-complex" and therefore computable; clear and obvious for a skilled and experienced subject matter expert in the area of knowledge to which the system relates to understand, and the set of elements, connections, types/categories, conditions, boundaries, and interaction patterns are still fully understood. Control techniques can still be used to eliminate all risk of errors from the system.
- **Complex system**: Not entirely predictable or unpredictable and therefore not deterministic. The system is "complex" and therefore NOT computable; tend to lack clear boundaries, tend to be constantly changing and evolving, there tend to be large

45 Seattle Method, *Computability*, https://seattlemethod.blogspot.com/2026/08/computability.html
46 Seattle Method, *System*, https://seattlemethod.blogspot.com/2025/01/system.html
47 Seattle Method, Theory, https://seattlemethod.blogspot.com/2025/12/theory.html
48 Seattle Method, *Complexity*, https://seattlemethod.blogspot.com/2025/01/complexity.html

numbers of elements, connections, types/categories, conditions and interaction patterns which are not completely understood, the system seems to contradict itself on occasion, and the number of forces impacting the system tends to be large and the dynamics are not well understood. Control techniques cannot be used to fully eliminate all risk from the system.

Note that some complex systems can have specific subsystems which are simple systems or complicated systems and if these systems can be separated then some aspects of a complex system can be made computable.

A complex system which can be "dumbed down" to a degree to enable the system to be computable but also adequately meet the goals and objectives of system stakeholders can be make both computable and perhaps useful to its stakeholders.

1.13. Artificial Intelligence

As a certified public accountant serving many different businesses, I’ve heard countless business owners say some version of, “*I run my business on the back of an envelope.*” And for many small, simple enterprises, that approach actually works; key numbers and other important information scribbled down on paper, the rest stored in the owner’s head.

But as a business grows, the “back-of-the-envelope” method begins to break down. Add a second location, introduce additional products, hire more employees, expand operations; complexity multiplies. Eventually the enterprise reaches a point where informal mental bookkeeping is no longer sufficient or safe.

At that stage, the business transitions from envelope-based thinking to a basic accounting information system. Maybe the owner buys QuickBooks or a similar basic bookkeeping system. This may be achieved by contracting with a Client Account Services (CAS) firm.

Soon after, they realize not everyone should see everything in QuickBooks, so they add a dashboard tool like BlueIQ or some other similar type of dashboard to give managers the specific information they need but without exposing information the owner prefers to keep private.

The first challenge is to get the right information into QuickBooks. That itself has challenges, but I don't want to go down that rabbit hole.

Then comes the next challenge: getting the right information out of QuickBooks and into BlueIQ, in the right form, at the right time, for the right people. As the organization grows; say 25 employees, two locations, five product lines; this becomes increasingly difficult.

This is why accounting systems, management systems, dashboards, org charts, workflow diagrams, spreadsheets, word-processed documents, emails, PDFs, and other artifacts exist. They are not created for the sake of only "documentation." They exist to extend and strengthen the envelope-based approach; so owners, managers, and employees can *operate the enterprise effectively*.

These artifacts, that documentation, improves shared understanding across the members of the organization. As the enterprise evolves, its information systems must evolve with it, shifting from static documents to living representations of how the business actually works.

Enter artificial intelligence.

AI can help owners, executives, managers, and employees stay aligned and operate the enterprise more effectively and in many different ways. But for AI to perform to its full potential or even to simply succeed, the information originally created for human interpretation must also now be correctly interpretable by machines; and perhaps often by humans and machines working together to perform their work tasks[49].

This raises the bar. AI requires a coherent, consistent representation of the enterprise; and preferably one that both humans and machines can interpret in the same consistent way. Getting two humans to agree on the "truth" of the business is hard enough. Now machines must also understand that truth, and do so reliably.

A faithful, justifiable representation of enterprise reality in machine interpretable form becomes a strategic capability. It strengthens operations, management, quality, other insights, and overall mission performance.

The informational artifacts of an enterprise are not the enterprise itself. For example, a financial statement of an enterprise is not actually the enterprise itself; it is simply a representation of some specific aspects of an enterprise per some specific model.

Historically, those documents were documents meant for human interpretation; enhancing the "back-of-the-envelope". But now those

[49] Digital Financial Reporting, *Reconfiguring the Economics of Human Memory*, https://digitalfinancialreporting.blogspot.com/2026/08/reconfiguring-economics-of-human-memory.html

documents must also support machine interpretation. But because how humans represent information in the form of a document can be quite arbitrary; machines have a hard time properly interpreting the information within those documents.

So how do we maximize the likelihood that AI succeeds; maximizing the potential of AI?

Two primary approaches tend to emerge:

1. Teach AI to read and correctly interpret all existing human-oriented documentation.
2. Redesign how documentation is created; structuring it so machines can interpret it easily, and then projecting that same information into human-friendly formats.

A third path blends both approaches, evolving toward a hybrid that works in practice.

Think of it this way. What might be better; (a) a business operating just as it has for the past 10 years or (b) a business taking advantage of artificial intelligence to create an advantage over competitors who are not using artificial intelligence?

Accounting and finance tend to commonly be a struggle for small and medium sized businesses. Many small and medium sized businesses lack the scale to employ a full-time team that includes a CFO, controller, cost accountants, bookkeeping staff, and others. Instead, they might have one or two employees trying to manage everything, those employees are usually over worked and sometimes things fall through the cracks. Or, in some cases, the business owner might manage accounting and finance by themselves. Sometimes a services firm is engaged to help out.

The "back-of-the-envelope" approach works fine for some things. Not so much for others.

Artificial intelligence fundamentally reshapes the cost of human forgetfulness by dramatically reducing the penalty associated with imperfect recall. Intelligent software agents act as extensions of human memory and reasoning, using knowledge graphs as persistent, structured external memory. This shifts the cognitive landscape: remembering becomes less about internal storage and more about orchestrating reliable, machine-supported recall.

And so, how do we leverage artificial intelligence capabilities in our enterprise? How do we enable humans and machines to work together effectively?[50]

1.14. Approaches to Artificial Intelligence

There are two approaches which can be used to think about and take advantage of the possibilities related to artificial intelligence:

1. Layer artificial intelligence over existing legacy information systems
2. Understand what will enable artificial intelligence to operate with its full potential and work back from there.

Here we take the second approach. An AI-First or Digital-First or Graph-First[51] mindset begins with a genuine paradigm shift: you recognize that the landscape has fundamentally changed, and the old map no longer describes the territory. Instead of trying to retrofit yesterday's operating logic, you adopt an entirely new logic; one built around how cyberspace actually behaves, how digital systems truly operate, and how artificial intelligence creates, transforms, and accelerates value.

And so, we have laid the ground work for you to understand artificial intelligence and knowledge graphs.

Let us now dig into knowledge graphs.

2. Knowledge Graphs

A knowledge graph is a graph of knowledge (e.g. semantic network) which is machine interpretable and can be projected into a form that is also interpretable to a human trying to understand the knowledge. Ideally, that human interpretable projection could also be edited by a human with those edits going back into the machine interpretable knowledge graph. To be clear; not two copies, rather one copy usable by both machine and human users.

Structured documents like electronic spreadsheets, HTML web pages, and word processor documents are knowledge graphs. But those knowledge graphs are structured for presentation of a document. They

[50] Digital Financial Reporting, *Human Task Performance*, https://digitalfinancialreporting.blogspot.com/2026/06/human-task-performance.html

[51] Digital Financial Reporting, *AI-First, Digital-First, Graph-First*, https://digitalfinancialreporting.blogspot.com/2026/08/ai-first-digital-first.html

are interpretable by machines so that the machine can present the information to a human in the form of a document. But those documents tend to be very hard for a machine to interpret in terms of the meaning the document is carrying because of the arbitrary way humans tend to create documents using vastly differing formatting of the information and terminology used.

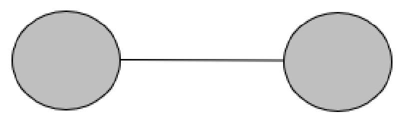
Knowledge graphs are graphs of meaning.

2.1. Graphs

When I use the term graph, I am referring to the term in the context of graph theory[52] which is a discipline of mathematics. Wikipedia's definition of graph theory and graph is:

> 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 **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	Element
Vertex	Relationship	Line	Path	Connection

Reconciling this to something you might know from grammar[53],

- **Noun**: naming words; person, place, or thing. (represented as an edge)
- **Verb**: doing words; action or doing (represented as a vertex)
- **Adjective**: describes a noun (property of an edge)

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

[53] Teaching Resource, https://www.teachstarter.com/au/teaching-resource/nouns-verbs-adjectives-adverbs-posters/

- **Adverb**: describes a verb (property of a vertex)

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.

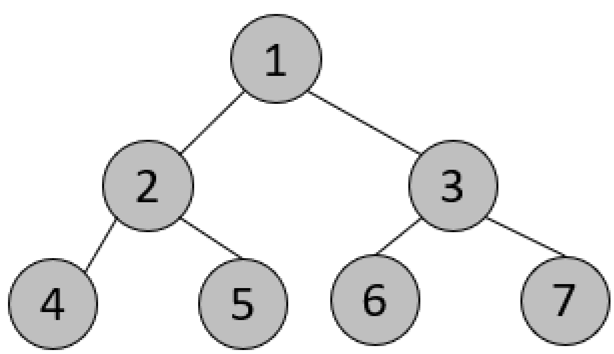
To better understand graphs, let's look at some subtle but very important differences between some different types of graphs.

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.

2.2. 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.

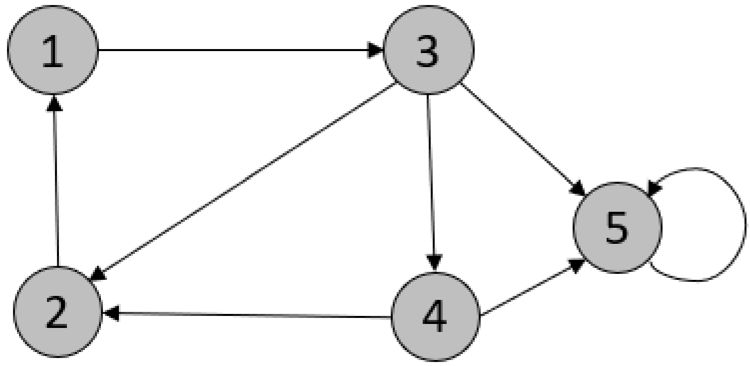
The following is an example of a graph that is also a tree:



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.

2.3. 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:

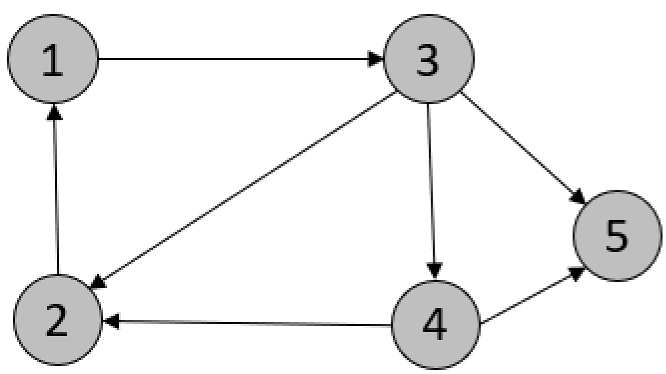


Note that each vertex (line) has an arrow that points in a specific direction; that is what makes the graph a directed graph. Note edge (node) number 5 which has an arrow that points to itself; that is a cycle. Cycles like that can cause issues such as causing an infinite loop. Those sorts of issues 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.

2.4. 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:



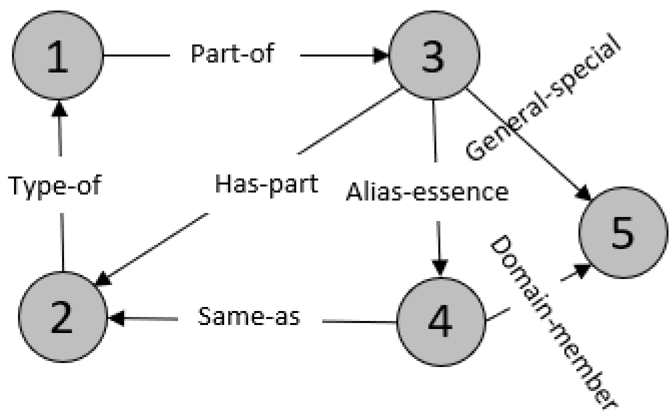
Note again that each vertex has an arrow which specifies a direction and that there are no cycles making this a directed acyclic graph. Note that there is not one specific edge (node) that can be considered the root of the graph.

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)?

2.5. Labeled 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:



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.

2.6. 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". This is explained in, *What is the Truth and Who Owns the Truth*[54].

Also, there seems to be a critically important difference between RDF graphs and graph databases. Seems that RDF graphs are typed, but

[54] Digital Financial Reporting, *What is the Truth and Who Owns the Truth?*, https://digitalfinancialreporting.blogspot.com/2026/08/what-is-truth-and-who-owns-truth.html

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 seems to have been fixed in RDF Star[55].

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*[56] and understanding TypeDB[57] are important. Strongly typed graph 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.

2.7. Named Graphs

There is something that is incredibly important and useful that is worth explicitly pointing out. The idea relates to what is called a named graph[58].

A named graph is essentially a subset of some other graph or graphs that has been given a unique label (referred to as a name) so that that specific subset can be referred to and used again. A named graph is sort of like a report.

Relational closure is the foundational mathematical property stating that the output of every relational operation is itself a relation. Named graphs follow the notion of relational closure. This means named graphs can be chained together.

55 Seattle Method, *Semantic Web and AI: Can we finally realize the original vision?*, https://seattlemethod.blogspot.com/2025/04/semantic-web-and-ai-can-we-finally.html

56 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

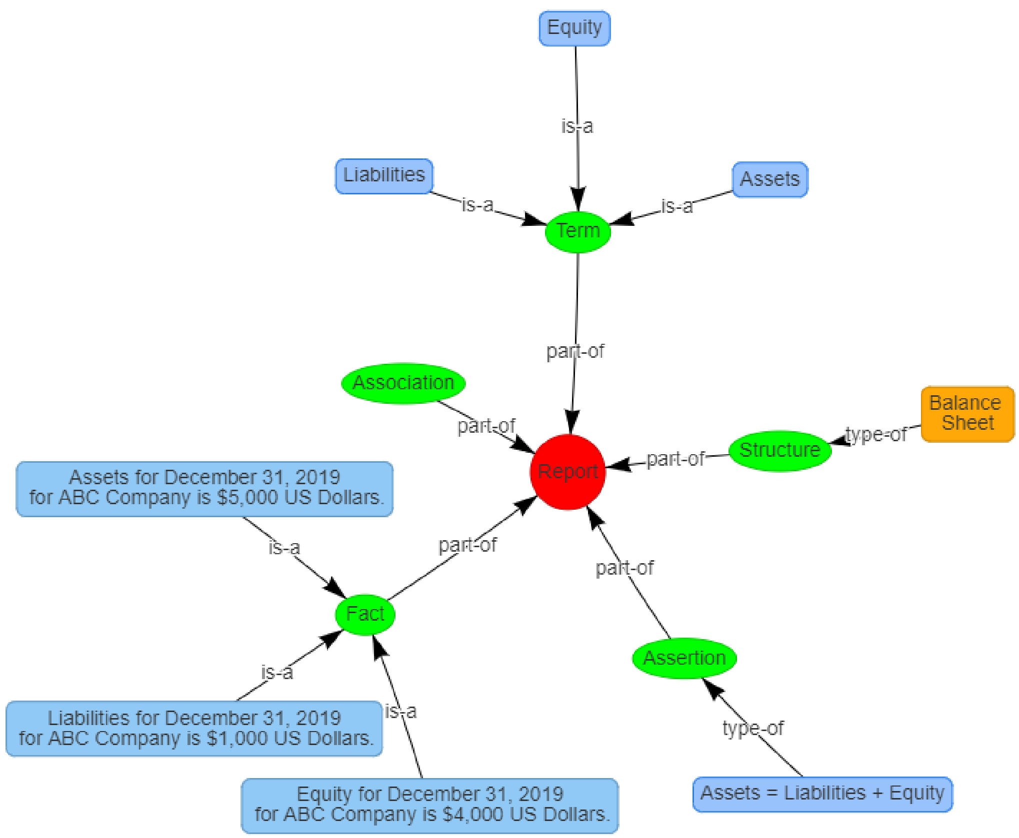
57 Vaticle, Strongly Typed Database, https://vaticle.com/

58 Digital Financial Reporting, *Named Graphs*, https://digitalfinancialreporting.blogspot.com/2024/08/named-graphs.html

2.8. Visualizing Graph of Knowledge

The precise visualizations provided by software tools that implement a graph of knowledge can be different. Colors can be used to enhance visualizations. Different shapes can be used for showing edges (nodes). Visualizations might be laid out in a variety of different ways. There is not necessarily one standard visualization.

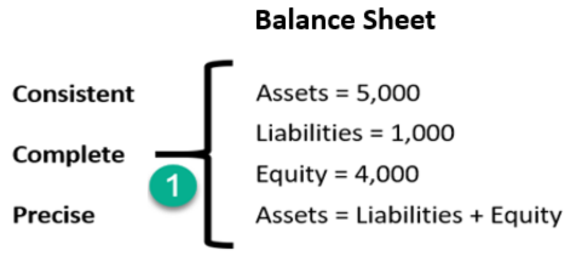
Here is information related to the accounting equation and three facts reported within that report model represented in the form of a labeled directed acyclic graph:



Here is that same information presented in a manner that might be more familiar to professional accountants:

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



2.9. *Visualization of Special-purpose Knowledge Graphs*

Knowledge graphs are general-purpose tools that can be modified and turned into special-purpose tools by adding a specific logical model that both constraints and controls the functionality of the general-purpose model.

Converting from a general-purpose tool to a special-purpose tool has two consequences. First, special-purpose tools are less functional and less flexibly than general-purpose tools. Secondly, special-purpose tools are an order of magnitude easier to use that a general-purpose tool.

If you give up flexibility that you don't need then you lose nothing but you gain ease of use. That is the benefit of creating special-purpose tools.

This is what a special-purpose financial statement knowledge graph might look like:

Human readable rendering:

Reporting Entity [Axis]	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442
Unit [Axis]	USD
Comprehensive Income Statement [Line Items]	Period [Axis] 2020-01-01/2020-12-31
Comprehensive Income [Roll Up]	
Revenues	7,000
(Expenses)	(3,000)
Gains	1,000
(Losses)	(2,000)
Net Income	3,000

Model:

Label	Report Element Class	Period	Balance	Preferred Label Role	Name
Comprehensive Income Statement [Hypercube]	[Table]			Standard Label	proof:ComprehensiveIncomeStatementHypercube
Comprehensive Income Statement [Line Items]	[LineItems]			Standard Label	proof:ComprehensiveIncomeStatementLineItems
Comprehensive Income [Roll Up]	[Abstract]			Standard Label	proof:ComprehensiveIncomeRollUp
Revenues	[Concept] Monetary	For Period	Credit	Standard Label	proof:Revenues
(Expenses)	[Concept] Monetary	For Period	Debit	Negated Label	proof:Expenses
Gains	[Concept] Monetary	For Period	Credit	Standard Label	proof:Gains
(Losses)	[Concept] Monetary	For Period	Debit	Negated Label	proof:Losses
Net Income	[Concept] Monetary	For Period	Credit	Standard Label	proof:NetIncome

Facts:

#	Reporting Entity	Period	Concept	Fact Value	Unit	Rounding	Parenthetical Explanations
1	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-01-01/2020-12-31	Revenues	7000	USD	0	
2	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-01-01/2020-12-31	(Expenses)	3000	USD	0	
3	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-01-01/2020-12-31	Gains	1000	USD	0	
4	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-01-01/2020-12-31	(Losses)	2000	USD	0	
5	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-01-01/2020-12-31	Net Income	3000	USD	0	

Conditions: (a.k.a. assertions)

Label	Rendered Value	Op	Reported Value	Calculated Value	Balance	Result	Name
Comprehensive Income Statement [Line Items]							proof:ComprehensiveIncomeStatementLineItems
Comprehensive Income [Roll Up]							proof:ComprehensiveIncomeRollUp
Revenues	7,000	+	7,000		Credit		proof:Revenues
(Expenses)	(3,000)	-	3,000		Debit		proof:Expenses
Gains	1,000	+	1,000		Credit		proof:Gains
(Losses)	(2,000)	-	2,000		Debit		proof:Losses
Net Income	3,000		3,000	3,000	Credit	Verified	proof:NetIncome

Can you see the knowledge graph in the different representations of information? Think dynamic pivot table.

Next, let us apply the general ideas of a knowledge graph to financial statements.

2.10. Control

Financial statements tell a story. That story is about the financial position and financial performance of a reporting economic entity. That story must be "true" and "fair". That story is a "signal" your organization sends.

The information conveyed by that story (contained in that signal) should be the same whether a traditional human readable report is used as the medium or whether a machine readable knowledge graph is used as the medium.

Financial statements are not "standard forms". Report models can be modified or "customized" by reporting entities which can use different "subtotals", different disclosure "alternatives", and even report additional disclosures which the economic entity feels is important to understanding that specific economic entity. That flexibility is a feature of financial reporting frameworks such as US GAAP and IFRS.

But while financial statements are not "standard forms", they are also not "random". There are patterns. There are "good practices" and "best practices".

When a report model can be modified/customized, the "wild behavior" of accountants creating reports and report models must be controlled and preferably even eliminated, keeping report models within permitted boundaries. While permitted boundaries can be defined differently by, say, different CPA firms or even different accountants within the same CPA firm; patterns exist and those patterns can be leveraged.

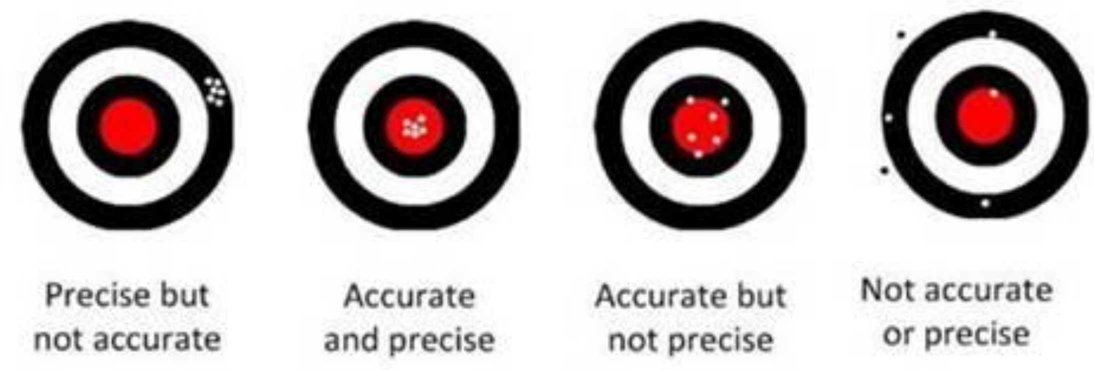
A financial reporting framework represented using an XBRL taxonomy which is then used to represent a report model for a report created by an economic entity in machine readable form serves multiple purposes:

- **Description**: It is a clear and should be complete description of a report model (specification of what is permitted); created by standards setters or regulators or anyone else specifying a report. And obviously the clear and complete description should represent accounting and reporting rules precisely and accurately.
- **Construction**: It is a guide to the creation of a report based on that permitted report model description whereby a human can be assisted by software applications utilizing that machine readable description of permitted report models.
- **Verification**: The actual report constructed can be verified against the clear, complete description assisted by software applications utilizing that machine readable description.
- **Extraction**: Information can be effectively extracted from machine readable reports and report models assisted by software utilizing that machine readable clear and complete description.

Note that the machine readable version of the report model description and report can be automatically converted from the machine readable format to a human readable format using automated processes.

To reiterate; a machine readable representation of a financial reporting framework in an XBRL taxonomy must be ***clear***, ***complete***, and reflect accounting and reporting rules ***precisely*** and ***accurately***[59].

[59] *What is Accuracy?*, https://www.adamequipment.com/aeblog/what-is-accuracy



Traditionally, financial reporting frameworks have been represented in books and can often be unclear. The US GAAP and IFRS XBRL taxonomies are, as they are represented today, are not clear as the really could be or need to be, they are not complete, they are missing rules.

Further, US GAAP and IFRS reporting rules are sometimes ambiguous, but that lack of clarity can be "worked around" by a skilled and experienced accountant that understands accounting and reporting principles and the existing rules well and if they have the right skill and experience they can create "sensible" alternatives considering the ambiguous (i.e. not clear) articulation of the rules by standards setters and regulators.

Whether accounting and reporting rules are in human readable form or machine readable form; these knowledge graphs should be clear, complete, accurate, and precise. The advantage of machine readable form is that the process can be more formalized and tasks and processes can be automated.

2.11. Graphs vs Relational Databases

In his paper, *At Its Core: How Is a Graph Database Different from a Relational One?*[60], Philipp Brunenberg summarizes the fundamental difference between graph databases and relational databases: "Graph and relational databases differ in one fundamental design principle: Graphs do have a concept of a relationship and relational don't. That's why a graph database can manage interconnected data much more efficiently. Still, both have their reasons for existence: Graphs perform better and are more intuitive to use when analyzing an entire context close to a single data point — potentially with multiple hops. However,

[60] Philipp Brunenberg, *At Its Core: How Is a Graph Database Different from a Relational One*?, https://towardsdatascience.com/at-its-core-hows-a-graph-database-different-from-a-relational-8297ca99cb8f

if the exploration of highly joined and densely connected data is not a requirement, a relational model may serve the needs similarly well."

In the hands of someone with no skills or experience working with knowledge graphs; they tend to produce clumsy, ugly, barely functional output. But in the hands of a skilled craftsmen, knowledge graphs can produce works of utility, elegance, beauty, and durability.

2.12. Graphs vs Documents

We need to start thinking beyond the document[61]. A document is a "costume" a financial statement wears[62]. Beneath the surface of what looks like a document lies a rigorously designed logical substrate[63]. That is why rules-based artificial intelligence excels at validating consistency, coherence, and provability; because the content is fundamentally logical and mathematical like a graph, not linguistic like a document.

What if the graph and the document were separated? What if a single version/copy of the truth existed? What if that single version of the truth could be provided in both a reliably interpretable machine readable format and then projected into a reliably interpretable human readable format?

Remember the obstacle related to making use of a computer's capabilities related to business professional idiosyncrasies and inconsistent understanding of an area of knowledge by business professionals? Documents tend to be arbitrarily and therefore inconsistently created by humans because of these idiosyncrasies. The underlying logic conveyed by the documents tends to be very consistent. Why not use a "graph-first" and "digital-first" philosophy/strategy[64] and give artificial intelligence the best chance of success?

Enter the digital information organism.

---

[61] Digital Financial Reporting, Thinking Beyond the Document, https://digitalfinancialreporting.blogspot.com/2026/01/thinking-beyond-document.html

[62] Digital Financial Reporting, *Document is a Custom a Financial Statement Wears*, https://digitalfinancialreporting.blogspot.com/2026/02/document-is-costume-financial-statement.html

[63] Digital Financial Reporting, *Example Financial Statement Holon*, https://digitalfinancialreporting.blogspot.com/2026/05/example-financial-statement-holon.html

[64] Digital Financial Reporting, *AI-First, Digital-First, Graph-First*, https://digitalfinancialreporting.blogspot.com/2026/08/ai-first-digital-first.html

2.13. Digital Information Organism

A digital information organism[65] pulls all the ideas that have been presented thus far into one mechanism.

Think information Legos. Digital. Scalable. Industrial strength. Maintainable. Modular. Versatile. Global open industry standards based. Artificial intelligence enabled. These digital information Legos (a.k.a. digital information organisms) can be used to replace key mission critical electronic spreadsheets with a scalable industrial strength alternative. These digital information blocks are systems which can replace those "craft" based electronic spreadsheets.

To **industrialize** something means to turn it into a routinized, repeatable, scalable, reliable process that can be done the same way every time. That includes:

- **Standardization**: You define the steps clearly so the work is done the same way each time.
- **Repeatability**: Anyone (or any machine) can follow the steps and get the same result.
- **Scale**: You can do it not just once, but hundreds or thousands of times.
- **Efficiency**: You remove unnecessary variation, waste, or improvisation.
- **Transfer from “craft” to “system”:** Before industrialization work depends on individual skill, judgment, or memory. After industrialization work depends on a documented, controlled process.

A simple metaphor. If you cook a meal from memory, that’s *craft*. If you write a recipe that anyone can follow, that’s *standardization*. If you build a kitchen that can produce 500 identical meals a day, that’s *industrialization*.

Knowledge graphs scale. Electronic spreadsheets are hard, or even impossible, to scale effectively.

2.14. Essence of Accounting

The final idea is to take the digital information organism one step further into the realm of accounting, reporting, auditing, and analysis.

---

[65] Seattle Method, *Digital Information Organism*, https://seattlemethod.blogspot.com/2026/05/digital-information-organism.html

Accounting is a special type of system. As explained in the *Essence of Accounting*[66]; accounting is a deterministic system. Accounting is a well understood de facto standard mathematical model[67].

Accounting, the universal technology of accountability[68], is one of the most mature, enduring and robust information processing system/frameworks ever devised in human history. With over more than 7,000 years of incremental, iterative refinement; accounting has evolved into *one of the most stable, resilient, and interoperable set of standard information systems conventions and features ever developed*. The system includes accounting of business events between rational agents in the form of financial transactions, the reporting of that financial information, assurance of that information in the form of an audit when deemed necessary, and analysis of that financial information.

A *ledger* is not just a list. A ledger[69] is a formal computational substrate within which work is performed. Using another idea from mathematics, or more specifically the branch of mathematics called *topology*[70]; which has the notion of a *manifold*[71] which is another type of mathematical structure.

In topology, a manifold is defined as a topological space that locally resembles Euclidean space. That means if you zoom in on any small region of a manifold, it looks like ordinary flat space, even though the entire shape may be curved or complex globally. Think of topology as the rules of connectivity and manifolds as the spaces that obey those rules while still looking locally flat. Now, all these ideas can be applied to the deterministic system called accounting[72].

[66] Seattle Method, *Essence of Accounting*, https://seattlemethod.blogspot.com/2026/02/essence-of-accounting.html

[67] Seattle Method, *The Accounting Manifold*, https://seattlemethod.blogspot.com/2026/07/the-accounting-manifold.html

[68] Digital Financial Reporting, *Universal Technology of Accountability*, https://digitalfinancialreporting.blogspot.com/2023/02/universal-technology-for-accountability.html

[69] Seattle Method, *Ledger*, https://seattlemethod.blogspot.com/2026/06/ledger.html

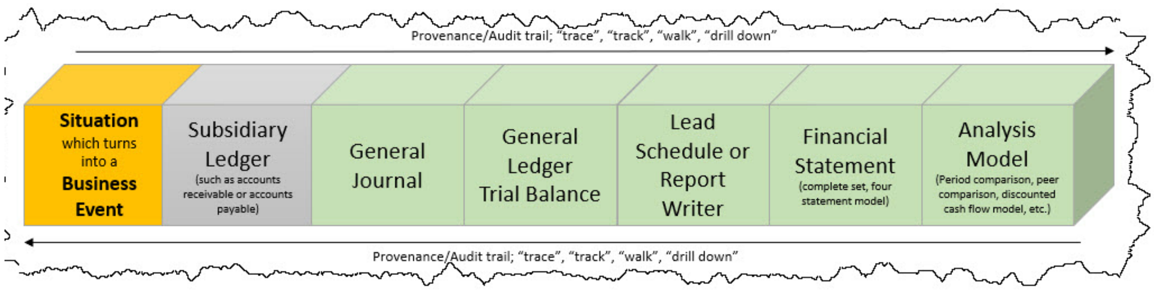
[70] Wikipedia, Topology, https://en.wikipedia.org/wiki/Topology

[71] Wikipedia, *Manifold*, https://en.wikipedia.org/wiki/Manifold

[72] Seattle Method, *The Accounting Manifold*, https://seattlemethod.blogspot.com/2026/07/the-accounting-manifold.html

The *Accounting & Audit by Design (A&AD) Framework*[73] pull all these ideas together into a leverageable framework that can be used to implement these ideas within an enterprise.

Business events[74] and financial transactions[75] follow these mathematical rules. The idea[76] is to leverage the characteristics of well constructed knowledge graphs and these capabilities of mathematics to refactor accounting, reporting, auditing, and analysis to create better tools for accountants than just the electronic spreadsheet[77].



What is even better is that these same ideas can be applied to general business reporting and information exchange in many cases.

2.15. General Business Reporting

While accounting, reporting, auditing, and analysis are somewhat of a special system; these same ideas can be applied to general business reporting.

In fact, an accounting information system provides somewhat of a "skeleton" of an enterprise. Much of the nonfinancial related stuff of an enterprise is connected to the financial stuff of an enterprise that flows through an enterprises accounting information systems. What this offers is a reliable template or framework for explaining the enterprise knowledge graph beyond financial accounting.

---

73 Seattle Method, *Accounting & Audit (A&AD) Framework*, https://seattlemethod.blogspot.com/2026/07/accounting-audit-by-design-framework.html

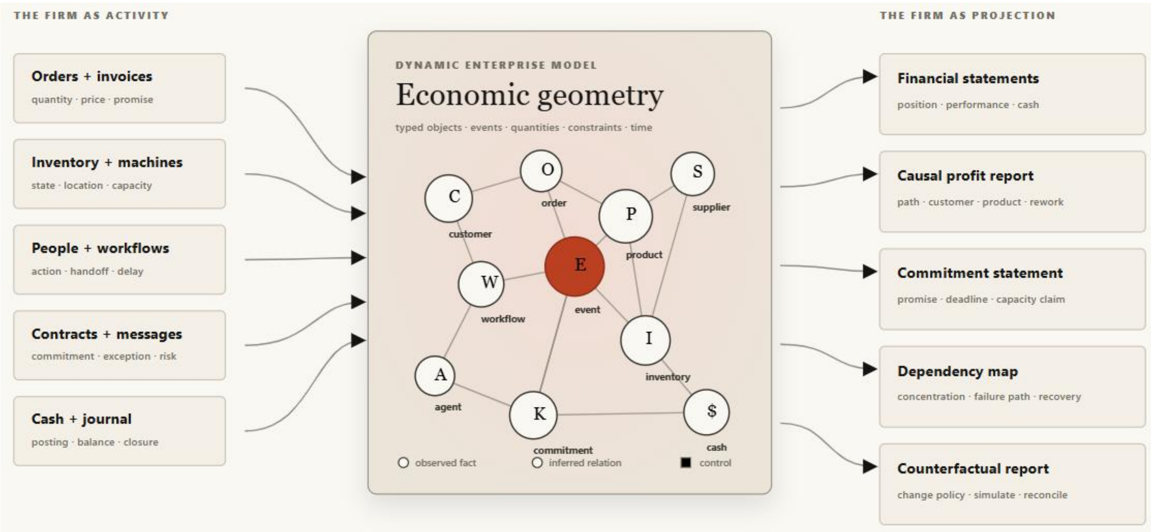
74 Digital Financial Reporting, *Business Events Ledger*, https://digitalfinancialreporting.blogspot.com/2026/06/business-events-ledger.html

75 Digital Financial Reporting, *Consequences of Starting at the End of the Chain*, https://digitalfinancialreporting.blogspot.com/2025/10/consequences-of-starting-at-end-of-chain.html

76 Digital Financial Reporting, *The Vision*, https://digitalfinancialreporting.blogspot.com/2025/09/the-vision.html

77 Digital Financial Reporting, Electronic Spreadsheets, https://digitalfinancialreporting.blogspot.com/2026/02/electronic-spreadsheets.html

As the paper *The Financial Statement Is a Projection*[78] points out; a financial statement is a projection. Connected to the accounting information systems of an enterprise are other systems which may, or may not, but financial in nature. But they all tend to be connected to some system that is financial and flows through the accounting information systems of an enterprise.



And so, in this section we laid out the general idea. In the next section we will explain knowledge graphs as applied to financial statements.

3. Knowledge Graph Theory and Conceptual

In the previous section we summarized the general ideas related to knowledge graphs using the well understood financial statement which is a type of knowledge graph understood by most business professionals.

We pointed out that a graph is a very flexible and expressive communications tool.

In this section we apply those general ideas to the specific use case of financial statements but focus on the theoretical and conceptual ideas which should be understood. Practical examples are provided in the next section.

[78] Puget Research, *The Financial Statement Is a Projection*, https://pugetresearch.com/posts/financial-statement-is-a-projection/

Financial statements are knowledge graphs of the logic conveyed by the information within the financial statement. A financial statement represents "state" or "stocks" (e.g. balance sheet) and "changes in state" or "flows" (e.g. incomes statement, statement of changes in equity, statement of cash flows) and different aggregations and disaggregations, policies, information about broad transaction categories, other quantitative information and qualitative information.

A financial statement tells a story. That story is a signal conveyed to regulators, investors, or others. The human readable and machine readable versions of that story should be the same.

3.1. Function of a General-purpose Financial Statement

A general-purpose is a true and fair representation of information about an economic entity. A financial statement is not the actual economic entity; it merely conveys information about an economic entity that is generally of very high-quality. Consider the following use case of a general-purpose financial statement:

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

There is no natural way to represent an economic entity the way it "really is" in the real world; there are just certain purposeful selections of specific aspects of an economic entity, call them abstractions or models, that provide a useful enough simplification that satisfies some specific goal we might have. That is the nature of a general-purpose financial statement, to represent information about an economic entity for a specific purpose. That representation is good enough to be useful.

Some people use the analogy of a financial statement to that of an X-ray or a compression technology[79]. Another analogy that is sometimes uses is that of a "digital twin"[80]. A financial statement is less of a digital twin which would be far more robust; a financial statement is more like the skeleton of an enterprise.

Financial statement knowledge graphs can be interrogated systematically and logically using machine-based processes.

3.2. Essence of a General-purpose Financial Statement

A general-purpose financial statement is a high-fidelity, high-resolution, high-quality information exchange mechanism. The report is a compendium of complex logical information required by statutory requirements and regulatory rules plus whatever management of an economic entity wants to voluntarily disclose. The report represents quantitative and qualitative information about the financial condition and financial performance of an economic entity. There are a number of different financial reporting frameworks that might be used to create a general-purpose financial statement such as US GAAP, IFRS, IPSAS, GAS, FAS, etc.[81].

Financial statements are not uniform[82]. Financial statements are not forms; they have variability. This consciously allowed variability is an essential, characteristic trait of robust reporting schemes such as US GAAP, IFRS, and others. This allowed variability contributes to the richness, high-fidelity, and high-resolution of reported financial information that is unique to an industry sector, a style of reporting, or an economic entity. This variability is a feature of such reporting schemes. Different reporting styles, different subtotals used to aggregate details, and using some specific approach given a set of allowed alternatives are examples of variability. Variability does not mean "arbitrary" or "random". There are known identifiable patterns.

Rules are used to articulate allowed variability and "channel" creators of financial statements in the right direction and therefore control variability, keeping the variability within standard limits. That keeps

---

79 Puget Research, *The Financial Statement is a Projection*, https://pugetresearch.com/posts/financial-statement-is-a-projection/

80 *Digital Twins of an Organization for Enterprise Modeling*, https://www.dfki.uni-kl.de/~maus/dok/RissMausJavaid%2B2020.pdf

81 *Financial Reporting Schemes*, http://xbrlsite.azurewebsites.net/2020/master/ElementsOfFinancialStatements.pdf

82 *Essence of Accounting*, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf

quality where it needs to be. Rules enable things like preventing a user from using a concept meant to represent one thing from unintentionally being used to represent something different.

Further, the discipline of describing something in a form a computer algorithm can understand also assists you in understanding the world better; weeding out flaws in your understanding, myths, and misconceptions about accounting and reporting standards.

3.3. Economic Entity Report Model

Because each financial statement can be different, each financial statement created by each economic entity essentially has its own specific report model. However, every reporting economic entity report model must comply with some specified reporting framework[83]. All reporting frameworks and report models fit into one financial statement metamodel that is described by the *Logical Theory Describing Financial Report*[84].

It is that logical conceptualization[85] of a financial statement that turns a general-purpose knowledge graph into a special-purpose knowledge graph. That special-purpose knowledge graph is specified by and constrained by a logical theory[86] which explains that conceptualization.

The theory specifies the elements, permissible connections between those elements, organization of the connections, conditions that must be satisfied (a.k.a. assertions, restrictions, constraints), process related rules, inference rules, and how actual instantiations of observations and measurements must relate to one another per those specified conditions.

To use this special-purpose knowledge graph, professional accountants need only understand the fundamentals of knowledge graphs, understand the logic of a financial statement, and understand the financial statement metamodel which is used to create all financial statement models for every reporting economic entity. Much of this

[83] Seattle Method, *Reference Reporting Frameworks*, https://seattlemethod.blogspot.com/2026/01/reference-reporting-frameworks.html
[84] Charles Hoffman, CPA, et al, *Logical Theory Describing Financial Report*, http://accounting.auditchain.finance/framework/LogicalTheoryDescribingFinancialReport.pdf
[85] Seattle Method, *Conceptualization*, https://seattlemethod.blogspot.com/2025/09/conceptualization.html
[86] Seattle Method, *Theory*, https://seattlemethod.blogspot.com/2025/12/theory.html

understanding can be buried within software to hide complexity[87] from the software user.

To create a standard financial statement logical conceptualization, we want to build on top of a business report logical conceptualization because a financial statement is a special type of the more general business report. What this achieves is to enable our special-purpose report model to also be used more generally (e.g. for nonfinancial information)

3.4. Logical Theory Explained in Simple Terms

A system[88] can be explained by a theory[89]. A logical theory is an abstract conceptualization[90] 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**[91] enables a community of stakeholders trying to achieve a specific goal or objective or a range of goals/objectives to agree on important statements used for capturing meaning or representing a shared understanding of and knowledge in some area of knowledge.

A logical theory is made up of a set of *models*, *structures*, *elements*, *connections*, *conditions*, and *facts*. In very simple terms,

- **Logical theory**: A *logical theory* is a set of models that are consistent with and permissible per that logical theory.
- **Model**: A *model*[92] is a set of structures that are consistent with and permissible interpretations of that model.
- **Structure**: A *structure* is a set of logical statements which describe the structure.

---

87 Seattle Method, Complexity, https://seattlemethod.blogspot.com/2025/01/complexity.html

88 Seattle Method, *System*, https://seattlemethod.blogspot.com/2025/01/system.html

89 Seattle Method, *Theory*, https://seattlemethod.blogspot.com/2025/12/theory.html

90 Seattle Method, *Conceptualization*, https://seattlemethod.blogspot.com/2025/09/conceptualization.html

91 Seattle Method, *Logical Theory*, https://seattlemethod.blogspot.com/p/logical-theory.html

92 Seattle Method, *Essence of Modeling*, https://seattlemethod.blogspot.com/2026/08/essence-of-modeling.html

- **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 theory relates. There are four broad categories of logical statements:
  - **Elements**: *Elements* (a.k.a. terms, entities, nodes, report elements, things) are logical statements that define ideas or notions used by the logical theory. An element may be primitive and therefore not decomposable or an element can be compound and decomposable into a set of primitive elements. An example of primitive elements might be "assets", "liabilities", and "equity". An example of a compound element might be "balance sheet".
  - **Connections**: *Connections* (a.k.a. relations, associations) are logical statements that describe permissible interrelationships between the elements such as "assets is part-of the balance sheet" or "operating expenses is a type-of expense" or "an asset is a 'debit' and is 'as of' a specific point in time and is always a monetary numeric value".
  - **Conditions**: *Conditions* (a.k.a. assertions, restrictions, constraints, rules) are logical statements that describe what tend to be 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 (a.k.a. observations, measurements) that are provided by an economic entity within a business report. For example, the financial statement, 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.

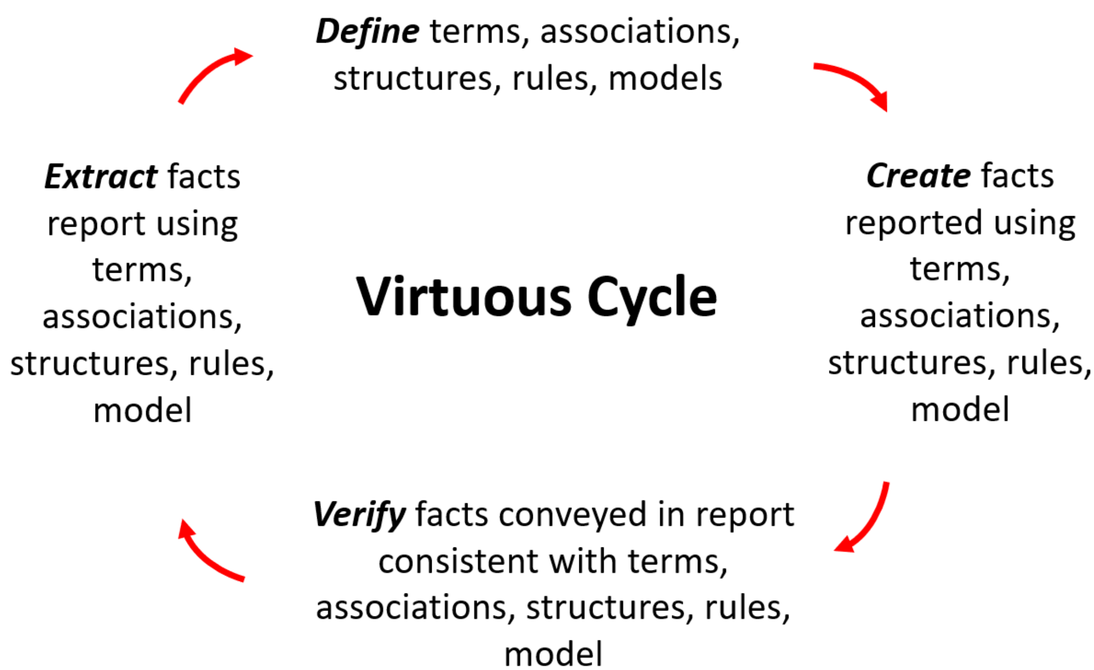
Fundamentally, a logical theory is a set of logical statements. Those logical statements can be represented in human-readable form or they could be expressed in machine-readable form. Once in machine-readable form, those logical statements can be interrogated using software applications. To the extent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

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.

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 system is working right, it creates a virtuous cycle[93].



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

3.5. Logical Theory Describing Business Report

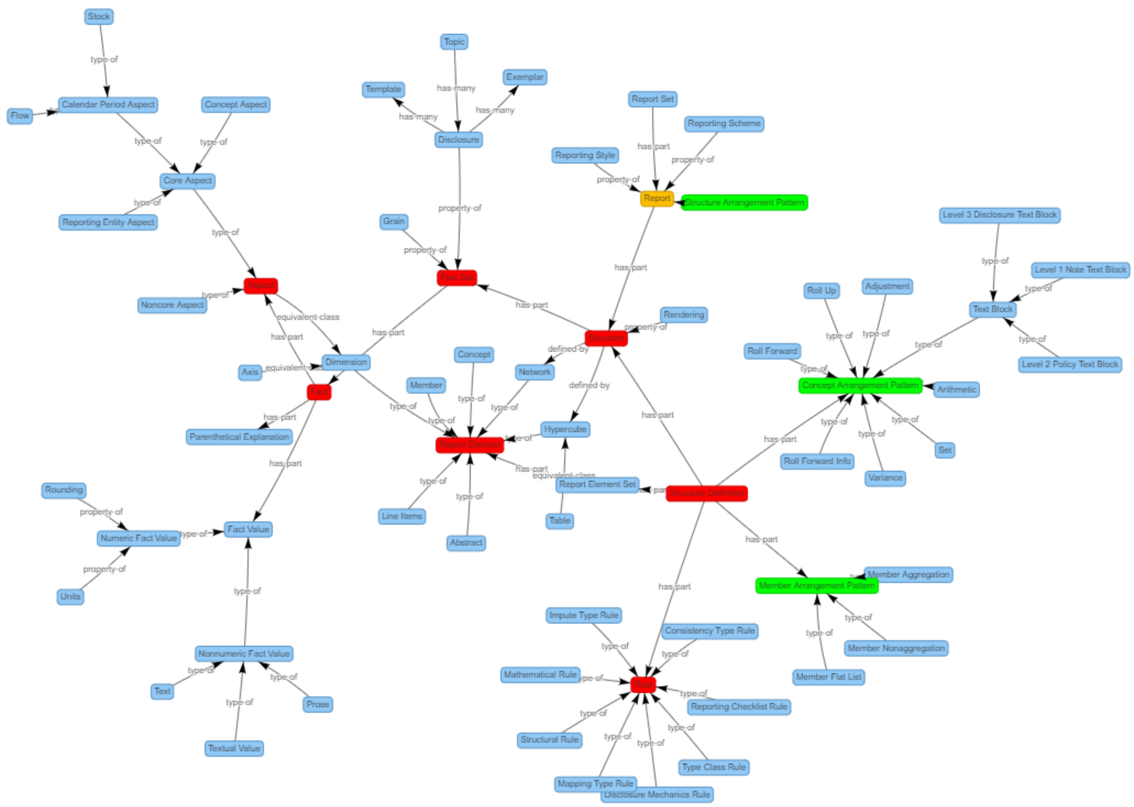
The business report metamodel is simply a logical system that is based on a standard logical conceptualization of a business report which is

---

[93] Seattle Method, *Virtuous Cycle*, https://seattlemethod.blogspot.com/2025/01/virtuous-cycle.html

based on XBRL International's *Open Information Model*[94] (OIM) and/or the Object Management Group's *Standard Business Report Model* (SBRM)[95]. These two models are well aligned and the Seattle Method logical conceptualization of a business report is based on the OIM model and is the basis for the SBRM model[96].

The following is an overview of the business report metamodel explained in simple terms[97].



3.6. Logical Theory Describing Financial Report

The financial statement metamodel is simply a standard business report model that is further constrained to provide additional artifacts exclusive

---

[94] XBRL International, *Open Information Model (OIM)*, https://www.xbrl.org/the-standard/what/introducing-the-oim/

[95] OMG, *Standard Business Report Model* (SBRM), https://www.omg.org/spec/SBRM/1.0/Beta1/PDF

[96] Seattle Method, Reconciling Seattle Method, OIM, SBRM, XBRL, https://seattlemethod.blogspot.com/2025/12/reconciling-seattle-method-oim-sbrm-xbrl.html

[97] Seattle Method, *Business Report Pieces*, https://xbrlsite.azurewebsites.net/seattlemethod/BusinessReportPieces.html

to financial statements and additional constraints and restrictions. Remember, all financial statements fit into one financial statement metamodel that is described by the *Logical Theory Describing Financial Report (Terse)*[98].

Additional artifacts that are added to the SBRM to meet the needs of a financial statement include:

- Topics
- Disclosures
- Business events
- Fundamental accounting concepts
- Reporting styles
- Specific type-subtype associations
- Specific consistency rules
- Specific derivation rules
- Templates
- Exemplars

Financial statement models can be made unique for each financial reporting frameworks including US GAAP and IFRS. A complete inventory of the logical objects that might exist in a financial statement is provided by the PROOF reference implementation.

3.7. Very Simple Example of Financial Statement Model

We will provide a very, very basic example of a financial reporting framework[99] and financial statement report model[100] to introduce you to financial statement models. This very, very basic example is expanded on in section four which provides many more practical high quality examples of XBRL-based digital financial reporting frameworks and financial statements.

---

98 Seattle Method, *Logical Theory Describing Financial Report (Terse)*, https://xbrlsite.azurewebsites.net/seattlemethod/archive/LogicalTheoryDescribingFinancialReport_Terse.pdf

99 Seattle Method, *Accounting Equation Reporting Framework*, https://xbrlsite.azurewebsites.net/seattlemethod/platinum/ae/ae_ModelStructure.html

100 Seattle Method, *Accounting Equation Reference Implementation*, https://xbrlsite.azurewebsites.net/2026/ref-github-ae/index2.html

A very simple example of a financial statement model is the **accounting equation**. Here is a description of the accounting equation financial statement model in both human-readable terms and machine-readable terms using XBRL[101]:

Elements: Three simple elements are defined: Assets, Liabilities, Equity. One complex element is defined, Balance Sheet.

Structure: One structure is defined, the Balance Sheet, and identified using the element Balance Sheet.

Connections: The three elements Assets, Liabilities, and Equity are connected in that they are all PART-OF the structure balance sheet.

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

Facts: Instances of three facts (e.g. observations, measurements) are established to exercise the model: Assets of $5,000; Liabilities of $1,000; Equity of $4,000.

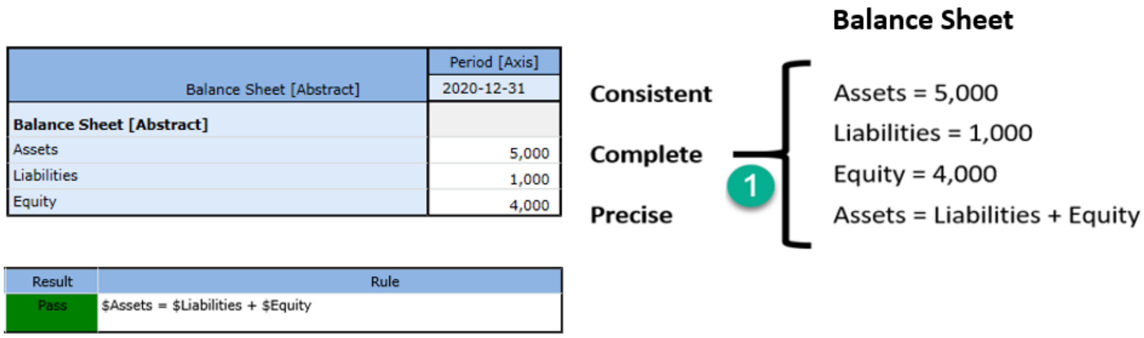
Model: All of the elements, connections, conditions, structures, and facts describe the model. We created only one model, or permissible interpretation, of the financial statement model.

(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. But we are not using that permissible version of the accounting equation within this financial statement model.)

Because this is a very simple example with only a few logical statements it is easy to get your head around this specific financial statement model 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:

---

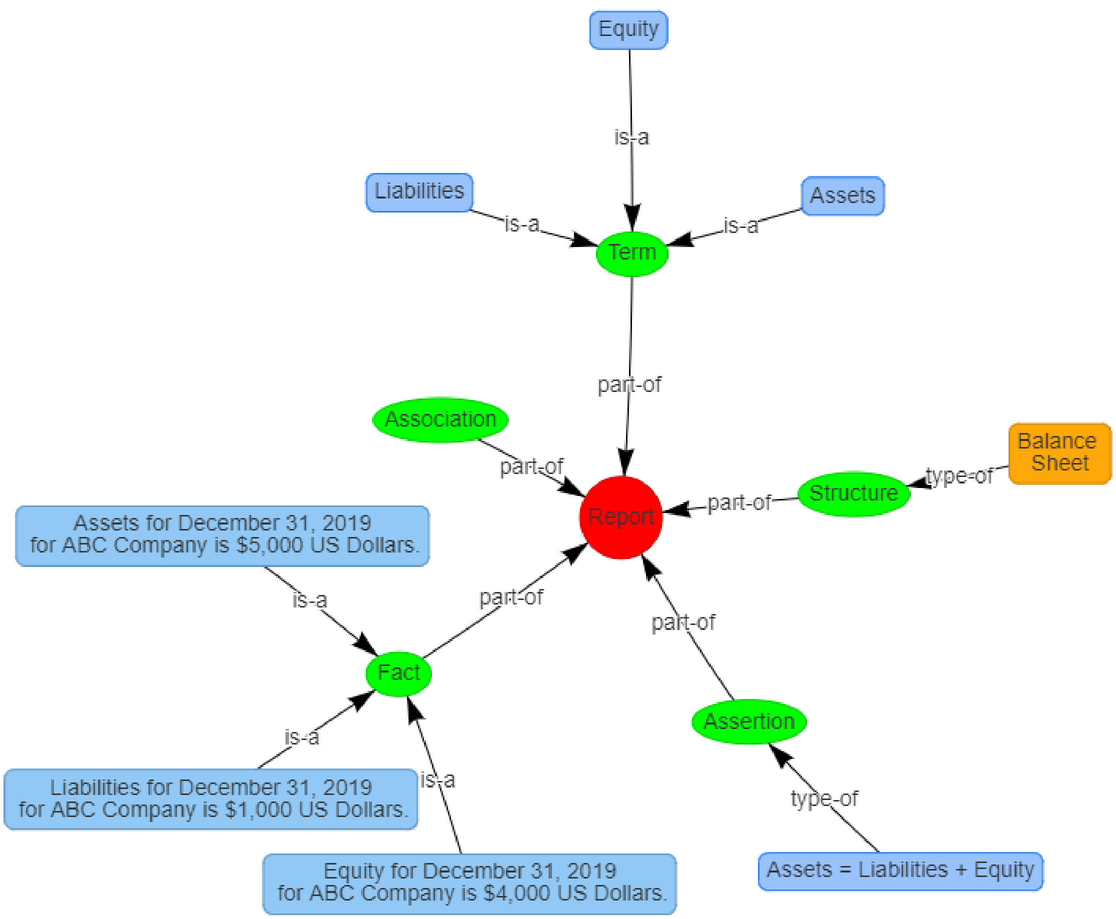
[101] Seattle Method, *Accounting Equation*, https://xbrlsite.azurewebsites.net/seattlemethod/platinum/ae/ae_ModelStructure.html



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

As the size of the financial statement model increases it becomes increasingly more challenging to verify that the logical system is properly functioning using manual processes. We will discuss the types of things that can go wrong with a system in a later section. Essentially, the models, terms, structures, rules, and facts form a labeled directed acyclic graph or knowledge graph such as this simplified knowledge graph which describes the system we are discussing:



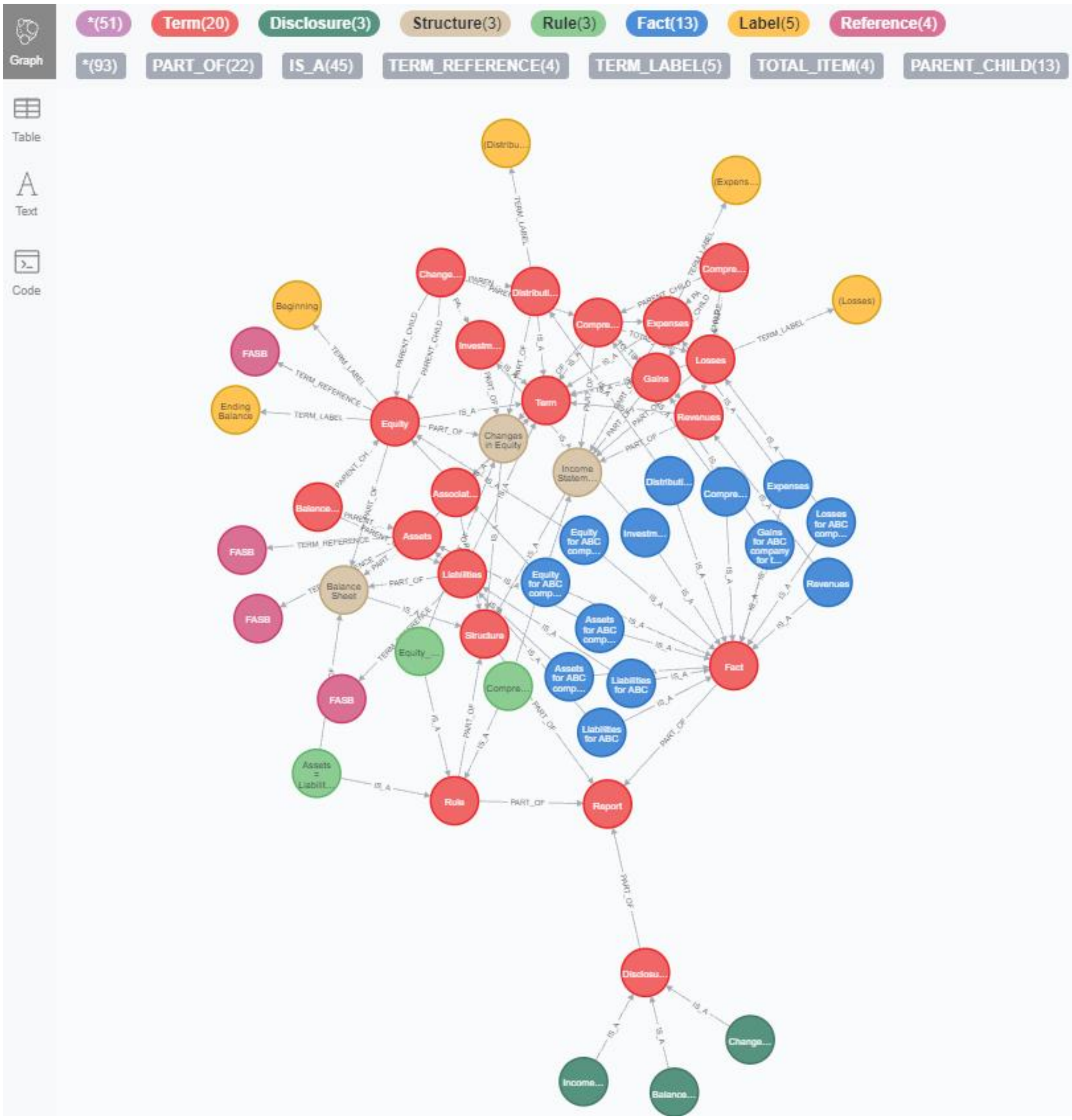
Hopefully you get the general idea from this simplified wireframe representation of our logical system as a knowledge graph.

While a typical financial statement is significantly larger in volume (i.e. the Microsoft 2017 10-K[102] is made up of 194 structures; 2,035 facts; 3,296 connections; etc.) every financial statement knowledge graph represented using XBRL works the same as this very simple example but just has more pieces.

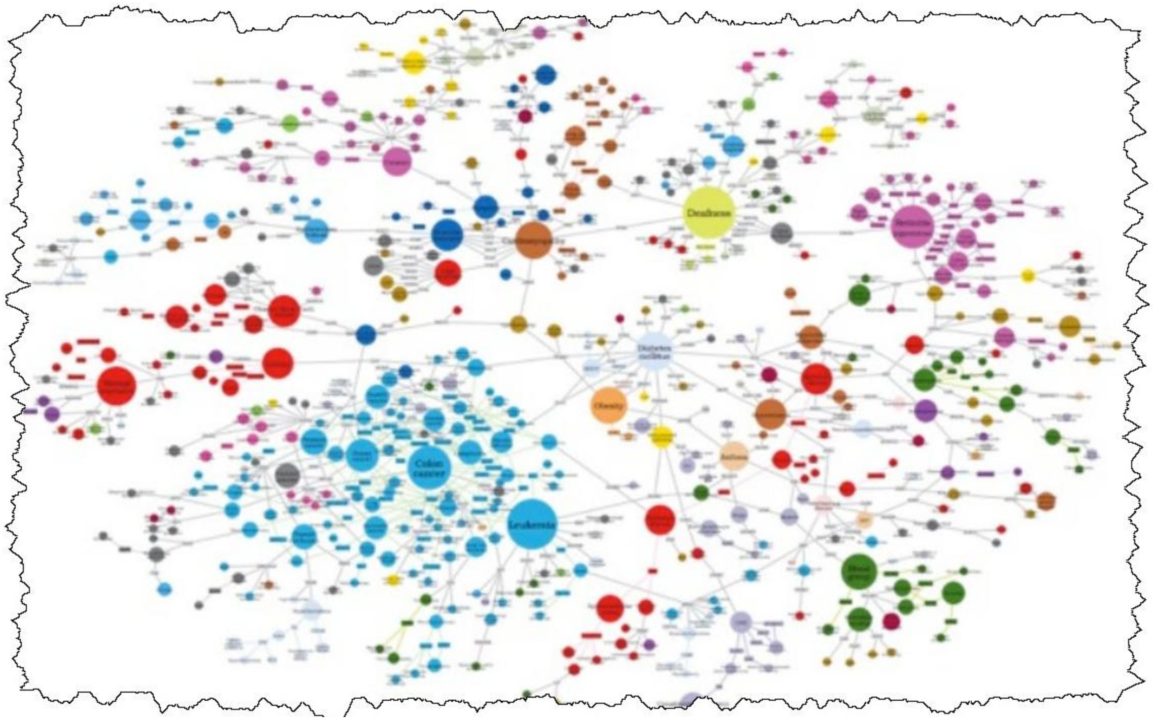
3.8. Visualizing Financial Statement Knowledge Graph

When you work with a financial statement knowledge graph in a general-purpose tool for working with any knowledge graph from any area of knowledge, what you see might look something like the following:

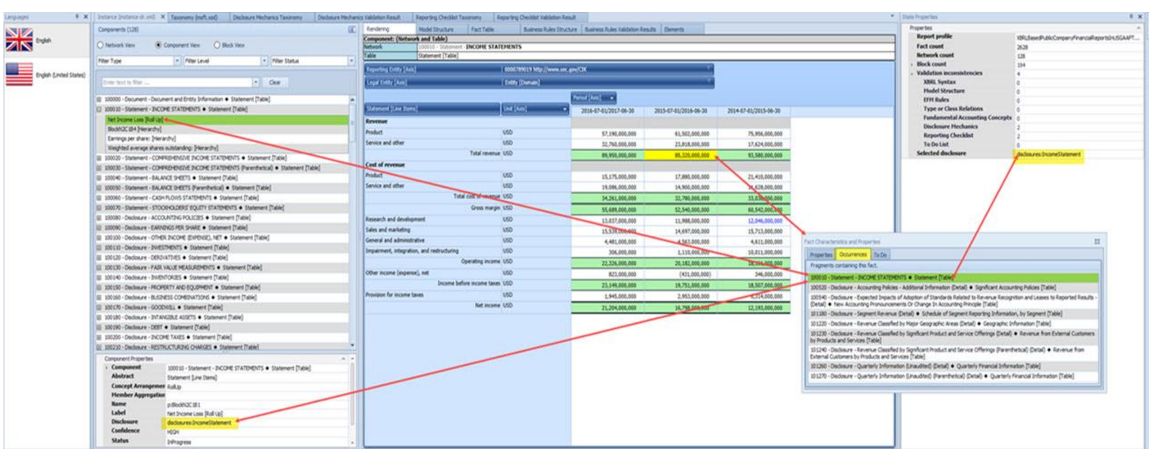
[102] SEC, Microsoft Form 10-K, 2017-06-30, https://www.sec.gov/Archives/edgar/data/789019/000156459017014900/0001564590-17-014900-index.htm



Above you see the knowledge graph of a smaller prototype financial statement as seen within Neo4j which is a graph database. The more information you have in the financial statement knowledge graph, the more "nodes" and "edges" you will be able to see. A more comprehensive financial statement might look as follows using different colors and different sizes to differentiate node types:



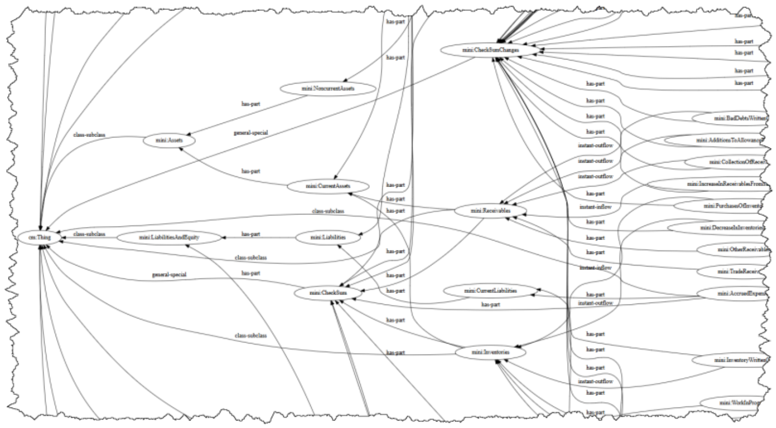
If you look at the same knowledge graph of information about a financial statement in a special-purpose tool for working with such financial statement knowledge graphs, it might look something like this[103]:



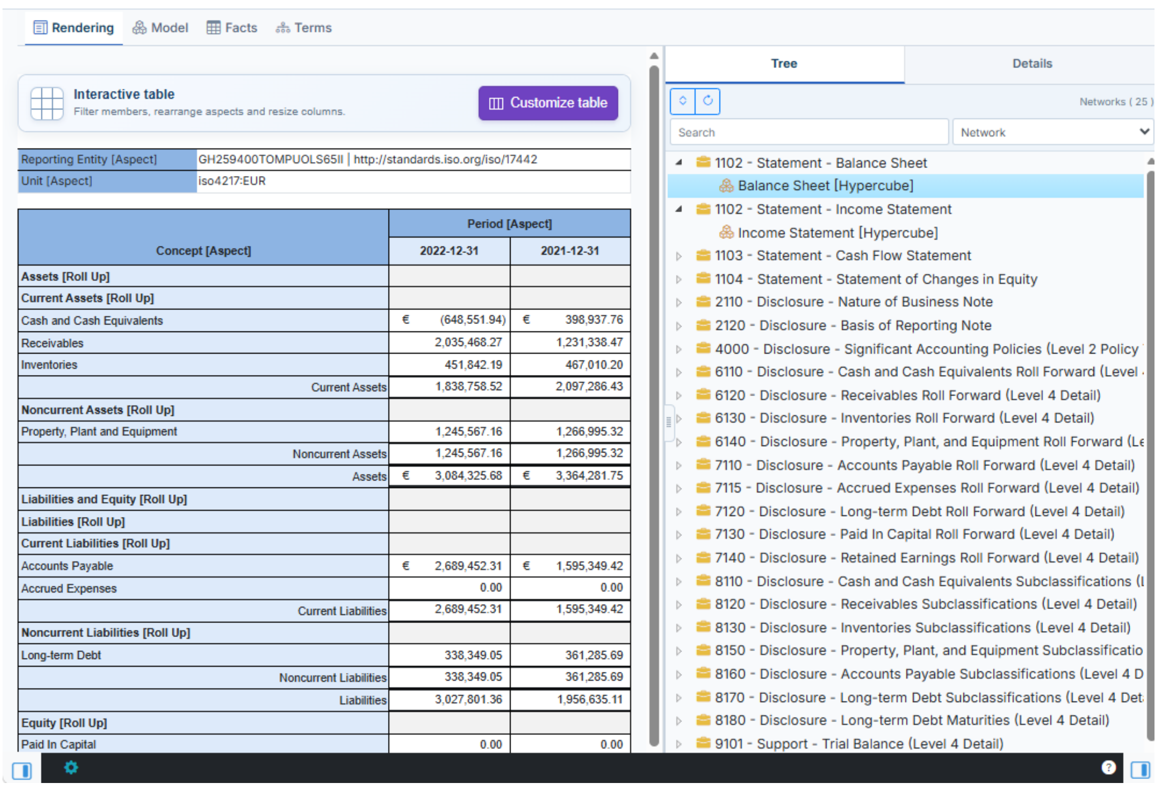
A specialized tool understands all those edges and vertices within the knowledge graph and the labeled directed acyclic graphs that are represented and can use this information to dynamically work with the

103 *Pesseract*, http://xbrlsite.azurewebsites.net/2021/library/KnowledgeGraph_Pesseract.jpg

financial statement logical model. Here is an example of how Auditchain Pacioli sees a financial statement knowledge graph[104]:



But that same knowledge graph can be seen as a dynamic software application, here is Auditchain's Luca Suite basic viewer[105].



---

[104] Seattle Method, MINI, Type-subtype Relations, https://xbrlsite.azurewebsites.net/2026/ref-github-mini/pacioli/typeSubTypeGraph.html

[105] Auditchain, *Luca Suite Basic Viewer, MINI reference financial statement*, https://luca.pacioli.ai/luca/view/9175b4014583e9fb8113bc253bf24aa764caa86880eec558111ca1cb0f8b1f67dc0beb6b/index

The point here is that both general tools and specialized tools specifically for working with financial statements in specific ways can be used to read knowledge graphs. Applications might be static or dynamic.

3.9. XBRL-based Digital Financial Statements

XBRL is a global standard technical syntax that is used in over 60 countries for representing financial statements. Teaching you to use XBRL is not in the scope of this resource because good software applications will completely abstract the XBRL technical syntax away from professional accountants.

3.10. Intentional Computational Substrate

A financial statement is a knowledge graph. A machine interpretable knowledge graph can be used to represent a financial statement. A financial statement is effectively a computational substrate that follows the double entry bookkeeping model[106].

The mathematics of double-entry bookkeeping is built on a single invariant: every economic event must preserve the balance of the accounting equation: Assets = Liabilities + Equity. (a.k.a. Resources = Obligations)

Resources and obligations don't simply appear out of thin air. Resources and obligations are not just account categories. Resources and obligations are topological constraints that carve the shape of the accounting manifold[107]. Resources define the positive directions on the manifold (economic capacity). Obligations define the counter-directions (claims against that capacity). The invariant relationship between them, the accounting equation, is the constraint surface that the ledger can never leave. This is the "conservation law that governs movement on the manifold".

To enforce this, each transaction is recorded as a pair of equal and opposite entries; debits and credits; which ensures that the system behaves like a closed, conserved structure. In mathematical terms, a ledger becomes a matrix whose columns (transactions) always sum to zero, creating a self-balancing network of flows. This simple constraint gives rise to a powerful formal system capable of representing complex financial activity with internal consistency, traceability/trackability,

---

[106] Seattle Method, *Essence of Accounting*, https://seattlemethod.blogspot.com/2026/02/essence-of-accounting.html

[107] Seattle Method, *The Accounting Manifold*, https://seattlemethod.blogspot.com/2026/07/the-accounting-manifold.html

provenance, and auditability. This mathematical artifact was intentionally designed and has specific useful characteristics:

- **Zero-sum structure**: Every transaction is a vector whose debit and credit components sum to zero, making the ledger a matrix with zero-sum columns.
- **Conservation law**: The system behaves like a conserved quantity in physics; resources and obligations move between accounts, but the total structure remains balanced because of the zero-sum structure.
- **Classification logic**: Account types (assets, liabilities, equity, revenue, expense) determine how debits and credits change balances, giving the system its algebraic rules.
- **Rearrangeable intermediate components**: Basic account types have superordinate and subordinate hierarchies which serve as totals and subtotals which can be reorganized offering flexibility in creating structures.
- **Articulation**: Intentionally mathematically interconnected structures or reports or primary statements are used to provide information about the "state" or "stocks" (e.g. balance sheet) and about "changes in state" or "flows" (e.g. statement of income, statement of cash flows, statement of changes in equity).
- **Temporal ordering**: Transactions form a time-ordered sequence, allowing the ledger to be treated as a dynamic system evolving under strict constraints.

Accounting can be understood as a topology-constrained system: a space of all possible financial states shaped by invariants such as the accounting equation and the rules of double-entry bookkeeping. Each valid ledger state lies on a constraint surface; a manifold-like structure embedded within a higher-dimensional space of potential account balances. Transactions act as continuous transformations that move the system from one point on this surface to another without ever leaving it.

3.11. Skeleton of Enterprise Digital Twin

This computational substrate that is the journals, ledgers, and financial statements of an enterprise forms what amounts to a “skeleton” or “x-ray” of an enterprise[108]. This skeleton serves as the trusted truth of the

---

[108] Digital Financial Reporting, *Compliance Reporting as a "Skeleton" of the Enterprise Knowledge Graph*,

enterprise. That skeleton can be expanded with additional information which supplements that skeleton.

Further, this information skeleton can be "packaged" as what is referred to as a holon[109]. A holon is something that is both a whole and a part at the same time.

3.12. Levels of Report

To clearly and precisely understand XBRL-based digital financial reporting[110], it helps to think about the levels of reports[111] as a spectrum of financial statements in terms of levels similar to how levels are helpful in understanding the capabilities of self-driving cars[112].

The term "self-driving" means different things to different people so it makes it difficult to have a precise conversation about that topic. But breaking the description into a precise spectrum of descriptions is very helpful to the communication process.

This is similarly true for the levels of an XBRL-based digital financial statement represented as a knowledge graph. Below we will break down a financial statement into helpful levels that will enable a precise and clear discussion. We will provide a very brief description, a little bit of information, and a link to specific examples that instantiate a report per each specific level.

The marginal difference between each level is very helpful in providing the reader with a solid understanding of the different levels.

Here is an overview of the levels related to financial reporting as I see them beginning with the least functional in terms of both human and machine use of the information from with a financial statement.

- **Level 0**: Not machine readable. *An example of Level 0 is a clay tablet, papyrus, or paper as the report medium*.

---

https://digitalfinancialreporting.blogspot.com/2026/08/compliance-reporting-as-skeleton-of.html

[109] Digital Financial Reporting, *Example of Financial Statement Holon*, https://digitalfinancialreporting.blogspot.com/2026/05/example-financial-statement-holon.html

[110] Digital Financial Reporting, *XBRL is an Extra Fancy Knowledge Graph*, https://digitalfinancialreporting.blogspot.com/2024/05/xbrl-is-extra-fancy-knowledge-graph.html

[111] Digital Financial Reporting, *Levels of Reports*, https://digitalfinancialreporting.blogspot.com/2024/01/levels-of-report.html

[112] Truecar, The 5 Levels of Autonomous Vehicles, https://www.truecar.com/blog/5-levels-autonomous-vehicles/

- **Level 1**[113]: Machine readable, nonstandard, structured for presentation. *PDF, HTML, or XHTML are examples of Level 1*.
- **Level 2**[114]: Machine readable, nonstandard, structured for meaning, but no explanation of the reported information; no taxonomy (a.k.a. dictionary), no rules, no report model. *An XBRL-based report instance without an XBRL taxonomy schema, without XBRL relations and resources, and without XBRL Formulas is an example of Level 2*.
- **Level 3**[115]: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary), incomplete rules, incomplete high-level report model. *An XBRL-based report with a XBRL taxonomy schema, with XBRL relations and resources, but without XBRL Formulas is an example of Level 3.*
- **Level 4**[116]: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary), complete set of rules provided, incomplete high-level report model. *An XBRL-based report with a XBRL taxonomy schema, with XBRL relations and resources, and with XBRL Formulas that completely describes the report is an example of Level 4*.
- **Level 5**[117]: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary), complete set of rules provided, complete global standard high-level report model, yields PROVEN properly functioning system and UNDERSTANDABLE report information. *An XBRL-based report with all the characteristics of Level 4, plus consistency cross checks, type-subtype relations, consistent 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 is an example of Level 5*.

[113] Level 1 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level1/
[114] Level 2 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level2/
[115] Level 3 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level3/
[116] Level 4 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level4/
[117] Level 5 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level5/

- **Level 6**: All of Level 5 PLUS blockchain-anchored XBRL to increase trust. *An XBRL-based report with all the characteristics of Level 5, plus information within a digital distributed ledger that assures no one has tampered with the report is an example of Level 6*.
- **Level 7**: All of Level 6 PLUS blockchain-anchored accounting transactions and events with provenance and tracing/tracking information. *An XBRL-based report with all the characteristics of Level 6, plus information that indicates that assures no one has tampered with transactions is an example of Level 7*.
- **Level 8**: All of Level 7 PLUS blockchain-anchored business event information linked to supporting documentation for each business event. *An XBRL-based report with all the characteristics of Level 7, plus access to a business events ledger*[118] *that is connected to documentation*[119] *which supports the business event*.

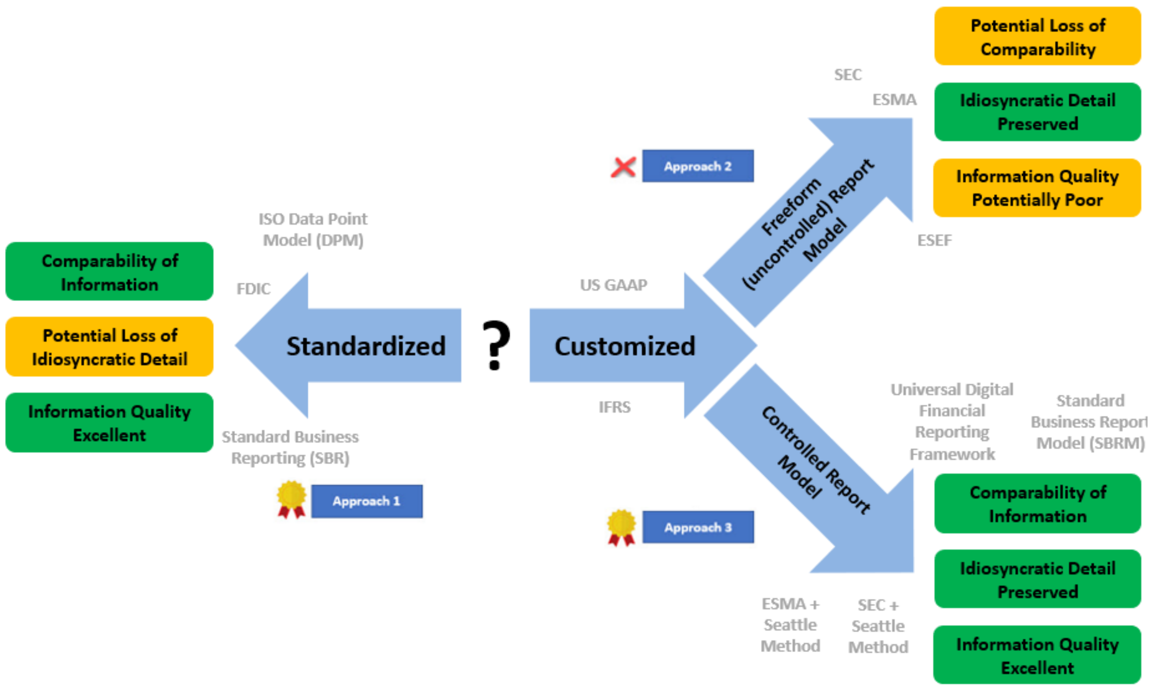
There is something very important to note here. The set of logical statements that is used to specify/describe how a report should be created (say by a regulator or standards setter), used to actually create a report (say an accountant), verify that the report was created consistently to the specification/description (say an accountant or software application used by an accountant), independently confirm that the report was created consistently with the specification/description (say by an independent auditor), or extract information from the created report (say by a financial analyst or regulator) are all the same set of logical statements.

3.13. Reporting Approaches

The paper *Critical Reflection on XBRL: A "Customisable Standard" for Financial Reporting?*[120], breaks reporting into two approaches: standardized reporting and customized reporting. I modified this breakdown slightly breaking customized reporting into two distinct approaches, "freeform customization" and "controlled customization". I

[118] Digital Financial Reporting, *Business Events Ledger*, https://digitalfinancialreporting.blogspot.com/2026/06/business-events-ledger.html
[119] Digital Financial Reporting, *Theory of Documentation*, https://digitalfinancialreporting.blogspot.com/2026/08/theory-of-documentation.html
[120] Reporting Approaches + XBRL Approaches + Implementation Approaches, http://xbrl.squarespace.com/journal/2021/12/30/reporting-approaches-xbrl-approaches-implementation-approach.html

then reflected the three approaches in the following graphic inspired by the graphic in the referenced paper[121]:



This yields three distinct modeling approaches:

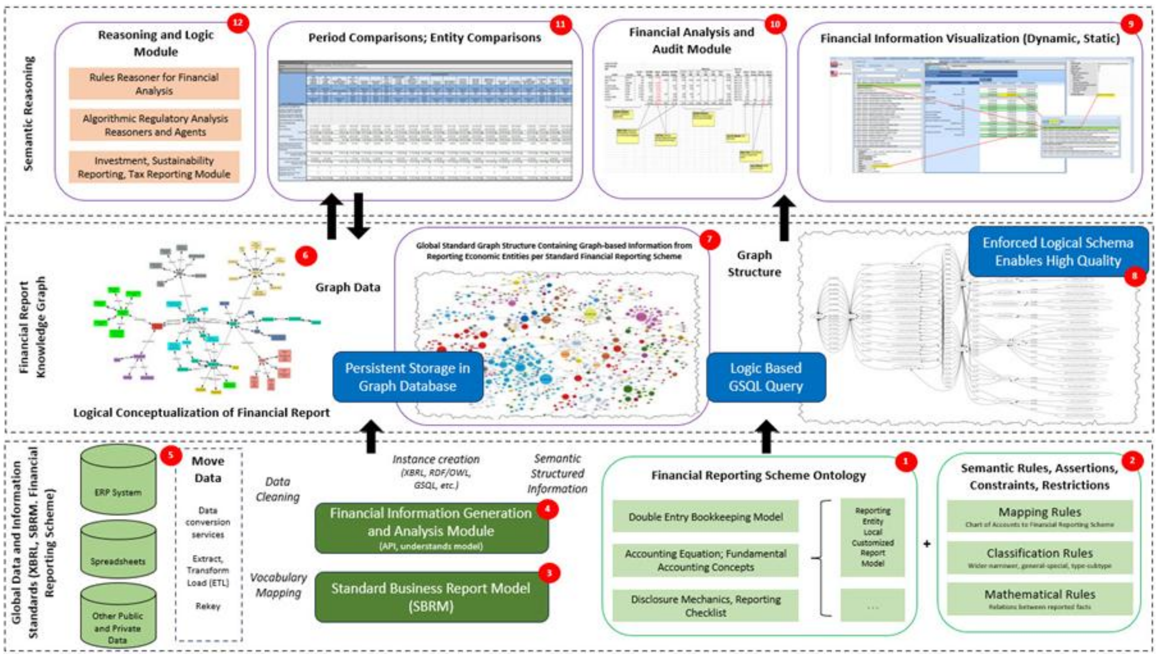
- **Standard form model**: No modifications are allowed to the report model.
- **Freeform, Uncontrolled model**: Modifications are permitted to report model, but those modifications are not controlled in any way. As such there is no differentiation between permitted and unpermitted modifications to the model.
- **Controlled model**: Modifications are permitted to report model and a mechanism is provided to control report model modifications; permitted and unpermitted report model modifications are clearly delineated and control mechanisms keep report model modification within permitted boundaries.

Effectively, uncontrolled customization of report models simply will not work.

121 Taxonomy creation approaches, http://xbrlsite.azurewebsites.net/2022/library/TaxonomyApproachesSeattleMethod.jpg

3.14. Workflow Pipeline

As best as I can understand it and describe it, what will exist in the future is a knowledge graph system[122] for financial reporting, more like a pipeline[123] than one stand alone system. This is a graphic of the components of the big picture as I see it, a knowledge graph system for financial reporting[124]:



Such a knowledge graph system is made up of knowledge assemblies.

3.15. Knowledge Organization System

A knowledge assembly[125] 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 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.

---

122 *Knowledge Graph System*, https://digitalfinancialreporting.blogspot.com/2023/07/knowledge-graph-system.html

123 Digital Financial Reporting, *Building Out the Ontology Pipeline*, https://digitalfinancialreporting.blogspot.com/2025/07/building-out-ontology-pipeline.html

124 *Knowledge Graph System for Financial Reporting*, https://digitalfinancialreporting.blogspot.com/2023/07/knowledge-graph-system-for-financial.html

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

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, not the technical jargon/terminology of computer science.

Another term to describe this is a knowledge organization system[126]. A knowledge organization system (KOS) is a broad but neutral umbrella term that seems to be evolving and spans traditional library tools and modern information science and semantic web structures. Knowledge organization systems include a variety of approaches that are fundamentally used to organize, manage, and support retrieval and to make use of information. A knowledge organization system is a tool for organizing that shared, trusted accumulated information which forms knowledge for a group of stakeholders.

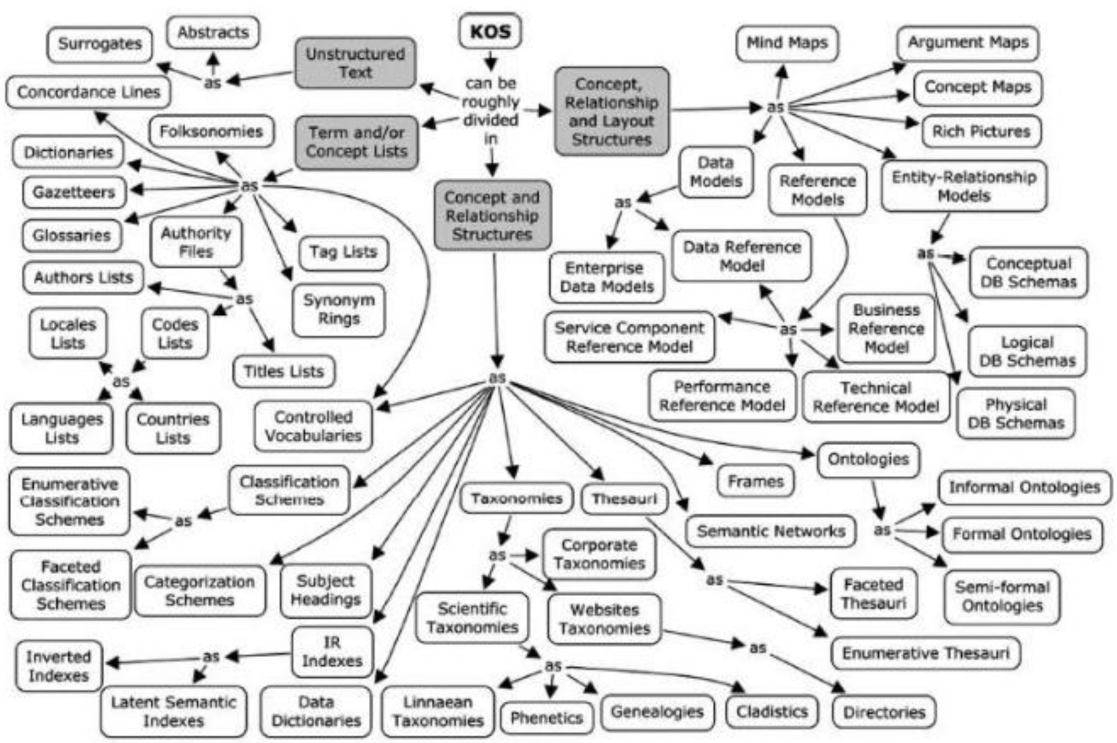


Fig. 1: Souza et al.'s (2012) classification of KOSs

126 Seattle Method, *Knowledge Organization System*, https://seattlemethod.blogspot.com/2026/04/knowledge-organization-system.html

3.16. Global Standard Knowledge Assembly

My PROOF[127] (and my other examples[128]) provides examples of a global standard XBRL-based knowledge assembly[129] of a financial reporting scheme, a financial statement model created by a reporting economic entity using that financial reporting framework, and a financial report using that financial report model that is based on the financial reporting framework. That entire knowledge assembly is validated using a rules engine that is specialized for this specific type of knowledge assembly to verify that the report and report model are complete, consistent, and precise. This mechanism is described by the Seattle Method.

3.17. Platforms and Ecosystems

Platforms and ecosystems[130] provide leverage. A platform is an enabling foundation, framework, and rules others build on which enables ***value creation***. An ecosystem is the interdependent environment formed when many platforms and actors co-create value; enables ***value flow***.

A *platform* is an enabling foundation that others can build on. A platform is a shared foundation of rules and resources that allows independent actors to build, connect, and create value on top of that platform.

An *ecosystem* is a self-sustaining network of interdependent platforms, actors, and value flows. An ecosystem is a dynamic environment formed by multiple platforms, participants, and value flows that mutually depend on each other and evolve together. An ecosystem is a system formed by organisms in interaction with their environment.

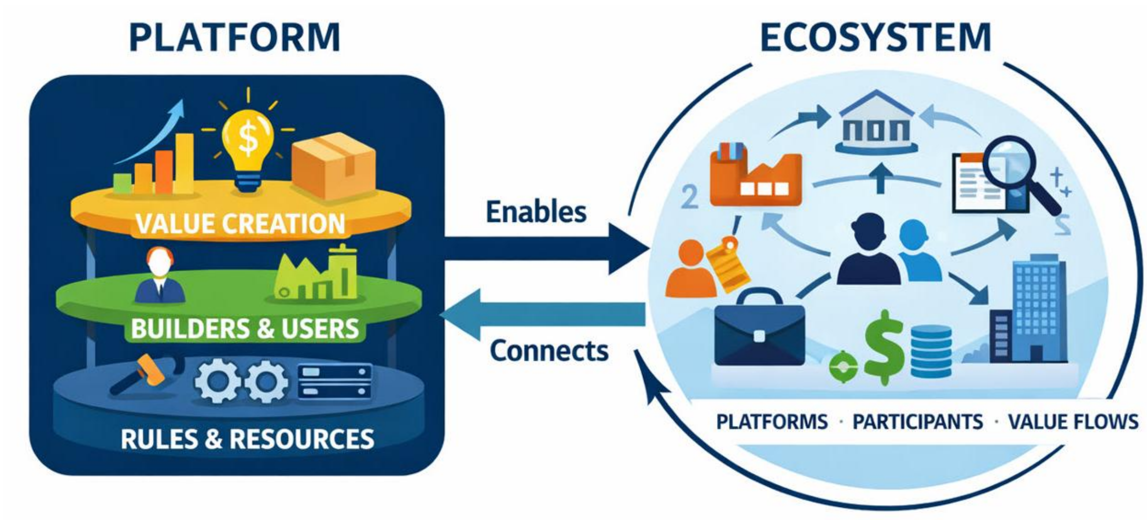
---

127 *PROOF Financial Reporting Scheme* (Prototype), http://www.xbrlsite.com/seattlemethod/platinum/proof/base-taxonomy/proof_ModelStructure.html

128 *PLATINUM XBRL-based Financial Report Examples*, https://digitalfinancialreporting.blogspot.com/2023/07/platinum-xbrl-based-financial-report.html

129 *Global Standard Knowledge Assembly*, https://digitalfinancialreporting.blogspot.com/2023/08/global-standard-knowledge-assembly.html

130 Digital Financial Reporting, *Platforms and Ecosystems*, https://digitalfinancialreporting.blogspot.com/2026/07/platforms-and-ecosystems.html



3.18. Best Practices

A **best practice** (a.k.a. good practices) based method that I am calling the *Seattle Method*[131] was created in order to effectively create XBRL-based financial statements that are provably properly functioning logical systems.

Using this method, enterprises can reliably and effectively stream a high-quality machine-readable XBRL-based global standard knowledge graph of a complete, consistent, and provably correct general purpose financial statement. Further, an entire record-to-report process can be automated effectively. This method provides both the flexibility and the control necessary to effectively hit this target within an enterprise.

The target of this method is Level 5 and above. Below Level 5 the functionality what we generally need from such reports in terms of quality and effective use of reported information in automated machine-based processes is not good enough. It is possible to create a Level 4 XBRL-based report that is properly functioning. Level 5 provides a guarantee that the Level 4 financial statement is properly functioning within a provides specification articulated with a complete set of rules. Level 5 measures quality whereas Level 4 quality is essentially based on what amounts to luck or hope which are not effective engineering techniques.

[131] Charles Hoffman, CPA, Seattle Method, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf

3.19. Modern Work Systems and Modern Accountancy

As explained in *Computational Professional Services*[132], unprecedented human-machine collaboration will be made possible using knowledge graphs and artificial intelligence.

Humans are very capable of carrying out financial accounting, reporting, auditing, and analysis steps and performing tasks. Leveraging knowledge graphs, mechanical devices can also be created to carry out such steps and perform tasks. For example, a vending machine is such a device. A calculator is also such a device. Computers via the software programs they run can likewise carry out steps and perform tasks. Machines will augment humans enabling humans to do what they do best and machines to assist in the steps and tasks that they do best. Entire record-to-report processes can be controlled and therefore automated.

New types of work systems[133] can, and will, be created. Human-machine teaming will be the norm[134].

3.20. General Business Processes and Work

The take away from all this is the following: Implementing these ideas of artificial intelligence, digital, graph first is easy for accounting, reporting, audit, and analysis because of the robust models and standards that exist in that space. This proves the ideas can work. Accounting, reporting, audit, and analysis processes will be industrialized[135] over the coming years.

These same ideas can, and will be, applied to other nonfinancial related processes and work.

---

132 Seattle Method, Computational Professional Services, http://xbrlsite.azurewebsites.net/seattlemethod/archive/ComputationalProfessionalServices.pdf

133 Digital Financial Reporting, *Work System*, https://digitalfinancialreporting.blogspot.com/2026/07/work-system.html

134 Seattle Method, *Neuro-Symbolic Artificial Intelligence with Human Teaming*, https://seattlemethod.blogspot.com/2026/06/neuro-symbolic-artificial-intelligence.html

135 Seattle Method, *Industrial Process*, https://seattlemethod.blogspot.com/2026/03/industrial-process.html

4. Knowledge Graph Examples

This section provides specific practical, tangible, high quality examples of knowledge graphs provided by financial reporting frameworks and financial statements created per those reporting frameworks. The examples start small but then get incrementally more and more complicated to help the reader understand what contributes to complexity.

Each example is provided by a set of reference reporting frameworks[136]. All of the examples below are provided on GitHub[137].

4.1. Accounting Equation

While not an actual reporting framework, the accounting equation[138] example provides a very basic financial statement knowledge graph which has most of the characteristics of any reporting framework or report created per that reporting framework.

Reporting Entity [Aspect]	GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442
Unit [Aspect]	iso4217:EUR

Concept [Aspect]	Period [Aspect]
	2022-12-31
Balance Sheet [Arithmetic]	
Assets	€ 5,000
Liabilities	1,000
Equity	4,000

The accounting equation example includes three elements: Assets, Liabilities, and Equity. The is one fixed structure in this case and there is one condition which must be met by a financial statement: Assets = Liabilities + Equity.

[136] Seattle Method, *Reference Reporting Frameworks*, https://seattlemethod.blogspot.com/2026/01/reference-reporting-frameworks.html
[137] GitHub, *Seattle Method*, https://github.com/seattlemethod
[138] Seattle Method, *Accounting Equation*, https://xbrlsite.azurewebsites.net/2026/ref-github-ae/index2.html

4.2. SFAC6

Likewise, SFAC6 is not really a reporting framework but it is used as an example for its simplicity and small size. The primary difference between the accounting equation example and the SFAC6 example[139] is that the accounting equation has on structure and the SFAC6 example has three structures.

Similar to the accounting equation example, the SFAC6 example provides a balance sheet structure. In the SFAC6 example there are two balance sheet periods, whereas in the accounting equation example there was only on balance sheet period.

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]	
	2025-12-31	2024-12-31
Balance Sheet [Arithmetic]		
Assets	$ 2	$ 0
Liabilities	1	0
Equity	1	0

The second structure of SFAC6 is the statement of comprehensive income.

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]
	2025-01-01 | 2025-12-31
Comprehensive Income [Roll Up]	
Revenues	$ 2
(Expenses)	(1)
Gains	1
(Losses)	(1)
Comprehensive Income	$ 1

And finally, the third structure of SFAC6 is the statement of changes in equity.

139 Seattle Method, *SFAC6*, https://xbrlsite.azurewebsites.net/2026/ref-github-sfac6/index2.html

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]	
	2025-01-01 | 2025-12-31	
Changes in Equity [Roll Forward]		
Equity, Beginning Balance	$	0
Comprehensive Income		1
Investments by Owners		1
(Distributions to Owners)		(1)
Equity, Ending Balance	$	1

And so while the volume of the example increases and the number of structures and types of structures increase, the general concepts are the same in both examples.

4.3. SFAC8

The SFAC8 example[140] introduces the notion of reporting styles. All the other ideas are the same, only increasing in volume, so here we will focus on explaining the difference introduced in this example.

The idea introduced is the notion of the reporting style. To highlight this notion of a reporting style, we will show you two different approaches or styles to representing a statement of comprehensive income. This first style is similar to SFAC6:

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]	
	2025-01-01 | 2025-12-31	
Comprehensive Income [Roll Up]		
Revenues	$	2
(Expenses)		(1)
Gains		1
(Losses)		(1)
Comprehensive Income	$	1

140 Seattle Method, *SFAC8*, https://xbrlsite.azurewebsites.net/2026/ref-github-sfac8/index2.html

But this second reporting style example uses a different approach to organizing the information represented in the statement of comprehensive income:

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]
	2025-01-01 | 2025-12-31
Comprehensive Income [Roll Up]	
Income from Normal Activities of Entity	$ 1
Income from Peripheral or Incidental Transactions of Entity	0
Comprehensive Income	$ 1

There are two ideas that you should come away with from this specific example. First, the understanding that in many cases financial reporting allows a reporting economic entity to use different line items, subtotals, and totals when preparing a financial statement. For example, a software company like Microsoft and a financial institution like Wells Fargo organize their financial statements different. Second, XBRL-based knowledge graphs were specifically designed to be able to handle this use case.

4.4. Common Elements

The Common Elements example[141] introduces the notion of articulation and occurrences. The first thing you will notice is that the examples are looking more and more like a real financial statement. Not quite there yet, but moving in that direction.

Another idea introduced is something that most accountants clearly understand which is the notion of articulation[142]. Articulation is the notion that a set of primary financial statements is intentionally interlinked mathematically.

Connected to this idea of articulation is the notion that a fact can occur in more than one location in a report. For example, the fact "Equity"

[141] Seattle Method, *Common Elements*, https://xbrlsite.azurewebsites.net/2026/ref-github-common/index2.html
[142] Digital Financial Reporting, *Understanding Articulation*, https://digitalfinancialreporting.blogspot.com/2023/08/understanding-articulation.html

exists on both the balance sheet and in the statement of changes in equity.

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]	
	2025-12-31	2024-12-31
Assets [Roll Up]		
Current Assets	$ 0	$ 0
Noncurrent Assets	0	0
Assets	$ 0	$ 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	0	0
Equity Attributable to Noncontrolling Interests	0	0
Equity	0	0
Liabilities and Equity	$ 0	$ 0

Here is the statement of changes in equity. Note that the two line items for Equity, the beginning and ending balances in this statement of changes in cash flow are the exact same facts for the line item Equity that you see in the balance sheet. They are not different facts as they might be represented in an electronic spreadsheet. They are literally the same facts.

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]
	2025-01-01 | 2025-12-31
Changes in Equity [Roll Forward]	
Equity, Beginning Balance	$ 0
Comprehensive Income	0
Investments by Owners	0
(Distributions to Owners)	0
Equity, Ending Balance	$ 0

There is one additional thing worth pointing out. Note the column for the Period [Aspect] in the prior screen shot. Compare that with the screen shot below which puts the Period [Aspect] in the rows of the representation.

Concept [Aspect]	Period [Aspect]	Value
Changes in Equity [Roll Forward]		
Equity, Beginning Balance	2024-12-31	$ 0
Comprehensive Income	2025-01-01 | 2025-12-31	0
Investments by Owners	2025-01-01 | 2025-12-31	0
(Distributions to Owners)	2025-01-01 | 2025-12-31	0
Equity, Ending Balance	2025-12-31	$ 0

This second screen shot is more "pure" in terms of formatting the representation, note that the period types are different for the beginning and ending balances (which are instances) as contrast to the changes being represented (which are durations). The first representation is a bit of a short cut to make the rendering more familiar to how accountants create a roll forward representation in a document.

4.5. Office of Comptroller of Currency

The Office of Comptroller of Currency example[143] is basically a very small but what might be considered a compete reporting framework and report.

This example helps the reader consolidate their understanding and it introduces the notion of the "information block" and the "disclosure".

If you look the screen shot below you see two roll ups of information; a roll up of "Resources" and a roll up of "Liabilities".

Each of those roll ups are completely independent "blocks" of "information"; or "information blocks".

But the "Report to Comptroller" is what is being reported, or disclosed, in this example. The one disclosure, "Report to Comptroller" is made up of two separate blocks of information: that roll up of resources which is separate from the roll up of liabilities.

[143] Seattle Method, *Office of Comptroller of Currency*, https://xbrlsite.azurewebsites.net/2026/ref-github-occ/index2.html

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]
	2025-12-31
Resources [Roll Up]	
Loans, Including Overdraft	$ 1
U. S. Bonds and Premiums	1
Real Estate, Furniture and Fixtures	1
Redemption Fund with U. S. Treasurer	1
Cash and Due from Banks	1
Resources	$ 5
Liabilities [Roll Up]	
Capital	2
Surplus and Undivided Profits	1
Circulation	1
Deposits	1
Liabilities	$ 5

4.6. MINI

The MINI example[144] is a rather basic but what can be considered a complete financial statement example and a basic financial reporting framework. It also highlights a unique approach to creating a report.

There are two specific unique aspects to notice. First, roll forward is provided for every real account or balance sheet line item. Second, business event groupings are used which provide the change in real account summaries which end up being line items on the cash flow statement and statement of changes in equity. I don’t want to go into detail here, just be aware that there are some interesting and unique aspects to this financial statement.

Here is a screen shot of the balance sheet:

[144] Seattle Method, *MINI*, https://xbrlsite.azurewebsites.net/2026/ref-github-mini/index2.html

Reporting Entity [Aspect]	GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442
Unit [Aspect]	iso4217:EUR

Concept [Aspect]	Period [Aspect]	
	2022-12-31	2021-12-31
Assets [Roll Up]		
Current Assets [Roll Up]		
Cash and Cash Equivalents	€ (648,551.94)	€ 398,937.76
Receivables	2,035,468.27	1,231,338.47
Inventories	451,842.19	467,010.20
Current Assets	1,838,758.52	2,097,286.43
Noncurrent Assets [Roll Up]		
Property, Plant and Equipment	1,245,567.16	1,266,995.32
Noncurrent Assets	1,245,567.16	1,266,995.32
Assets	€ 3,084,325.68	€ 3,364,281.75
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities [Roll Up]		
Accounts Payable	€ 2,689,452.31	€ 1,595,349.42
Accrued Expenses	0.00	0.00
Current Liabilities	2,689,452.31	1,595,349.42
Noncurrent Liabilities [Roll Up]		
Long-term Debt	338,349.05	361,285.69
Noncurrent Liabilities	338,349.05	361,285.69
Liabilities	3,027,801.36	1,956,635.11
Equity [Roll Up]		
Paid In Capital	0.00	0.00
Retained Earnings	56,524.32	1,407,646.64
Equity	56,524.32	1,407,646.64
Liabilities and Equity	€ 3,084,325.68	€ 3,364,281.75

Only one reporting style is provided for this reporting framework to keep it as simple as possible and as small as possible but also look like a real financial statement.

4.7. PROOF

The PROOF example[145] satisfies another specific purpose. What the PROOF does is provide a small as possible example but with all of the different information representation pattern types. It does this for two reasons. First, it provides an example of each information pattern type. Second, it tests, proves, and demonstrates each of the information pattern types working correctly with every other information pattern type.

Here is one of the approximately nine different information organization pattern types, a "restatement" (a.k.a. adjustment):

Reporting Entity [Aspect]	GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Report Date [Axis]	Period [Aspect]
		2021-12-31
Prior Period Errors [Adjustment]		
Equity, Origionally Stated	Prior Report [Member]	$ 2,000
Changes in Accounting Policy	Current Report [Member]	(1,500)
Correction of an Error	Current Report [Member]	(500)
Equity, Restated	Current Report [Member]	$ 0

That is only one of the 14 different structures provided in this example. Again, this is one of the most significant examples which is very important to understand to get a proper grounding in model based financial statements.

4.8. AASB 1060

The AASB 1060 example[146] is an almost complete financial reporting framework. Again, this satisfies a specific purpose. I figure that this example represents about 80% of a real financial reporting framework as perfectly as I can create the reporting framework and report. Further, I have 100% control over both the reporting framework and the report examples.

There are numerous reporting styles provided for; not necessarily all, but enough so that you get the idea of how flexible XBRL-based model-based reporting is.

[145] Seattle Method, *PROOF*, https://xbrlsite.azurewebsites.net/2026/ref-github-proof/index2.html

[146] Seattle Method, *AASB 1060*, https://xbrlsite.azurewebsites.net/2026/ref-github-aasb1060-a/index2.html

Here is one structure randomly selected structure from the AASB 1060 reporting framework. There are about 70 in total.

Reporting Entity [Aspect]	GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442
Unit [Aspect]	iso4217:AUD

Concept [Aspect]	Period [Aspect] 2022-01-01 | 2022-12-31 Class [Dimension] Crops [Member]	Livestock [Member]	Other Miscellaneous Biological Assets [Member]	All Classes [Member]
Biological Assets [Roll Forward]				
Biological Assets, Beginning Balance	$ 0	$ 0	$ 0	$ 0
Gain (Loss) from Changes in Fair Value Less Costs to Sell	0	0	0	0
Increases from Purchases of Biological Assets	0	0	0	0
Decreases from Harvest of Biological Assets	0	0	0	0
Increases in Biological Assets from Business Combinations	0	0	0	0
Effects of Translation of Foreign Currency on Biological Assets	0	0	0	0
Other Increaes (Decreases) in Biological Assets	0	0	0	0
Biological Assets, Ending Balance	$ 0	$ 0	$ 0	$ 0

4.9. IFRS for SMEs

The IFRS for SMEs example[147] is a commercially available complete reporting framework which has been adjusted and augmented to be consistent with the best practices specified by the Seattle Method. I did not have control of this reporting framework, but it was possible to adjust, augment, and other supplement all of the missing pieces to make this consistent with best practices.

Here is just one of the approximately 75 structures which are provided for in this IFRS for SMEs financial reporting framework:

[147] Seattle Method, *IFRS for SMEs*, https://xbrlsite.azurewebsites.net/2026/ref-github-ifrs-smes/index2.html

Reporting Entity [Aspect]	1111111 | http://asab.gov/id
Unit [Aspect]	iso4217:USD

	Period [Aspect]								
	2025-01-01 | 2025-12-31								
	Measurement [axis]								
	At fair value [member]			At cost [member]			Aggregated measurement [member]		
	Carrying amount, accumulated depreciation, amortisation and impairment and gross carrying amount [axis]								
Concept [Aspect]	Gross carrying amount [member]	Accumulated depreciation, amortisation and impairment [member]	Carrying amount [member]	Gross carrying amount [member]	Accumulated depreciation, amortisation and impairment [member]	Carrying amount [member]	Gross carrying amount [member]	Accumulated depreciation, amortisation and impairment [member]	Carrying amount [member]
Disclosure of reconciliation of changes in biological assets [abstract]									
Reconciliation of changes in biological assets [roll forward]									
Biological assets at start of period	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0
Changes in biological assets [roll up]									
Gains (losses) on fair value adjustment, biological assets	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0
Additions other than through business combinations, biological assets	0	0	0	0	0	0	0	0	0
Decrease due to harvest, biological assets	0	0	0	0	0	0	0	0	0
Acquisitions through business combinations, biological assets	0	0	0	0	0	0	0	0	0
Increase (decrease) through net exchange differences, biological assets	0	0	0	0	0	0	0	0	0
Increase (decrease) through other changes, biological assets	0	0	0	0	0	0	0	0	0
Disposals, biological assets	0	0	0	0	0	0	0	0	0
Total increase (decrease) in biological assets	0	0	0	0	0	0	0	0	0
Biological assets at end of period	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0	$ 0

4.10. Enterprise Knowledge Graph (EKG)

This final example is a working proof of concept enterprise knowledge graph prototype[148]. The reason for this example is to make the point that a financial statement provides a "skeleton" of an enterprise knowledge graph[149].

All the files for this example are provided here[150].

---

148 Seattle Method, *Enterprise Knowledge Graph*, https://luca.pacioli.ai/luca/view/0f24fd35e961e167a727b663c75a4c5ec9fb7eb86730d6292f46e6e180fc2018980cd52e/index

149 Digital Financial Reporting, *Compliance Reporting as a "Skeleton" of the Enterprise Knowledge Graph*, https://digitalfinancialreporting.blogspot.com/2026/08/compliance-reporting-as-skeleton-of.html

150 Seattle Method, *EKG*, https://xbrlsite.azurewebsites.net/seattlemethod/platinum-testcases/ekg/index2.html