1. Expert Systems

This section provides a comprehensive introduction to expert systems that is approachable by professional accountants.

In the *Conceptual Overview of an XBRL-based, Structured Digital Financial Report*[1] I state that digital financial report creation software will be an expert system much like CAD/CAM software which is used to create blueprints. But what is an expert system? How do you create one? What makes expert systems work?

What does that mean? Does it mean that professional accountants will be replaced by machines? If the value that you add is simply clerical, amounting to cutting and pasting information or rekeying information, then quite possibly some of the tasks you perform could be replaced by automated machine-based processes. As explained in the document *Comprehensive Introduction to Knowledge Engineering for Professional Accountants*[2], there are certain tasks machines can perform and other tasks which machines will never likely be able to perform.

Thousands of expert system tools of all prices and qualities are commercially available today for performing different tasks. Because XBRL-based financial reports are structured data, computer software programs, such as expert systems, offer new capabilities specifically for financial reporting[3]. But what new capabilities will be offered? What work might be automated? This document is intended to help professional accountants understand what an expert system is, how they work, and what capabilities they will bring to help them perform work. Understating or overstating these capabilities are both not helpful.

The point is that setting the right expectations helps one understand what is actually practical and useful. Thinking that technology will have no impact on how you perform your work will be a complete disaster for your career as an accountant and/or for your business.

The global consultancy firm Gartner classifies XBRL as a transformational technology[4]. Gartner defines transformational as something that "*enables new ways of doing business across industries that will result in major shifts in industry dynamics*". Major shifts means lots of change and some winners and some losers.

This document helps professional accountants sort through all the information and misinformation that they are hearing related to XBRL-based structured digital financial reporting.

1.1. Deconstructing the Notion of an Expert System

To understand expert systems one first needs to understand the notion of an expert. This section is dedicated to setting your perspective as to the notion of an expert and

---

[1] *Conceptual Overview of an XBRL-based, Structured Digital Financial Report*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.1_ConceptualOverviewOfDigitalFinancialReporting.pdf

[2] *Introduction to Knowledge Engineering for Professional Accountants*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.3_KnowledgeEngineeringBasicsForProfessionalAccountants.pdf

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

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

the fundamental notion of an expert system. The section provides specific definitions, deconstructing the pieces so that we can subsequently put the pieces back together.

1.1.1. Definition of an expert

Google defines expert[5] as "a person who has comprehensive and authoritative knowledge of skill in a particular area".
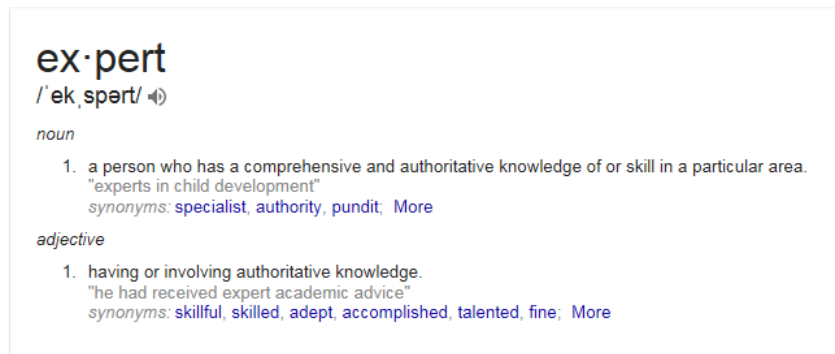
Professional accountants have a comprehensive and authoritative knowledge of and skills in the area of accounting and financial reporting. Accountants are experts at accounting and financial reporting.

1.1.2. Artificial intelligence

Artificial intelligence[6] is a branch of computer science. There are many good descriptions of artificial intelligence[7]. Here is one good definition:

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

Another more neutral term that basically means the same thing is "machine intelligence"[8]. Think of these tools as narrowly focused employees with great memories that are very good at performing one specific repetitive task well, over, and over, and over. Literally, like a machine.

Those trying to make artificial intelligence work over the past 40 or so years have had limited success[9]. But that is changing. People are putting the pieces together and the technology created from AI research are now available to experiment with. If expectations are not set too high, very useful functionality for limited, narrow problems can be successfully created. Both under estimating or over estimating the capabilities the computer software will be able to achieve can have catastrophic consequences.

One good example of using artificial intelligence is driverless cars. Many people get confused as to what is truly achievable and practical when it comes to driverless cars.

---

[5] Google search, *Expert definition*, retrieved August 21, 2016, https://www.google.com/search?q=expert+definition

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

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

[8] Shivon Zilis and James Cham, Harvard Business Review, *The Competitive Landscape for Machine Intelligence*, https://hbr.org/2016/11/the-competitive-landscape-for-machine-intelligence

[9] John F. Sowa, *Why Has AI Failed? And How Can it Succeed?*, http://www.jfsowa.com/pubs/micai.pdf

Driverless cars are on the streets of Singapore today[10]. While still in prototype mode now to work out details, by 2018 these taxis are anticipated to be commercially available. Uber is testing autonomous cars in Pittsburg[11].

One of the best ways to understand the capabilities of artificial intelligence is to try it out. Go test drive a Tesla which has driver-assist features.

Artificial intelligence programs that provide expert-level proficiency in solving problems by bringing to bear a body of knowledge about specific tasks are called knowledge based systems or expert systems.

There are two types of artificial intelligence, specialized and generalized:

- **Specialized**: An example of specialized artificial intelligence is programming a computer to play chess. The software performs one specific task. Specialized artificial intelligence is fairly easy to achieve.

- **Generalized**: An example of generalized artificial intelligence is the sort of science-fiction stuff you see in the movies. Why do you only see this in the movies? Because generalized artificial intelligence is extremely hard to make work.

1.1.3. Expert systems

Expert systems[12] is a branch of artificial intelligence. Expert systems, also called knowledge based systems or simply knowledge systems, are computer programs. The following is a definition of an expert system:

> Expert systems are computer programs that are built to mimic human behavior and knowledge. Expert systems are for reconstructing the expertise and reasoning capabilities of qualified experts within some limited, narrow domain of knowledge in machine-readable form. A model of the expertise of a domain of knowledge of the best practitioners or experts is formally represented in machine-readable form and the expert system reaches conclusions or takes actions based on that information when trying to solve some problem. The computer program performs tasks that would otherwise be performed by a human expert.

Expert systems are the most commercially successful applications of artificial intelligence research[13]. There are currently thousands of expert systems employed world-wide in industry and government.

Key to understating the capabilities of expert systems is an understanding the fundamental capabilities of computers. To understand the capabilities of computers,

---

[10] Reuters, *First driverless taxis hit the streets of Singapore*, http://www.reuters.com/article/us-autos-selfdriving-singapore-idUSKCN1100ZG

[11] The Atlantic, *Anybody Can Test a Driverless Car in Pennsylvania*, http://www.theatlantic.com/technology/archive/2016/09/anybody-can-test-a-self-driving-car-in-pennsylvania/499667/

[12] *Understanding the Components of an Expert System*, http://xbrl.squarespace.com/journal/2016/5/24/understanding-the-components-of-an-expert-system.html

[13] Edward Feigenbaum et. al, *KNOWLEDGE-BASED SYSTEMS IN JAPAN*, http://www.wtec.org/loyola/kb/execsum.htm

please be sure to read the document *Comprehensive Introduction to Knowledge Engineering for Professional Accountants[14]*.

1.1.4.Differentiating the mechanical aspects of a financial report and judgment

Computers are dumb beasts. Computers only follow instructions. Computers will never possess judgment. They may try and mimic judgment and could be successful to some degree.

A financial report itself is mechanical. While what must go into the financial report requires the judgement of a professional accountant, the mechanics of a financial report are objective. Balance sheets always balance[15]. Roll ups always roll up. Roll forwards always roll forward. Everything should always "tick and tie", "cross cast and foot". Accountants excel at performing these detailed tasks. But computers are completely capable of managing the mechanical details of a financial report. That frees professional accountants from having to worry about those mechanical details and focus on where they add the most value which is the aspects of creating a financial report that require judgment.

Not making the proper distinction between the mechanical aspects and the aspects that require judgement will cause someone to either overestimate the work a computer can perform or under estimate that work.

1.1.5.Business rules

Key to employing artificial intelligence and therefore making an expert system work is machine-readable business rules[16] of a domain of knowledge.

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

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

1.1.6.Objects as rules

Besides business rules, expert systems can use another form of knowledge representation. Objects, rather than rules, can play a leading role in knowledge representation. Although, object-based representations of knowledge can be regarded as rules organized in a different manner.

1.1.7.Human-readable and machine-readable business rules

Business professionals interact with business rules every day by may not even realize it. Most business rules are in human readable form. But business rules can be

---

[14] *Introduction to Knowledge Engineering for Professional Accountants*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.3_KnowledgeEngineeringBasicsForProfessionalAccountants.pdf

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

[16] *Comprehensive Introduction to Business Rules*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.4_ComprehensiveIntroductionToBusinessRules.pdf

represented in both human-readable form and machine-readable form. With the move to digital, more and more business rules are being represented in both human readable form and more importantly machine-readable form. Formalized, standardized machine-readable business rules can help automate processes which have been manual in the past.
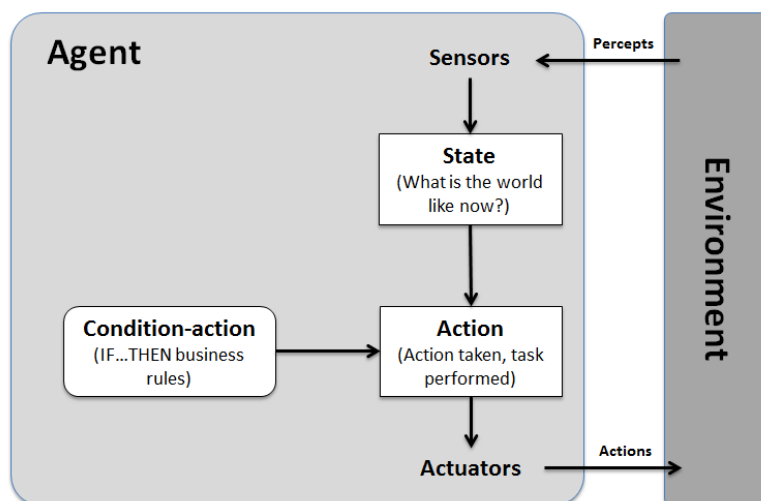
1.1.8. Intelligent software agents

As stated, artificial intelligence is the automation of activities that we associate with human thinking and activities such as decision making, problem solving, learning and so on. How do you create that sort of automation? The answer is using a style of computer programming called intelligent software agents.

An intelligent software agent[17] is software that assists people and acts on their behalf. Intelligent agents add value by allowing people to:

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

An intelligent agent is computer software capable of **sensing** the **state** of its **environment** and **acting** upon it based on a set of specified **rules**. An intelligent agent performs specific tasks on behalf of another. In the case of software, an agent is a software program. There are many different types of intelligent software agents[18].

Simple Reflex Agent

---

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

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

The document *Comprehensive Introduction to Intelligent Software Agents for Professional Accountants*[19] goes into significantly more detail on the topic of intelligent software agents.

1.2. Distinguishing Between Rule-based and Pattern-based Artificial Intelligence

The document *Expert System for Creating Financial Reports*[20] outlines the vision of a rules-based expert system for creating financial reports that leverages explainable artificial intelligence (XAI)[21]. This vision is based on working software applications, working proof of concepts, reverse engineering of XBRL-based financial reports submitted to the SEC.

But there is another type of artificial intelligence which is patterns-based which is also known as machine learning.

1.2.1. Artificial Intelligence

There are two approaches to artificial intelligence and, as I have said before, the right approach should be used for the given job[22]. The two approaches are:

- **Rules-based systems** (symbolic systems, good at reasoning, expert systems, three basic types)
  - o **Classification or diagnosis type**: helps users of the system select from a set of given alternatives.
  - o **Construction type**: helps users of the system assemble something from given primitive components.
  - o **Simulation type**: helps users of the system understand how some model reacts to certain inputs.
- **Patterns-based systems** (non-symbolic systems, good at learning, machine learning which can be supervised or unsupervised, five basic types[23])
  - o **Clustering algorithms**: categorize or group things
  - o **Explanatory algorithms**: explain the relationships between variables
  - o **Ensemble learning algorithms**: use multiple models
  - o **Similarity algorithms**: compute the similarity of pairs of things
  - o **Dimensionality reduction algorithms**: reduces variables in a dataset

---

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

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

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

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

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

Another helpful breakdown of pattern-based systems is provided by the article *Top Machine Learning Algorithms for Prediction: A Short Summary*[24].

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

Currently, the expert systems working proof of concepts and commercial software that I have created and experimented with only leverage rules-based systems. But it is

---

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

pretty clear that machine learning offers additional capabilities that will be inevitably leveraged.

Pacioli[25] is a logic/rules/reasoning/knowledge/insights engine. Frankly, I don't know exactly what to call Pacioli. At first, I called Pacioli a logic/rules/reasoning engine. Then I referred to it as a knowledge engine[26]. Now I am contemplating that Pacioli is an insights engine. For now, I consider Pacioli a logic/reasoning/rules/knowledge/insights engine in order to be complete, I guess.

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

The second role of Pacioli is to leverage the machine-readable logical schema to work with the information that makes up the financial report knowledge graph[27].

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

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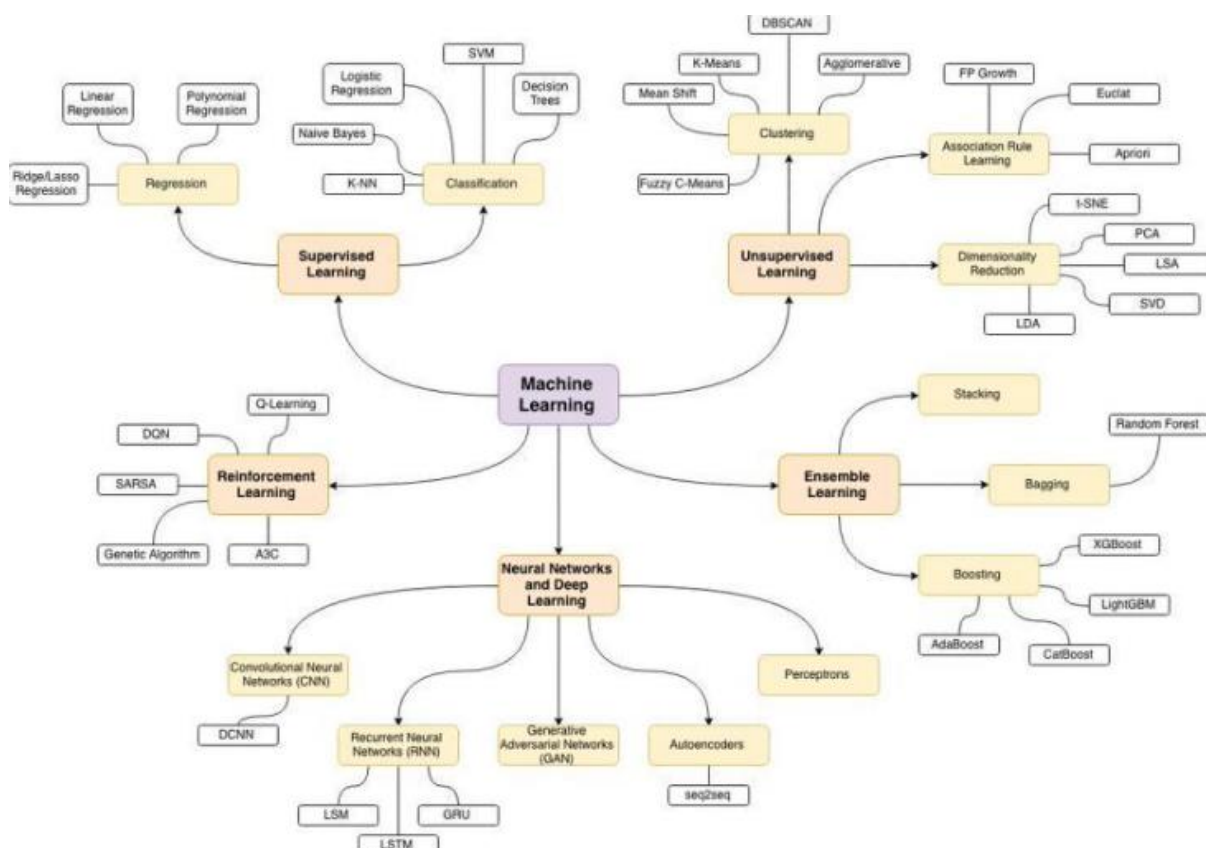
[25] Auditchain, *Pacioli Logic and Rules Engine*, https://docs.auditchain.finance/auditchain-protocol/pacioli-logic-and-rules-engine

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

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

1.2.2.Programmed Ability to Process Information

DARPA and PWC do an excellent job of explaining the capabilities of artificial intelligence. This presentation[28], video[29] and this article[30] provide a summary worth reading.
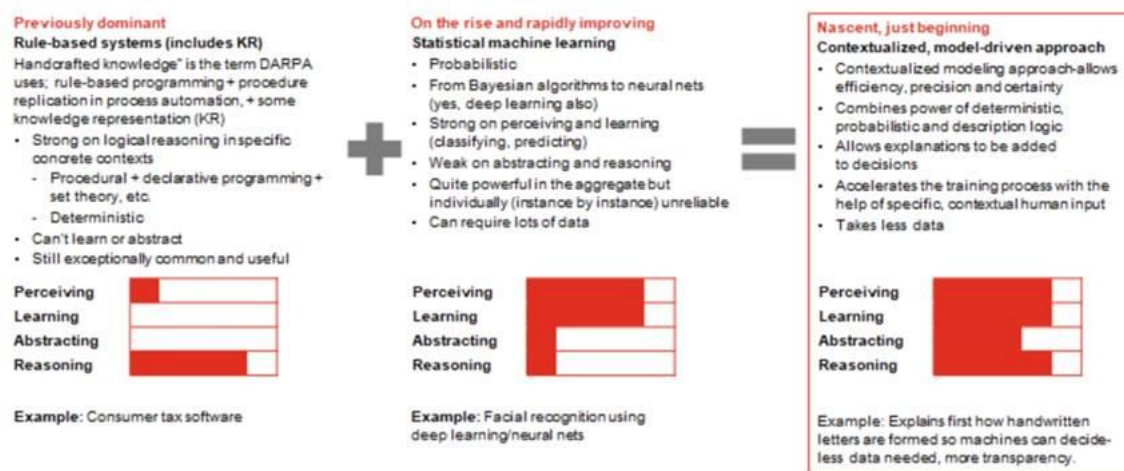
The graphic above shows three approaches to implementing artificial intelligence or the programmed ability to process information. Auditchain's Pacioli is a "rule-based system" that uses "handcrafted knowledge" that I created (the LEFT SIDE). Statistical machine learning will be added (the MIDDLE) to Pacioli over time most likely. Ultimately, where Auditchain and others will end up is on the RIGHT side which is the combined capabilities of rules-based systems and statistical machine learning. I suspect that this will occur within three to five years.

Here is another explanation that was inspired from a DARPA presentation[31]. Artificial intelligence is defined as a "**programmed ability to process information**". Sure, software like, Excel and Word can process "stuff"; but they don't really understand anything about accounting, reporting, auditing, or analysis. By "understand", this is what is meant, paraphrasing form the DARPA explanations of these terms:

- **Perceive**: ability to perceive rich, complex, subtle information.

- **Learn**: ability to learn about that information within a specific environment such as within a financial report.

- **Abstract**: ability to create new meaning from existing information.

- **Reason**: ability to plan, ability to decide, ability to verify.

---

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

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

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

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

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

| Perceiving	
Learning	
Abstracting	
Reasoning	

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

Perceiving	
Learning	
Abstracting	
Reasoning	

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

| Perceiving	
Learning	
Abstracting	
Reasoning	

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

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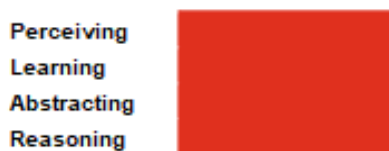
To summarize, there are three very important points to remember here. First, the right artificial intelligence should be used for each automation task. Second, handcrafted metadata is necessary to make automation actually work. Finally, while it is easy to discuss automation in general high level terms;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

1.2.3.Data Science and Business Science Connection

In the information age, data science[32] and business science[33] are connected. This graphic, the business science problem framework[34], helps one understand that connection.
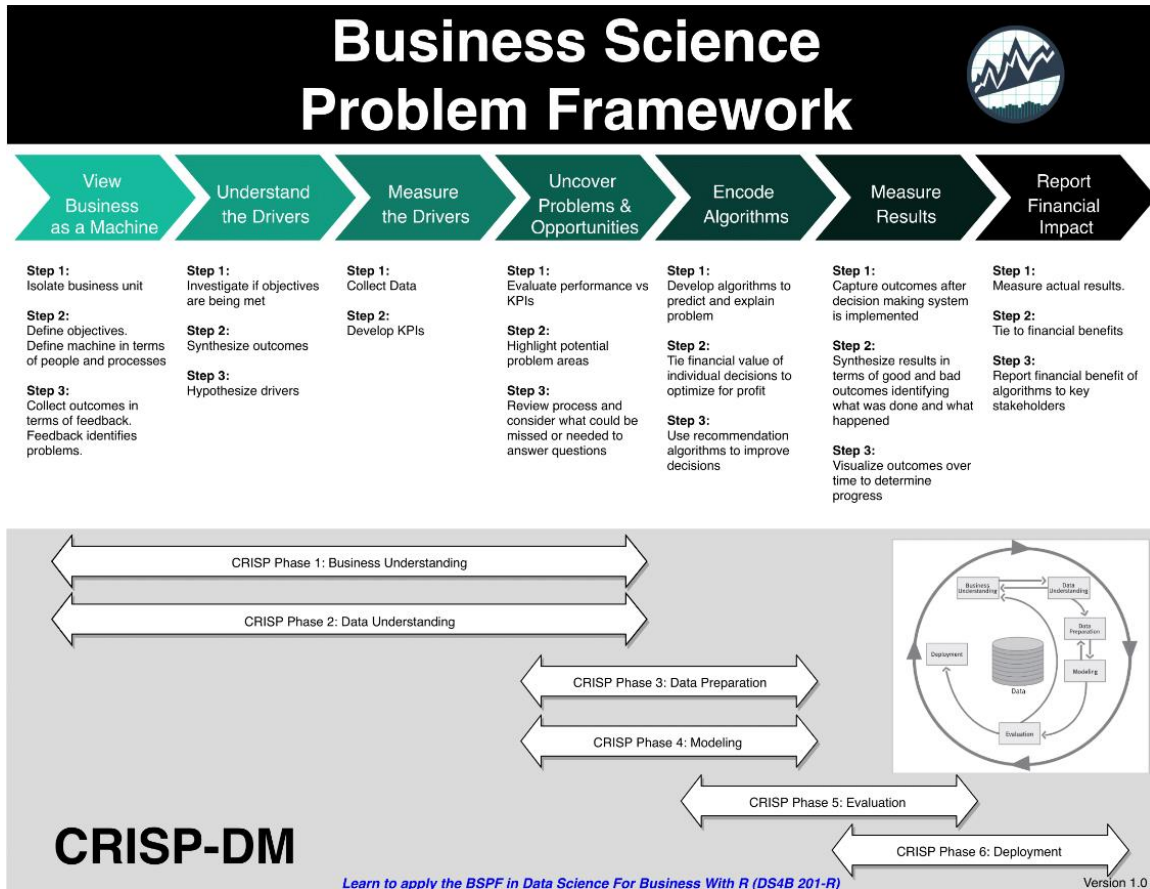
1.2.4.ChatGPT, Computational Reducibility and Computational Irreducibility.

Another article by Stephen Wolfram, *Will AIs Take All Our Jobs and End Human History—or Not? Well, It's Complicated…*[35], pulls ChatGPT into this discussion and uses different terms to describe rules-based and patterns-based artificial intelligence. That article uses the terms "computational reducibility" and "computational irreducibility". It seems to me that computational reducibility and rules-based are the same and computational irreducibility and patterns-based are the same.

---

[32] Data Science Cheat Sheets, https://github.com/FavioVazquez/ds-cheatsheets

[33] Business Science, https://www.business-science.io/bspf.html

[34] Business Science Problem Framework, https://raw.githubusercontent.com/FavioVazquez/ds-cheatsheets/master/Business_Science/img/Business_Science_Problem_Framework.png

[35] Stephen Wolfram, Will AIs Take All Our Jobs and End Human History—or Not? Well, It's Complicated…, https://writings.stephenwolfram.com/2023/03/will-ais-take-all-our-jobs-and-end-human-history-or-not-well-its-complicated/

ChatGPT[36], or more precisely GPT-4[37], seems to be taking the world by storm. People have made GPT-4 pass the bar exam, pass the CPA exam, complete a tax return, and help post an accounting transaction. Interesting stuff.

This is my summary of the article by Stephen Wolfram. That article is worth reading even though it is pretty long. My summary is:

1. WAY more things will be effectively automatable than I had expected.

2. There are two approaches to automation, those with "computational reducibility" and those that have the characteristic of "computational irreducibility". Those two types have critically important characteristics that need to be understood. These two approaches can be combined into a hybrid.

3. Pacioli is a "computational reducibility" type system, it can be used to create "pockets of computational reducibility" where things that are "higher level" things can be created, or "modular units" of things where a "proof chain" can be created (i.e. because of the computational reducibility); and work can be effectively automated. This type of system is precise and predictable. There is an "audit trail".

4. ChatGPT is a "computational irreducibility" type system. You don't know exactly how it reaches conclusions, and it is capable of providing useful work, but it is just as likely as having "unintended consequences" and the "possibility of surprise" and other such "wild behavior". But it can be pointed in the right direction and be quite useful for specific types of tasks.

5. It is critically important for accountants to grasp the difference between the capabilities of "computational reducibility" and "computational irreducibility". Using the wrong tool for the job will not provide much satisfaction.

There is a lot more provided in that article, such as the "cycle of new occupations". Again, I say, that article is worth reading.

1.3. Digging Deeper into Expert Systems and Knowledge Based Systems

The previous section sets a foundation for understanding expert systems. In this section we go into additional important details that help round out your understanding of expert systems and other knowledge based systems.

Simply put, a **knowledge based system** is a system that draws upon the knowledge of human experts that has been represented in machine-readable form and stored in a **fact database** and **knowledge base**. The system applies **problem solving logic** using a **problem solving method** to solve problems that normally would require human effort and thought to solve. The knowledge based system supplies an **explanation and justification mechanism** to help system users to understand the **line of reasoning** used and support **conclusions reached** by the knowledge based system and presents that information to the user of the system.

An expert system is a type of knowledge based system.

---

[36] ChatGPT, https://digitalfinancialreporting.blogspot.com/2022/12/someone-made-me-aware-of-chatgpt-couple.html

[37] Arxiv.org, GPT-4 Technical Report, https://arxiv.org/abs/2303.08774

Humans augmented by machine capabilities, much like an electronic calculator enabling a human to do math quicker, will empower knowledge workers who know how to leverage the use of those machines.

1.3.1. Creasing an expert system or knowledge based system

Creating a knowledge based system involves the transformation of machine-readable instructions in such a way as to explain to a machine how a system works and how to make a system work the way you want that system to work.

Then, brick-by-brick, much like building a house, business domain experts and software engineers can create tools that automate certain types of tasks in that process. Humans encode information, represent knowledge, and share meaning using machine-readable patterns, languages, and logic.

That will be the way an increasing number of work tasks will be performed in the Digital Age of accounting, reporting, and auditing. The result will be more efficient processes.

1.3.2. Types of expert systems

Frank Puppe explains in his book *Systematic Introduction to Expert Systems*[38] that there are three general categories of expert systems:

- **Classification or diagnosis type**: helps users of the system select from a set of given alternatives. The system tends to be instructional in nature.
- **Construction type**: helps users of the system assemble something from given primitive components.
- **Simulation type**: helps users of the system understand how some model reacts to certain inputs or create predictions based on the system.

The assembly of a financial report can be assisted by a construction-type expert system. Helping professional accountants understand what goes into that financial report can be assisted by a classification-type expert system. Creating forecasts and projections of future financial reports can be assisted by simulation-type expert systems.

1.3.3. Components of an expert system

A software based expert system has four primary components[39]:

- **Database of facts**: A database of facts is a set of observations about some current situation or instance. The database of facts is "flexible" in that they apply to the current situation. The database of facts is machine-readable. An example of a database of facts is information reported a financial report. An XBRL instance is a database of facts.
- **Knowledge base of rules**: A knowledge base is a set of universally applicable rules created based on experience and knowledge of the practices of the best domain experts generally articulated in the form of IF…THEN statements or a form that can be converted to IF...THEN form. At the highest level the

---

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

[39] Edward Feigenbaum Chair, et. al., *KNOWLEDGE-BASED SYSTEMS IN JAPAN*, http://www.wtec.org/loyola/kb/c3_s2.htm

knowledge base contains a conceptual model, definitions of things that make up that model, and relationships between the things in the model (types of things, structure of things, parts of things, mathematical relations between things, etc.). A knowledge base is "fixed" in that its rules are universally relevant to all situations covered by the knowledge base. Not all rules are relevant to every situation. But where a rule is applicable it is universally applicable. All knowledge base information is machine-readable. Business rules are declarative in order to maximize use of the rules and make it easy to maintain business rules. Knowledge that makes up the knowledge base is acquired using manual or automated knowledge acquisition processes. An XBRL taxonomy is a knowledge base of rules.

- **Reasoning/inference/rules engine**: A reasoning engine provides a machine-based line of reasoning for solving problems. The reasoning engine processes facts in the fact database, rules in the knowledge base. A reasoning engine is also an inference engine and takes existing information in the knowledge base and the database of facts and uses that information to reach conclusions or take actions. The inference engine derives new facts from existing facts using the rules of logic. The reasoning engine is a machine that processes the information. A reasoning engine has a specific problem solving logic and uses some problem solving method. An XBRL Formula processor, if built correctly, can be a reasoning engine and can perform logical inference.

- **Justification and explanation mechanism**: When an answer to a problem is questionable, we tend to want to know the rationale behind the answer. If the rationale seems plausible, we tend to believe the answer. The justification and explanation mechanism explains and justifies how a conclusion or conclusions are reached. It walks you through which facts and which rules were used and the line of reasoning used to reach a conclusion. The explanation mechanism is the results of processing the information using the rules processor/inference engine and justifies why the conclusion was reached. The explanation mechanism provides both provenance and transparency to the user of the expert system so that the user of the system understands the origin of all facts and rules.

- **Knowledge acquisition mechanism**: The power of a knowledge base system is proportional to the key ingredient of that system: high-quality domain knowledge available in machine-readable form. Knowledge acquisition can be by manual creation of rules by business professionals, machine-based processes for generating rules, or a combination of the two.
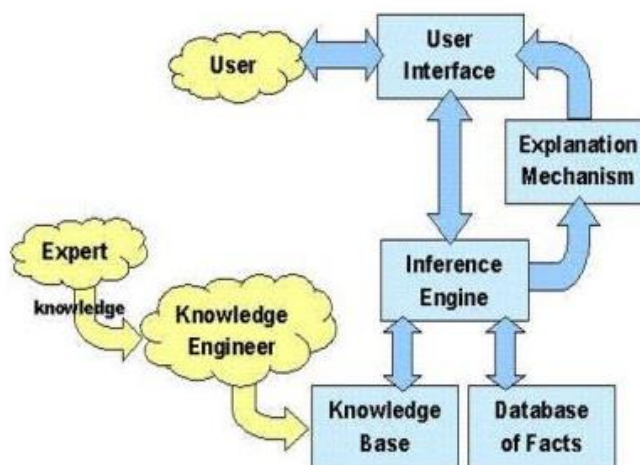
These pieces are exposed to the users of the expert system within software applications. One philosophical difference that knowledge based systems tend to have from typical software systems which tend to be procedural in nature is that knowledge based systems separate business domain logic and program control logic. This enables business domain logic to be managed/maintained by business domain experts and that the business domain logic is reusable by other software programs. Below is a summary of the components of an expert system or knowledge based system:
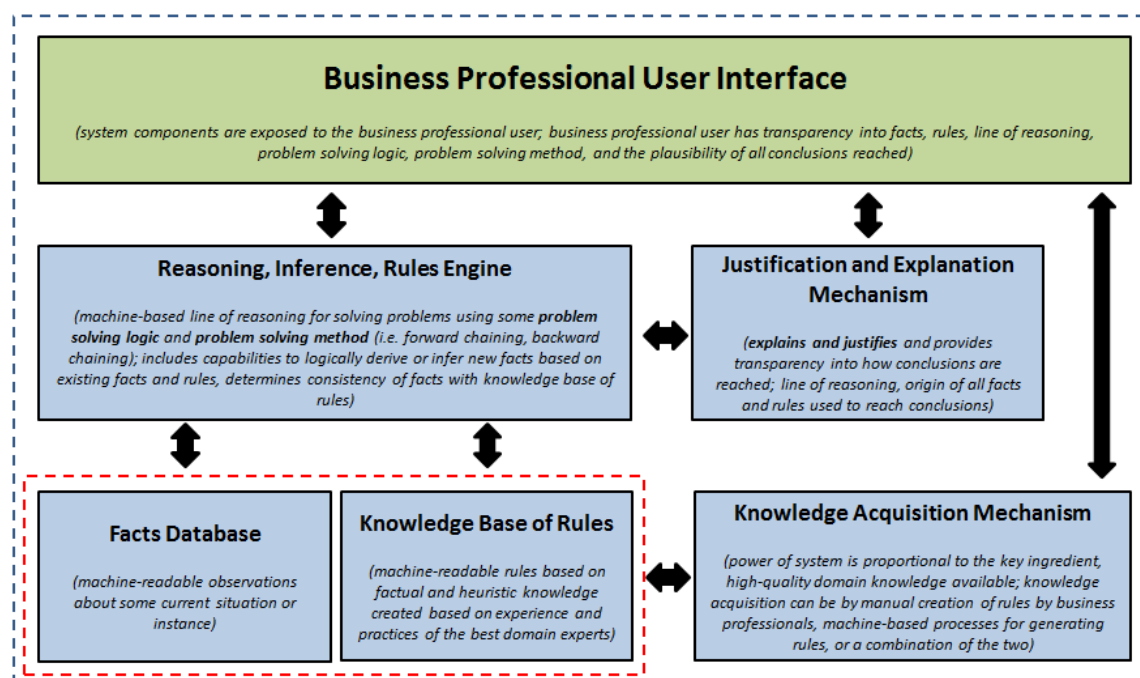
Expert systems provide *transparency* to their users and can explain the solutions they provide by quoting the knowledge used to reach that solution. Single pieces of knowledge can be easily added, changed, or removed; providing *flexibility*. The users of expert systems should require no knowledge of programming languages by either the creator of the expert system or user of the expert system, providing *ease of use*. The boundaries of the problem solving capabilities should be clear so users of the system understand what the system provides and what needs to be provided using alternative processes.

The following is another graphic[40] that shows how the components of an expert system interact with one another:

---

[40] IMS MBA, *Architecture of an Expert System*, https://imscdrmba.wordpress.com/206-unit-iii/ ; (direct line to image) https://imscdrmba.files.wordpress.com/2016/04/expert-systems.jpg

Software applications must provide all the pieces of the system. The *law of irreducible complexity* states basically that "A single system which is composed of several interacting parts that contribute to the basic function, and where the removal of any one of the parts causes the system to effectively cease functioning." That means that each of the parts need to exist for the system of an XBRL-based digital financial report to work correctly.

Further, the system must be usable by business professionals. The *law of conservation of complexity* essentially states, "Every software application has an inherent amount of irreducible complexity. That complexity cannot be removed from the software application. However, complexity can be moved. The question is: Who will have to deal with the complexity? Will it be the application user, the application developer, or the platform developer which the application leverages?"

1.3.4.Knowledge acquisition

A key ingredient in a knowledge based system is domain knowledge. The power of any knowledge based system is proportional to the high-quality domain knowledge available within that system. The fact that a thick metadata layer and the benefits of that metadata in terms of getting a computer to be able to perform useful and meaningful work is not disputed. What is sometimes disputed is *how* to most effectively and efficiently get that thick metadata layer. There are two basic approaches to getting this metadata, or machine-readable business rules, that makes up the knowledge base:

- **Have the computer figure out what the metadata is**: This approach uses artificial intelligence, machine learning, and other high-tech approaches to detecting patterns and figuring out the metadata.

- **Tell the computer what the metadata is**: This approach leverages business domain experts and knowledge engineers to piece together the metadata manually so that the metadata becomes available.

Because knowledge acquisition can be slow and tedious, much of the future of knowledge based systems depends on breaking the knowledge acquisition bottleneck and in codifying and representing a large knowledge infrastructure. However, this is not an "either/or" question. Both manual and automated knowledge acquisition methods can be used together.

1.3.5.Problem solving method

The objective of an expert system is to solve some problem. Conventional software applications work using sequential algorithms (software programs) and data. Expert systems separate the "algorithms" into two parts; knowledge and the problem solving method. Another way to say this is that expert systems separate software programs into domain logic and control logic.

- Forward chaining
- Backward chaining
- Combination of forward and backward chaining

Please see the document *Comprehensive Introduction to Business Rules for Professional Accountants*[41] for a more complete introduction to forward and backward chaining.

1.3.6. Problem solving logic

One of the most complicated things to understand about expert systems is the problem solving logic used by the expert system to solve problems. The problem solving logic drives the extent of rules that can be created because the nature of the logic constrains what rules can be expressed.

A logic can be defined as any precise notation for expressing statements that can be judged to be either true or false[42]. Almost any declarative notation, graphic or linear, could be treated as a version of logic: just specify it precisely. A logic needs to define foundational terminology such as: *there exists*, *every*, *and*, *or*, *if and only if*, *if-then*, *not*, *true*, and *false*[43]. Really, it is that simple. What makes this complicated are all the different notations that are used to refer to those terms. Using natural language make logic more understandable to business professionals.

Determining the problem solving logic is a balancing act. The objective is to have the maximum amount of expressiveness but the minimum chance that software will break. The logic needs to be safely implementable by software.

For the past 30 or so years, many different technical solutions have been created to solve different business problems. Few of these technical solutions achieved the appropriate balance or equilibrium and tended to not maximize expressiveness or not be safely implementable in the form of software applications. Benjamin Grosof, Michael Kifer, and Mike Dean summarize this history in their presentation, *Semantic Web Rules: Fundamentals, Applications, and Standards*[44].

Please see the document *Comprehensive Introduction to Knowledge Engineering for Professional Accountants*[45] for a complete discussion of this topic.

The bottom line is that the best balance between expressiveness and safe implementation has been achieved by the ISO/IEC global standard Common Logic. **Common Logic**[46] is a framework for a family of logic languages, based on first-order logic, intended to facilitate the exchange and transmission of knowledge in computer-based systems. That safely expressive sweet spot is also used by the OMG standard **Semantics of Business Vocabulary and Business Rules**[47] which was consciously designed to be logically equivalent to ISO/IEC Common Logic.

---

[41] *Comprehensive Introduction to Business Rules*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.4_ComprehensiveIntroductionToBusinessRules.pdf

[42] John F. Sowa, *Fads and Fallacies about Logic*, page 2, http://www.jfsowa.com/pubs/fflogic.pdf

[43] Wikipedia, *List of Logical Symbols*, https://en.wikipedia.org/wiki/List_of_logic_symbols

[44] Benjamin Grosof, Michael Kifer, and Mike Dean, *Semantic Web Rules: Fundamentals, Applications, and Standards*; http://coherentknowledge.com/wp-content/uploads/2013/05/talk-prelim-aaai13-rules-tutorial.pdf

[45] *Introduction to Knowledge Engineering for Professional Accountants*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.3_KnowledgeEngineeringBasicsForProfessionalAccountants.pdf

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

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

Rulelog[48] is a logic for expressing knowledge that is consciously engineered to be consistent with ISO/IEC Common Logic and OMG Semantics of Business Vocabulary and Business Rules. Rulelog is a dialect of W3C's RIF[49]. RuleML[50] is a syntax for implementing rules. Other standard and proprietary syntaxes exist for implementing rules.

The most important thing to realize is that there is a good, safe target in terms of an expressive logic that is also safely implementable in software so catastrophic failures are avoided. Another very good thing is that business professionals don't need to understand the underlying technical details of these logic standards, nor will they ever have to deal with them. Higher level languages that follow the foundations set by Common Logic, Semantics of Business Vocabulary and Business Rules, and Rulelog.

The following graphic shows the role Common Logic[51] plays, establishing a family of logical dialects shared between different software syntax implementations: (note that this graphic was modified, XBRL was added)
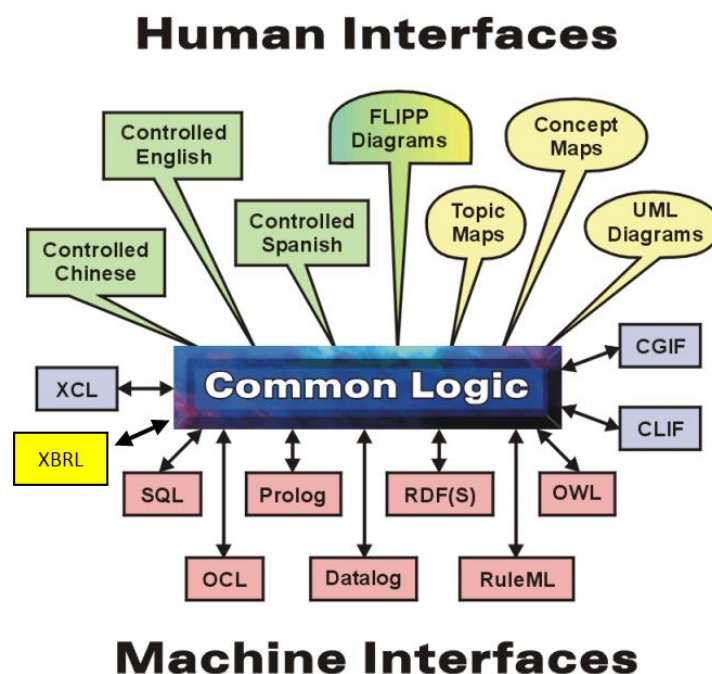
The W3C recommended standard syntax is RDF/OWL/SHACL, the semantic web stack. The language in which a problem is stated has no effect on complexity. Reducing the expressive power of a logic does not solve any problems faster; its only effect is to make some problems impossible to state[52].

---

[48] *Rulelog*, http://ruleml.org/rif/rulelog/spec/Rulelog.html

[49] W3C, *RIF Overview (Second Addition)*, http://www.w3.org/TR/rif-overview/

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

[51] John F. Sowa, *Common Logic: A Framework for a Family Of Logic-Based Languages*, page 5, http://www.jfsowa.com/ikl/SowaST08.pdf

[52] John F. Sowa, *Fads and Fallacies about Logic*, page 5, http://www.jfsowa.com/pubs/fflogic.pdf

1.3.7.Inference techniques

There are various types of inference techniques available[53]. See the problem solving logic document.

1.3.8.Benefits of an expert system

Benefits from the use of expert systems include:

- **Reduced costs by using automation**: elimination of routine, boring, repetitive, mundane, mechanical, rote tasks that can be automated

- **Increased uniformity**: consistent answers from the same question or facts; computers are good at performing repetitive, mechanical tasks whereas humans are not; computers do not make mistakes and are good at repeating exactly the same thing each time; performance level is consistent

- **Reduced down-time**: computer based expert systems are tireless and do not get distracted

- **Increased availability**: computer based expert systems are always available simultaneously in multiple places at one time; you get quick response times and can replace absent or scarce experts; convenient

- **Diligence and tenacity**: computers excel at paying attention to detail; they never get bored or overwhelmed and they are always available and will keep doing their job until the task is complete with the same attention to detail

- **Basis for training**: the best practices of the best practitioners can be available to those that are new to and learning about a domain of knowledge

- **Longevity and persistence**: computer based expert systems do not change jobs or retire so knowledge gathered by an organization can remain within that organization

- **Productivity**: computer based expert systems are cheaper that hiring experts and costs can be reduced a the same time that quality increases resulting in increased productivity

- **Multiple opinions**: Systems can integrate the view of multiple experts within a system and choose between the preferred view of multiple expert opinions in the same system

- **Objectivity**: computers apply the same inductive and deductive logic consistently; emotion and personal preferences can be eliminated where they should be eliminated; expert systems do not discriminate

- **Easier dissemination of knowledge**: expert systems are software and metadata and therefore once you have that software and metadata reproducing another version is trivial and the incremental cost is extremely low

In a knowledge based or expert system; knowledge is explicitly represented and can be evaluated, knowledge is permanent, knowledge is easily replicated, and the system is consistent. Operating costs of an expert system are low. Financial report creation

---

[53] Decision Support Systems and Intelligent Systems, Efraim Turban and Jay E. Aronson
6th ed, Copyright 2001, Prentice Hall, Upper Saddle River, NJ, *Inference Techniques*, Chapter 13,
http://www.indiana.edu/~bnwrbk/K510/ch13.ppt

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

1.3.9. Disadvantages of expert systems

Everything has advantages and disadvantages. The following can be disadvantages of expert systems:

- **Initial cost**: the initial cost of creating an expert system can be high; the primary cost is for creation of the expert knowledge which is used by the system

- **Maintaining knowledge**: human experts constantly update their knowledge through interaction with other experts, new ideas, common sense, etc.; expert systems have to be maintained to keep knowledge current

- **Garbage in, garbage out**: an expert system is only as good as the machine-readable knowledge which the system uses

- **No common sense**: humans have common sense, expert systems do not

- **Lacks human touch**: expert systems are computer application and have the same characteristics of a computer; they have no compassion, no intuition, cannot exercise real judgment, etc.

- **Inflexibility**: a system, once set up, is inflexible or rather only flexible to the extent that new knowledge is added to the system

- **Restricted**: an expert system usually has expertise in one specific domain of knowledge and is therefore restricted to that specific knowledge

To make expert systems work effectively, disadvantages must be overcome and expert systems should be used to solve problems they are truly capable of solving. Setting the right expectations is important.

1.3.10. Contrasting universal tools and domain specific tools

In his book Systematic Introduction to Expert Systems[54], Frank Puppe provides the graphic below. The graphic basically points out that universal, general tools are less restrictive but cost more to create domain-specific tools. In addition to universal, general tools being more costly to create and more difficult to create; domain specific tools are easier to create and much, much easier for business professionals to use because of the restrictions.

So, a "restriction" is not a flaw. The restriction is what makes the tool easier to use and cost less and make easier to develop. You don't need the universe of all possible options; you only need to create what that specific domain needs. As long as you get these restrictions correct, they really are not "restrictions" of the domain, they are the "boundaries" of the domain. You don't need them. Technical people don't typically understand these domain boundaries. Many times to play it safe technical people add flexibility in order to make certain that business domain user needs are being met. But this flexibility comes at a cost. Additional costs are incurred to create the flexibility and software is harder to use because business professionals need to figure out which option they should use.

---

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

Business domain people do understand the boundaries if they think about them. Many business professionals cannot properly articulate the appropriate boundaries or restrictions. This communications problem tends to lead to software that costs more to create than is necessary and harder to use than necessary.
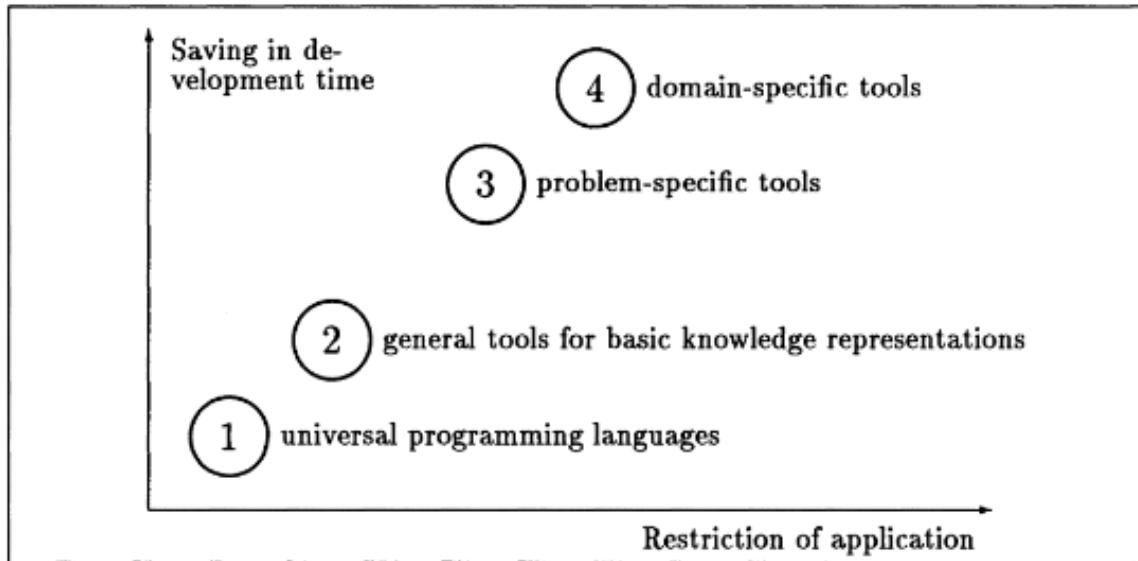
This is not an either-or choice. Sometimes universal tools are very appropriate. Other times domain-specific tools are appropriate. Being conscious of these dynamics will lead to the right software being created and the appropriate level of usability. Universal tools are not a panacea. Unconsciously constricting a domain-specific tool when it would have been better to create a more universally usable tool also can be a mistake one makes.

1.3.11. Examples of expert systems in other domains

Expert systems are available commercially at different price levels and with different capabilities. The following is a brief list of expert systems to give you an idea of the potential capabilities of expert systems:

- Chess game (for example, IBM's Deep Blue beat the grand master at the time)
- Medical triage and diagnosis
- Robotic surgery
- Aircraft accident investigation
- Patriot missile guidance system
- Numerically controlled manufacturing machine

1.4. Using CLIPS to Understand Expert Systems

CLIPS[55] is a tool for building expert systems originally developed by NASA. Since it was first released in 1986 it has undergone significant enhancements and was put into the public domain by NASA in about 2002. CLIPS continues to be maintained as public domain software.

While CLIPS is not a tool the average professional accountant will use, CLIPS does offer a way to understand how expert systems work.

Recognize that CLIPS is a system for building any expert system that you want. As such, you have to understand how to put the pieces of an expert system together. Professional accountants will not have to do this for expert systems which create financial reports. Just as architects don't have to build computer aided design or computer aided manufacturing expert systems from scratch, neither will accountants. Software developers will create expert systems that professional accountants will use.

However, some professional accountants will build expert systems. Tools such as CLIPS allow someone who understand logic programming to create expert systems for other domains. For example, a small business might have a task they perform manually which they might want replaced by an automated expert system. Professional accountants will help small businesses create such micro expert systems. This could be a niche service offered by professional accountants.

1.4.1.Overview of CLIPS

Consider the software application interface of CLIPS below:
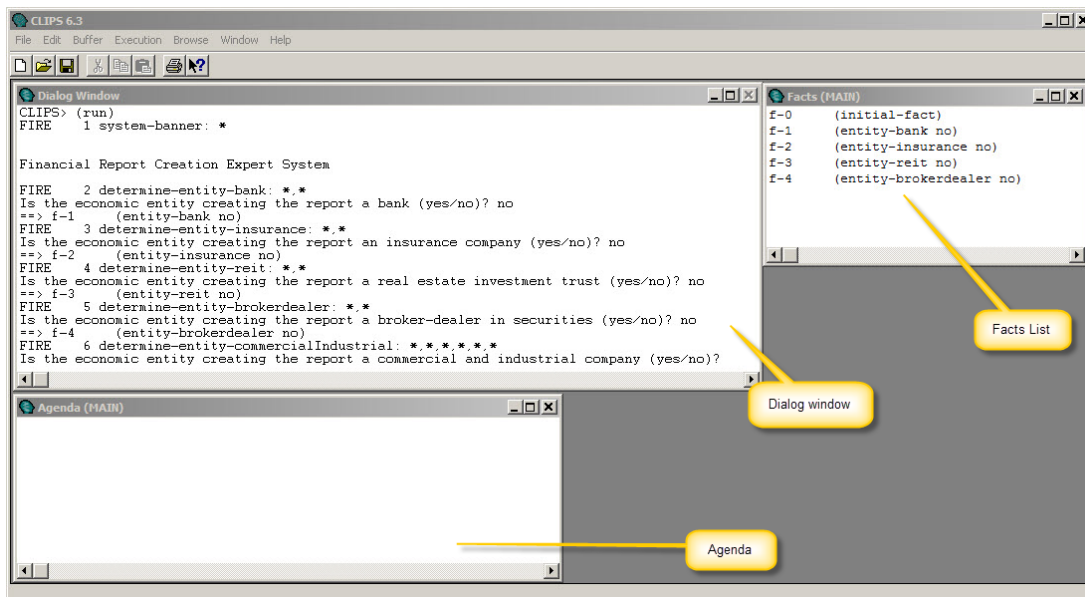

The interface is showing three parts:

---

[55] *Using CLIPS to Understand Expert Systems and Logic Programming,* http://xbrl.squarespace.com/journal/2016/9/15/using-clips-to-understand-expert-systems-and-logic-programmi.html

- **Fact list**: In the upper right hand corner you see a facts list or "database of facts".
- **Agenda**: In the lower left side you see an agenda window. Currently the agenda window is empty which means the expert system has no more tasks to complete.
- **Dialog window**: In the upper left, the largest window is the dialog window. The dialog window is where the user interacts with the expert system software.

Remember that CLIPS is a universal programming language that is designed to enable anyone with the necessary skills to build any sort of expert system. That is why the interface is general. So please don't get distracted by the nature of the interface. Focus on the logic of how an expert system works.

When the expert system starts, the facts list and the agenda look as follows:

Facts list:
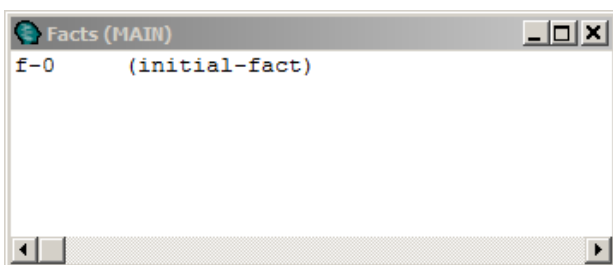
Agenda:
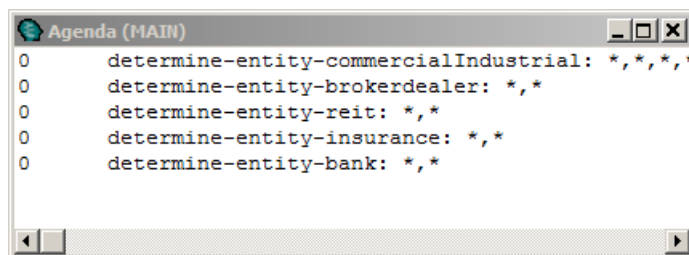
The fact list is empty (except for a default fact) and then the agenda has five activations. Basically, no facts are known (in the case of this system) and there are five items on the agenda to determine the accounting activity of the economic entity creating the financial report.

This is the terminology used by the CLIPS system for building expert systems which will give you an idea of how you interact with CLIPS. CLIPS uses a forward chaining problem solving method.

- **Goal**: The system will cease execution when no *activations* are on the *agenda*.
- **Strategy**: High-level plan for achieving a goal.

- **Fact**: A fact is the same as an XBRL definition of a fact.
- **Fact list**: A fact list is the set of facts the system is currently working with to arrive at some goal.
- **Rule**: A rule is a relation between facts or a relation between a fact and the characteristics or traits of a fact.
- **Assertion**: Assert a new fact.
- **Retraction**: Retract an existing fact.
- **Activation**: An activation is a rule that is active because it matches the forward chaining strategy.
- **Agenda**: The agenda is a collection of activations which are those rules which match pattern entities. Zero or more activations may be on the agenda.
- **Salience**: When multiple activations are on the agenda, the system automatically determines which activation is appropriate to fire. The system orders the activations on the agenda in terms of increasing priority or salience.
- **Depth strategy**: In the depth strategy, new activations are placed on the agenda after activations with higher salience, but before activations with equal or lower salience. All this simply means is that the agenda is ordered from highest to lowest salience.
- **Conflict resolution**: The inference engine sorts the activations according to their salience. This sorting process is called conflict resolution because it eliminates the conflict of deciding which rule should fire next.
- **Refraction**: Refraction is the management of when rules fire so trivial loops are avoided. Without refraction, expert systems always would be caught in trivial loops. That is, as soon as a rule fired, it would keep firing on that same fact over and over again. In the real world, the stimulus that caused the firing eventually would disappear.
- **Clear**: Removes all facts and rules from working memory, basically resetting the system.

Note that Jess[56] is a version of CLIPS written in Java.

1.5. Opportunities for Using Expert Systems in Financial Reporting

What type of expert systems could be useful in the process of creating a financial report? Why would you even want to create an agent for financial reporting[57]?

---

[56] Jessrules, *The Java Expert System Shell*, https://jessrules.com/jess/docs/45/

[57] *Benefits Offered by an Expert System,* Retrieved July 24, 2016; http://xbrl.squarespace.com/journal/2016/5/30/understanding-the-benefits-offered-by-expert-systems.html

1.5.1.Starting simple; example of one type of expert system

The following is the architecture of an agent that could be helpful in the process of creating a financial report[58]. I believe that such an expert system would be classified as a global standard rational, deliberative, non-learning, goal-based intelligent software agent.
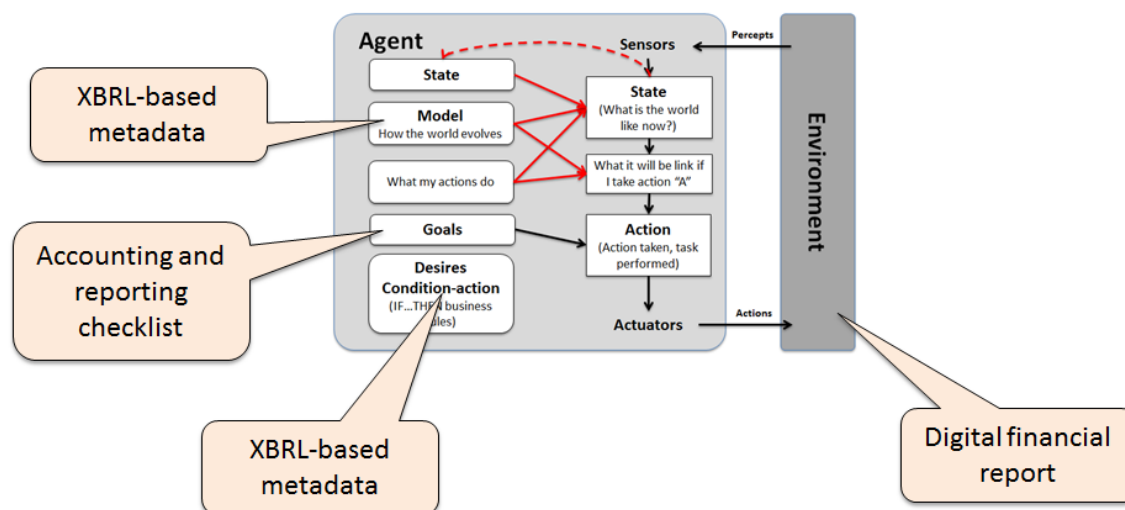
1.5.2.Automating accounting and reporting checklists

Most accountants are familiar with the disclosure checklist. They use that human readable checklist as a memory jogger in the process of creating financial reports. What if you made that checklist also readable by machines and what if financial reports were structured? Automating the disclosure checklist will be one of the first uses of intelligent agent software[59]. This will not be a batch process that you run when a financial report is complete; rather it will be an expert system intelligent software agent watching over you as you create the financial report.

1.5.3.World's first expert system for creating financial reports

I believe that a software developer and I have created what we can honestly call the world's first expert system for creating financial reports (as far as I am aware)[60].

But what is even more interesting is that what drives this expert system is a global standard XBRL-based general purpose business reporting expert system. The system is both in the form of a tool that is very approachable by business professionals and an API interface.

While the software application is admittedly rudimentary, it does successfully prove the concept of using an expert system in the process of creating a financial report.

---

[58] *Automating Accounting and Reporting Checklists,* Retrieved July 24, 2016;
http://xbrl.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html
[59] *Automating Accounting and Reporting Checklists,* Retrieved July 24, 2016;
http://xbrl.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html
[60] *World's First Expert System for Creating Financial Reports*,
http://xbrl.squarespace.com/journal/2017/4/27/worlds-first-expert-system-for-creating-financial-reports.html

Please watch this YouTube channel that will be updated to help explain this expert system for creating digital financial reports:

https://www.youtube.com/channel/UCRIbipm3f0DaGPuLK51rvHA

1.6. *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*

One type of practical knowledge is know-how; how to accomplish something.

Creating a knowledge based system involves the transformation of machine-readable instructions in such a way as to explain to a machine how a system works and how to make a system work the way you want that system to work.

Then, brick-by-brick, much like building a house, business domain experts and software engineers can create tools that automate certain types of tasks in that process. Humans encode information, represent knowledge, and share meaning using machine-readable patterns, languages, and logic. That will be the way an increasing number of work tasks will be performed in the Digital Age of accounting, reporting, and auditing. The result will be more efficient processes.

The document *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*[61] explains how a software engineer and I created proof of concept to test the feasibility of such a system.

---

[61] *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*, http://pesseract.azurewebsites.net/PuttingTheExpertiseIntoKnowledgeBasedSystem.pdf