

1. Comprehensive Introduction to Intelligent Software Agents

This is a dangerous time for professional accountants. Technology is changing very rapidly. For example, terms like “Big Data” and “Deep Learning” and “Artificial Intelligence” are being thrown about by software vendors, some of which are little more than snake oil salesman; with these new technologies comes a significant amount of hype. On the other hand, many of these software vendors have very useful products what will transform financial reporting as it is practiced today. Knowledge is very useful when one is trying to differentiate the hype from real possibilities.

Setting the right expectations is another big challenge. This example will help you understand what I mean. There is a lot of good and bad information relating to driverless cars. Tesla has functionality in its cars that is similar to driverless car-type functionality. Imagine trying to set a goal for what you want such a car to achieve:

- **Goal A:** Create a car that can travel anywhere in the United States without a driver.
- **Goal B:** Create a car that can assist the driver to drive anywhere in the United States.
- **Goal C:** Create a car that can assist a driver when driving on an Interstate Highway.

What is the relative cost of achieving goals A, B, and C above? What is the relative probability of even being able to achieve goals A, B, and C without significant, expensive research and development?

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 technology¹. 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.

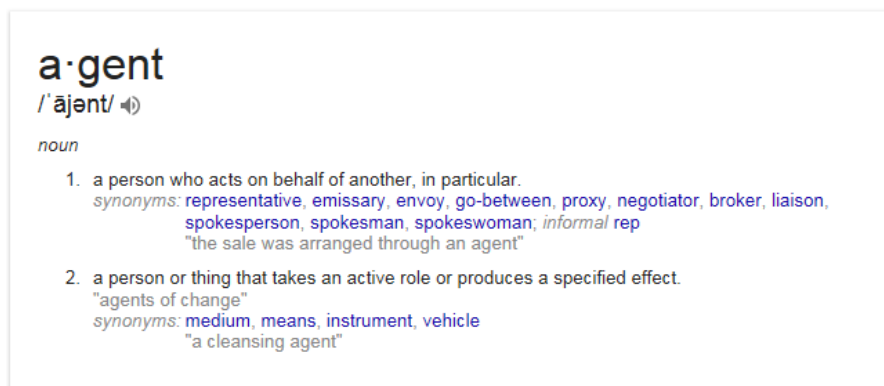
1.1. Deconstructing the Notion of an Intelligent Software Agent

To understand intelligent software agents one first needs to understand the notion of an agent. This section is dedicated to setting your perspective as to the notion of an agent. The section provides specific definitions, deconstructing the pieces so that we can subsequently put the pieces back together.

¹ Charles Hoffman and Liv Watson, *XBRL for Dummies*, page 145

1.1.1. Definition of an agent

Google defines agent² as “a person who acts on behalf of another” and “a person or thing that takes an active role or produces a specified effect”.



An agent performs specific tasks on behalf of another.

Important to the definition of agent is the law of agency³. Some principle, or the person making use of the agent, authorizes the agent to work on their behalf to perform an agreed upon specific task. The important key points are the *authority* to act and an *agreement* on the specific task or tasks that the agent is supposed to perform.

Business professionals generally understand the notion of an agent because they get introduced to this notion in their Business Law 101 class in college. This class introduces them to the notion of human agents. But what if the agent is not a human?

1.1.2. Software agents

In the case of computer science, an agent is a software program. Wikipedia defines software agent⁴ (paraphrasing) as “a computer program that acts for a user or other program in a relationship of agency to perform some action”.

Software agents are sometimes also referred to as “bots” from the term robot.

1.1.3. Intelligent agents

An intelligent agent⁵ is an abstract notion that links the *real world* agent and the notion of agency with an *implementation* of that functionality within software. An intelligent agent is the abstract functionality of a system similar to a computer program; it is not the computer software program itself.

Intelligent agent, as we are using it, is an idea related to artificial intelligence. An intelligent agent is an autonomous entity which observes its environment through sensors and acts upon that environment using actuators in the pursuit of some goal. Intelligent is a software design philosophy.

² Google search, *Agent definition*, retrieved August 14, 2016, <https://www.google.com/search?q=agent+definition>

³ Wikipedia, *Law of agency*, retrieved August 14, 2016, https://en.wikipedia.org/wiki/Law_of_agency

⁴ Wikipedia, *Software agent*, retrieved August 14, 2016, https://en.wikipedia.org/wiki/Software_agent

⁵ Wikipedia, *Intelligent agent*, retrieved August 14, 2016, https://en.wikipedia.org/wiki/Intelligent_agent

1.1.4. Intelligent software agents

The main difference between a software agent and an ordinary program is that a software agent is autonomous; that is, it must operate without direct intervention of humans or others.

Intelligent software agents are engineered using specific principles⁶.

1.1.5. Notion of an intelligent software agent which we will use

An agent is an entity capable of sensing the state of its environment and acting upon it based on a set of specified rules.

The consultancy firm McKinsey predicts that intelligent software agent use will increase in business⁷. Use of intelligent software is expected to be the norm in law firms⁸ by the year 2020.

1.2. Understanding what Intelligent Software Agents Do

This section helps you understand what intelligent software agents do and how they do it.

1.2.1. Basic example of an intelligent software agent

A basic example of an intelligent software agent is a basic thermostat^{9,10}. <http://xbrlsite.azurewebsites.net/2016/Library/ConceptualOverviewOfDigitalFinancialReporting.pdf>

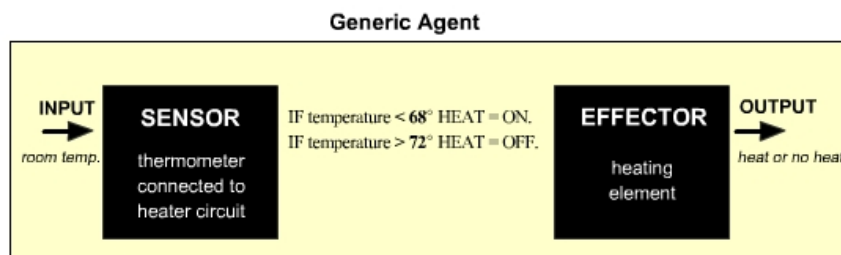


Fig. 1: Thermostat agent

Software reads the current room temperature using a sensor and uses a set of rules to turn the heat on or turn the heat off to impact the future room temperature in the room or environment in which the sensor is located.

⁶ Frances M.T. Brazier, Catholijn M. Jonker, Jan Treur
Vrije, *Principles of Component-Based Design of Intelligent Agents*,
<http://www.few.vu.nl/~wai/Papers/DKE02.princ.pdf>

⁷ McKinsey, *Artificial Intelligence Meets the C-Suite*, <http://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/artificial-intelligence-meets-the-c-suite>

⁸ Managing Partner, *Artificial intelligence to be 'the norm' in law firms by 2020*,
<http://www.managingpartner.com/news/business-strategy/artificial-intelligence-be-%E2%80%9898-norm%E2%80%9999-law-firms-2020>

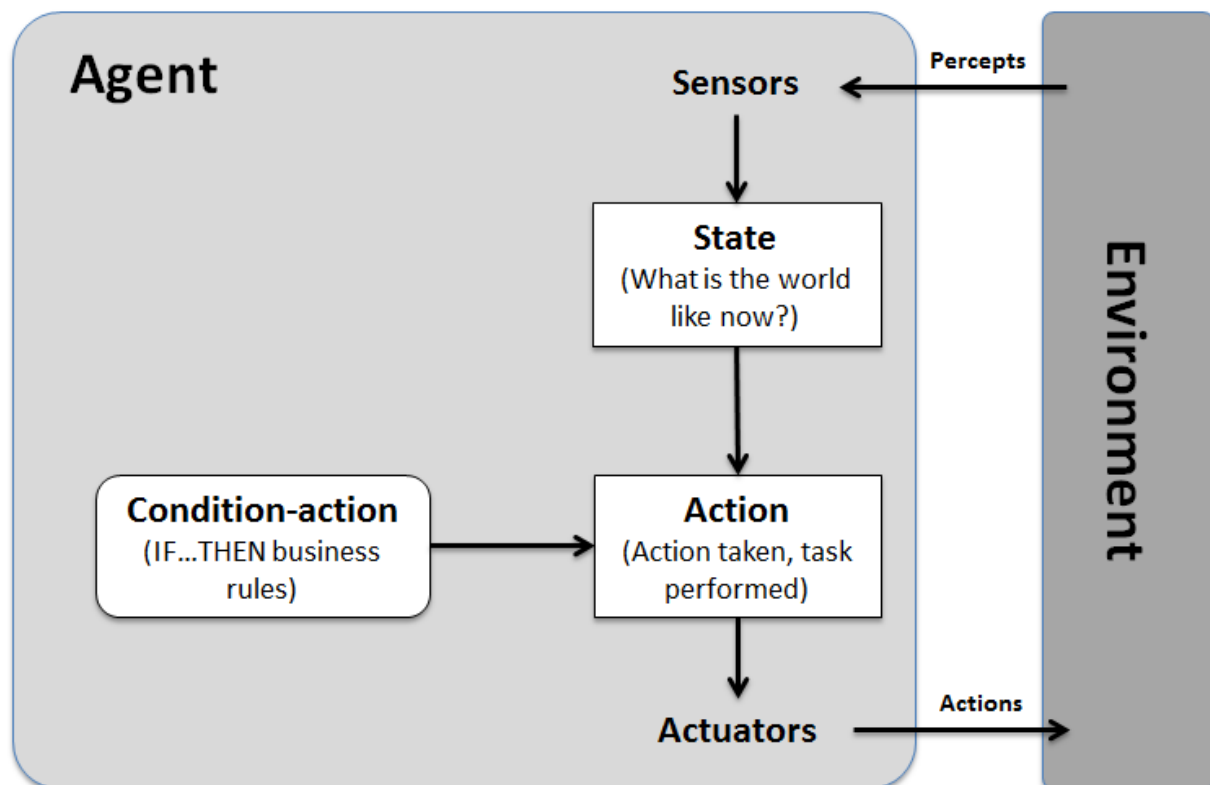
⁹ Consortium of Cognitive Science Instruction, *Introduction to Intelligent Agents*, Retrieved July 24, 2016;
http://www.mind.ilstu.edu/curriculum/ants_nasa/intelligent_agents.php

¹⁰ Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (Third edition)*, page 31,
<http://aima.cs.berkeley.edu/>

1.2.2. Basic model of an intelligent software agent

If you look at the basic intelligent software agent above and then you look at lots of other tasks that such intelligent software agents do you see patterns. Intelligent software agents leverage these patterns. Intelligent software agents¹¹ are computer code written in a specific way. An **agent** is an entity capable of **sensing** the **state** of its **environment** and **acting** upon it based on a **set of specified rules**. Remember that an agent performs specific tasks on behalf of another in order to achieve some agreed upon goal established by the principal that employs the agent and the agent which will act in behalf of the principle.

In the case of software, an agent is a software program. Consider that definition of an agent and look at the graphic below to get an idea of how intelligent agent software works:



An intelligent agent is software that assists people and acts on their behalf. Intelligent agents work by allowing people to:

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

¹¹ *Introduction to Intelligent Software Agents for Business Professionals*, http://xbrlsite.azurewebsites.net/2016/Library/02_IntroducingIntelligentAgents.pdf

- learn from you and even make recommendations to you.

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

An example of one major shift is provided by what professional accountants call the "disclosure checklist". Accountants creating financial reports often use accounting and reporting checklists or "disclosure checklists" as memory joggers to help them get the reports right¹². These memory joggers were created to be read by humans and can be a couple hundred pages. What if a financial report was structured, such as an XBRL-based public company financial report that must be submitted to the U.S. Securities and Exchange Commission. What if these human-readable memory joggers could be made machine-readable. And what if an intelligent software agent could be created to automate the manual task of checking a financial report to make sure that report was mechanically correct.

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

1.2.4. Important terminology related to intelligent software agents

The following is a summary of important terminology¹³ that is used to discuss intelligent software agents.

- **Agent:** An **agent** is an entity capable of **sensing** the **state** of its **environment** and **acting** upon it based on a **set of specified rules**. Remember that an agent performs specific tasks on behalf of another in order to achieve some agreed upon goal established by the principal that employs the agent and the agent which will act in behalf of the principle.
- **Environment:** environment in which the agent operates; description of the state of affairs that change over time as real world situations do.
- **Sensing capabilities:** capability of the agent to understand its environment; determines the sort of data the agent is capable of receiving as input.
- **Percept:** A percept refers to the agent's perceptual inputs at any given moment.
- **Percept sequence:** The percept sequence represents the complete sequence of percepts the agent has sensed or perceived during his lifetime.
- **State:** current conditions of the environment. (Or, a projected conditions of the environment based on actions taken.

¹² *Automating Accounting and Reporting Checklists*,
<http://xbri.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html>

¹³ Consortium of Cognitive Science Instruction, *Introduction to Intelligent Agents, Taxonomy of agents*, Retrieved July 24, 2016; http://www.mind.ilstu.edu/curriculum/ants_nasa/intelligent_agents.php

- **Actions:** change in the environment brought about by the agent, requiring the agent to update its model of the world, which in turn may cause the agent to change its immediate intention.
- **Condition-action rules:** A formal and implementable expression of some business user requirement. Includes definitions of terms, structural assertions, action assertions, and derivations.
- **Desires:** overall policies or goals of the agent.
- **Action selection architecture:** an agent decides what to do next by consulting both its internal state, the state of the world, and its current goal; then the agent uses decision making procedures to select an action.

1.2.5. Agents Working Together: Multi-agent Systems

Note that one agent must be able to communicate with other agents and exchange information between agents. Below is terminology related to the communication between agents¹⁴.

- **Multi-agent system:** When an agent coexists in an environment with other agents, perhaps collaborating or competing with them, the system is considered a multi-agent system.
- **Coalition:** A coalition is any subset of agents in the environment.
- **Strategy:** A strategy is a function that receives the current state of the environment and outputs the action to be executed by a coalition.
- **Blackboard:** A blackboard structure is a communication form that consists of a shared resource divided into different areas of the environment where agents can read or write any significant information for their actions.
- **Coordination:** Coordination is essential in multi-agent system because it provides coherency to the system behavior and contributes to achieving team or coalition goals.
- **Cooperation:** Cooperation is necessary as a result of complementary skills and the interdependency present among agent actions and the inevitability of satisfying some success criteria.
- **Competition:** Another possible model is that in which the agents are self-motivated or self-interested agents because each agent has its own goals and might enter into competition with the other agents in the system to achieve these goals. In this sense, competition might refer to accomplishing or distributing certain tasks.
- **Negotiation:** Negotiation might be seen as the process of identifying interactions based on communication and reasoning regarding the state and intentions of other agents.

When one intelligent software agent communicates with another intelligent software agent some technical syntax and semantics must be employed¹⁵. The XBRL global standard can be used for such communication and collaboration.

¹⁴ C# - Applying AI to a Multi-Agent 'Mini-Basketball' Game, Retrieved July 24, 2016; <https://msdn.microsoft.com/en-us/magazine/mt736456.aspx>

1.3. *Building your Understanding of Intelligent Software Agents*

We will build your understanding of intelligent software agents piece-by-piece in this section.

1.3.1. *Artificial intelligence*

Artificial intelligence¹⁶ is a branch of computer science. There are many good descriptions of artificial intelligence¹⁷. 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.

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

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.

1.3.2. *Expert systems*

Expert systems¹⁸ is a branch of artificial intelligence. The following is a definition of an expert system:

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

1.3.3. *Business rules*

Key to employing artificial intelligence and therefore making an expert system or intelligent software agent¹⁹ work is business rules²⁰ of the domain being put into machine-readable form.

¹⁵ Hanh Tran & Thaovy Tran, *Intelligent Agent, Agent Communication Languages*, http://groups.engin.umd.umich.edu/CIS/course.des/cis479/projects/agent/Intelligent_agent.html#dep6

¹⁶ *Introduction to Artificial Intelligence Terminology*, <http://xbri.squarespace.com/journal/2016/7/21/introduction-to-artificial-intelligence-terminology.html>

¹⁷ AlanTuring.net, *What is Artificial Intelligence?*, http://www.alanturing.net/turing_archive/pages/reference%20articles/What%20is%20AI.html

¹⁸ *Understanding the Components of an Expert System*, <http://xbri.squarespace.com/journal/2016/5/24/understanding-the-components-of-an-expert-system.html>

¹⁹ Wikipedia, *Intelligent Agent*, retrieved July 24, 2016; https://en.wikipedia.org/wiki/Intelligent_agent

1.3.4.Important distinctions between intelligent software agents

There are two important distinctions that you should keep in the back of your mind when we talk about intelligent software agents:

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

Every intelligent software agent that we are interested in is generally both a rational and autonomous agent.

1.3.5.Categories of intelligent software agent functionality

The functionality of an intelligent software agent can be classified into groups, or categories:

- **Reactive agent:** A reactive agent is capable of maintaining an ongoing interaction with the environment and responding in a timely fashion to changes that occur in it.
- **Pro-active agent:** A pro-active agent is capable of taking the initiative; not driven solely by events, but capable of generating goals and acting rationally to achieve them.
- **Deliberative agent:** A deliberative agent symbolically represents knowledge and makes use of mental notions such as beliefs, intentions, desires, choices and so on. (This is implemented using a belief-desire-intension model.)
- **Hybrid agent:** A hybrid agent is one that mixes some of all the different architectures.

Every intelligent software agent falls into one of those four categories.

1.3.6.Categories of intelligent software agent sophistication

Intelligent software agents can be grouped as to the level of sophistication offered by the agent:

- **Generic agent:** An agent is anything that perceives an environment through sensors and acts or reacts upon the environment through effectors.
- **Simple reflex agent:** A simple reflex agent looks up what it should do from a list of rules in response to its perception to the environment.
- **Model-based reflex agent:** A model-based reflex agent is the same thing as a simple reflex agent except that a model-based reflex agent has a model of how the environment evolves.

²⁰ *Comprehensive Introduction to Business Rules for Professional Accountants*, <http://xbrl.squarespace.com/journal/2016/8/3/comprehensive-introduction-to-business-rules-for-professiona.html>

- **Goal-based agent:** A goal-based agent has a goal or set of goals that it actively pursues in accordance with an agenda (so this type of agent is proactive, not just reactive). A goal-based agent has a representation of the current state of the environment and how that environment works. The agent pursues policies or goals that may not be immediately attainable. And so, goal-based agents do not live merely in the moment. These agents consider different scenarios before acting on their environments, to see which action will probably attain a goal. This consideration of different scenarios is called search and planning.
- **Utility-based agent:** A utility-based agent is a more sophisticated type of goal-based agent that also rates each possible scenario to see how well it achieves certain criteria with regard to production of the good outcome, therefore it is more adaptive. A utility measure is applied to the different possible actions that can be performed in the environment. The utility-based agent will rate each scenario to see how well it achieves certain criteria with regard to the production of a good outcome. Things like the probability of success, the resources needed to execute the scenario, the importance of the goal to be achieved, the time it will take, might all be factored in to the utility function calculations.

Every intelligent software agent falls into one of those groups in terms of sophistication of functionality offered.

1.3.7. Learning versus non-learning intelligent software agents

Every intelligent software agent can be either a learning or non-learning agent:

- **Learning agent:** A learning agent is one that requires some training to perform well, adapts its current behavior based on previous experiences and evolves over time.
- **Non-learning agent:** A non-learning agent is one that doesn't evolve or relate to past experiences and is hard-coded and independent of its programming.

A learning agent is significantly more sophisticated in terms of functionality, but the tradeoff is that they are also significantly more complex to create.

1.4. Intelligent software agents assisting humans

Artificial intelligence is the automation of activities that we associate with human thinking and activities such as decision making, problem solving, learning and so on²¹.

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

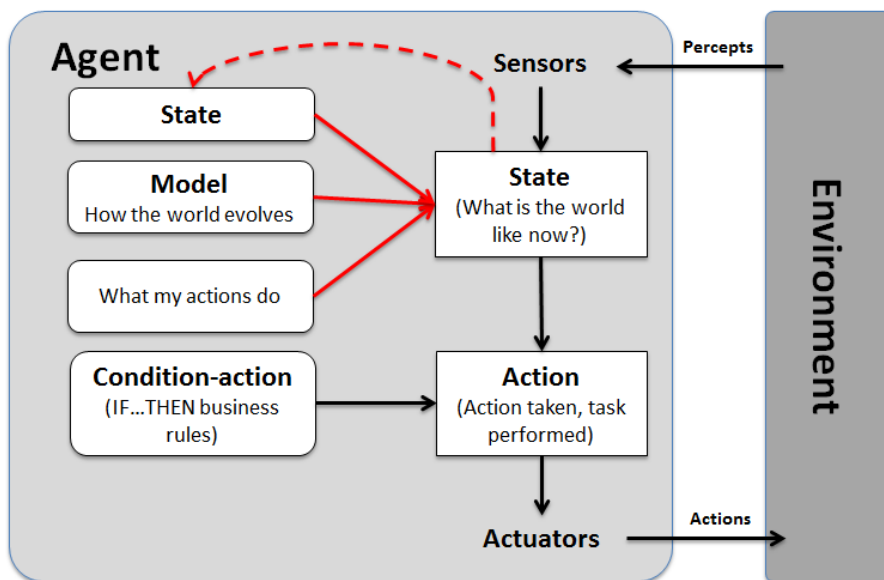
- delegate work that they could have done to the agent software,
- perform repetitive tasks,
- remember things you forgot,

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

- intelligently find, filter and summarize complex information,
- customize information to your preferences,
- learn from you and even make recommendations to you.

An agent is an entity capable of **sensing** the **state** of its **environment** and **acting** upon it based on a set of specified **rules**. An agent performs specific tasks on behalf of another. In the case of software, an agent is a software program.

The main difference between a software agent and an ordinary program is that a software agent is autonomous; that is, it must operate without direct intervention of humans or others. There are many different types of intelligent software agents²².



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

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

1.4.1. Getting the thick layer of metadata

No one disputes that you need a thick layer of metadata to get computers to perform useful work for you²³. The more metadata you have, the more a computer can do. What can be in dispute, particularly if you are not aware of the differences between the approaches, is the different ways you can get that thick layer of metadata. Here are the two ways:

²² *Introduction to Intelligent Agents for Business Professionals (DRAFT)*, http://xbrl.site.azurewebsites.net/2016/Library/02_IntroducingIntelligentAgents.pdf

²³ *Understanding the Need for a Framework and Theory*, <http://xbrl.squarespace.com/journal/2015/9/20/understanding-the-need-for-a-framework-and-theory.html>

- **Have the computer figure out what the metadata is:** This approach uses artificial intelligence, machine learning, deep learning, neural networks, and other high-tech approaches²⁴ 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 so that the metadata becomes available.

Both approaches can be appropriate in the right situation. If the tolerance for error is low, like in a financial report, machine learning and neural networks generally don't work. What works really well is when you tell the computer what the metadata is first which serves as training data and then machines can be employed to leverage that training data to derive more metadata which human domain experts sort through and make right.

So again, both approaches can be useful in the right situation. Understanding the pros and cons of each approach helps you make good choices of which approach to employ. Technology has its limits which computer science professionals sometimes forget to mention to business professionals.

1.4.2.Benefits offered by expert systems

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

- **Automation:** elimination of routine, boring, repetitive, mundane, mechanical tasks that can be automated
- **Consistency:** computers are good at performing repetitive, mechanical tasks without variation whereas humans are not; computers do not make mistakes and are good at repeating exactly the same thing each time
- **Diligence and tenacity:** computers excel at paying attention to detail; they never get bored or overwhelmed and they are always available and will keep doing their job until the task is complete with the same attention to detail
- **Reduced down-time:** computer based expert systems are tireless and do not get distracted
- **Availability:** such computer based expert systems are always available simultaneously in multiple places at one time; you get quick response times and can replace absent or scarce experts
- **Training:** the best practices of the best practitioners can be available to those that are new to and learning about a domain of knowledge
- **Longevity and persistence:** computer based expert systems do not change jobs or retire so knowledge gathered by an organization can remain within that organization
- **Productivity:** computer based expert systems are cheaper than hiring experts and costs can be reduced at the same time that quality increases resulting in increased productivity

²⁴ Understanding 'Deep Learning' and 'Neural Networks',
<http://xbrl.squarespace.com/journal/2015/9/17/understanding-deep-learning-and-neural-networks.html>

- **Multiple opinions:** Systems can integrate the view of multiple experts within a single system and choose between the preferred view of multiple expert opinions in the same system
- **Objectivity:** computers apply the same inductive and deductive logic consistently; emotion and personal preferences can be eliminated where they should be eliminated

Critical to understanding the sorts of tasks that expert systems will be capable of performing and should not or will never be able to perform takes the understanding of a domain professional. While computer based expert systems can effectively automate some work, this does not imply that these systems will automate all work or replace humans. They simply won't because they cannot. Computers are dumb beasts. There is a difference between subjectivity and objectivity; there is a difference between a mechanical task and a task requiring professional judgement. Professional accountants need to understand the difference²⁵.

1.4.3. Machines helping humans

Machines are good at some things. Humans are good at other things. Understanding the difference is important for a lot of reasons. Machines can be made to perform certain tasks effectively and efficiently.

McKinsey published an article which predicted²⁶:

"Many of the jobs that had once seemed the sole province of humans—including those of pathologists, petroleum geologists, and law clerks—are now being performed by computers."

Understanding the true capabilities of computer software and humans will help you understand what skills you will want to have in order to make sure you can add value to your organization. Those same skills will help you use intelligent software agents appropriately.

1.5. Details of Categories of Intelligent Software Agent Sophistication

In the section *Categories of intelligent software agent sophistication* we summarized the different categories of intelligent software agents and the level of sophistication offered. In this section we want to provide more details on the these different categories.

1.5.1. Generic agents

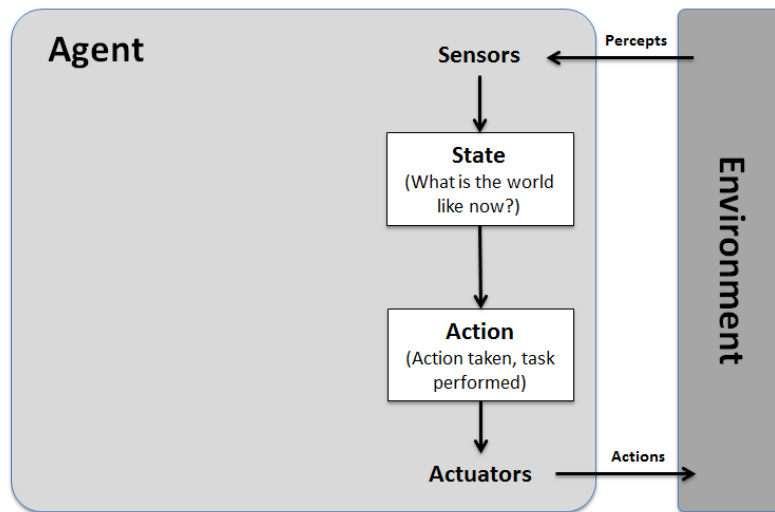
A generic agent²⁷ is anything that perceives an environment through sensors and acts or reacts upon the environment through effectors.

²⁵ Knowledge Engineering Basics for Professional Accountants, http://www.xbrlsite.com/2016/Prototype/00_KnowledgeEngineeringBasicsForAccountingProfessionals.pdf

²⁶ KcKinsey Quarterly 2014, *Artificial intelligence meets the C-suite*, <http://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/artificial-intelligence-meets-the-c-suite>

²⁷ Consortium of Cognitive Science Instruction, *Introduction to Intelligent Agents, Generic agent*, Retrieved July 24, 2016; http://www.mind.ilstu.edu/curriculum/ants_nasa/intelligent_agents.php

Generic Agent



The following is the basic control loop of a generic agent:

While true

1. observe the world (**environment**);
2. update internal world model (**state**);
3. deliberate about what intention to achieve (**goal**);
4. use means/ends reasoning to get a plan for the intention (**condition-action rules**)
5. execute the plan (**action**)

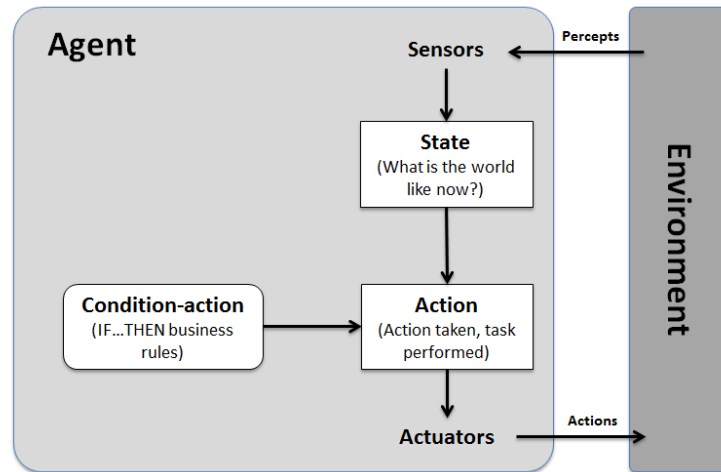
End while

1.5.2. Simple reflex agents

A simple reflex agent²⁸ looks up what it should do from a list of rules in response to its perception to the environment.

²⁸ Wikipedia, *Intelligent Agents, Simple Reflex Agents*, retrieved August 14, 2016, https://en.wikipedia.org/wiki/Intelligent_agent#Simple_reflex_agents

Simple Reflex Agent



A simple reflex agent looks up what it should do from a **list of rules** in response to its perception to the environment. This is the algorithm of simple reflex agent²⁹:

Function Simple-Reflex-Agent (percept) returns action

- **persistent:** *rules*, a set of condition-action rules
- *state* <<<< Interpret-Input (*percept*)
- *rule* <<<< (Rule-Match (*state*, *rules*))
- *action* <<<< *rule*.Action
- **return** *action*

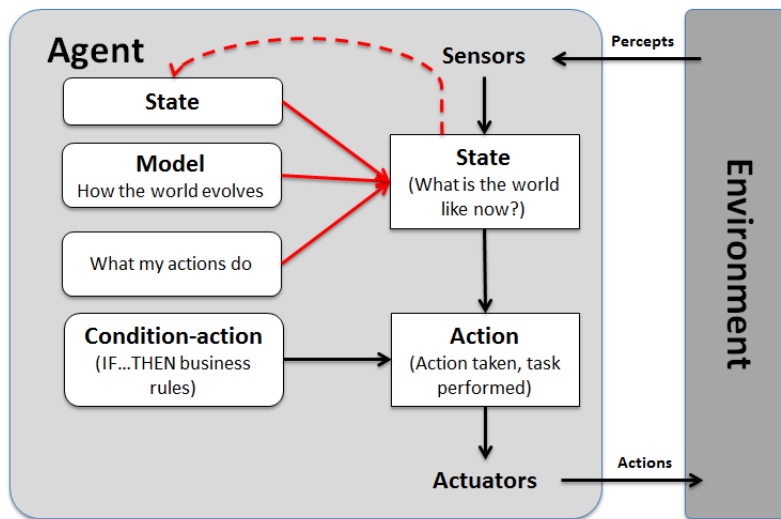
1.5.3. Model-based reflex agents

A model-based reflex agent³⁰ is the same thing as a simple reflex agent except that a model-based reflex agent has a model of how the environment evolves.

²⁹ Hugo Larochelle, *Intelligence Artificielle*, Retrieved July 24, 2016;
<https://www.youtube.com/watch?v=TUHAVbaBLIq>

³⁰ Wikipedia, *Intelligent Agents, Model-based Reflex Agents*, retrieved August 14, 2016,
https://en.wikipedia.org/wiki/Intelligent_agent#Model-based_reflex_agents

Model-based Reflex Agent



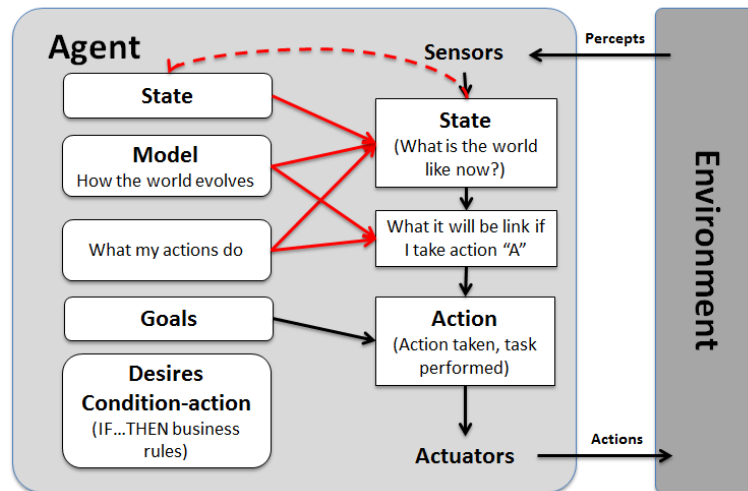
A model-based reflex agent is essentially the same as a simple reflex agent except that a model-based reflex agent adds the additional functionality of a model of the environment in which the agent operates.

1.5.4. Goal-based agents

A goal-based agent³¹ has a goal or set of goals that it actively pursues in accordance with an agenda (so this type of agent is proactive, not just reactive). A goal-based agent has a representation of the current state of the environment and how that environment works. The agent pursues policies or goals that may not be immediately attainable. And so, goal-based agents do not live merely in the moment. These agents consider different scenarios before acting on their environments, to see which action will probably attain a goal. This consideration of different scenarios is called search and planning.

³¹ Wikipedia, *Intelligent Agents, Goal-based Agents*, retrieved August 14, 2016, https://en.wikipedia.org/wiki/Intelligent_agent#Goal-based_agents

Goal-based Agent



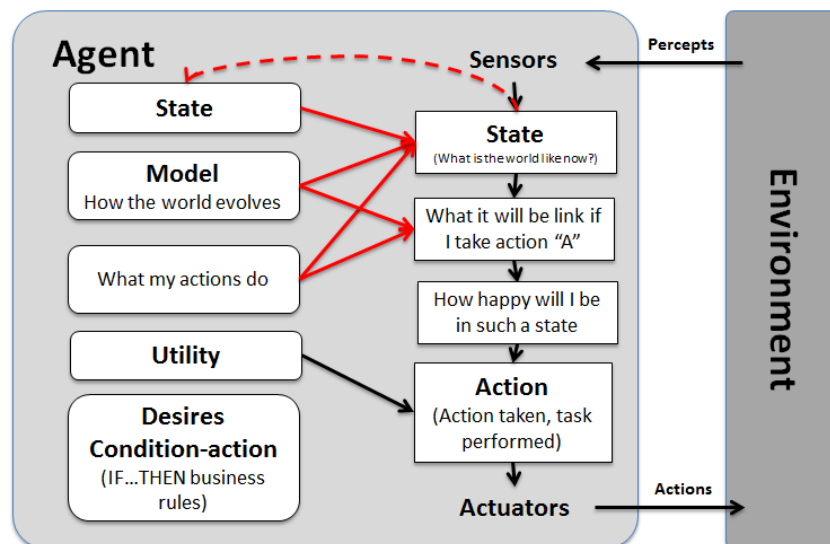
A goal-based reflex agent is essentially the same as a model-based reflex agent except that the goal-based reflex agent adds the additional functionality of wondering what the environment will be like if a specific action is taken, evaluating if that state is desirable or undesirable given specific goals, and the agent's desires given that state.

1.5.5. Utility-based agents

A utility-based agent³² is a more sophisticated type of goal-based agent that also rates each possible scenario to see how well it achieves certain criteria with regard to production of the good outcome, therefore it is more adaptive. A utility measure is applied to the different possible actions that can be performed in the environment. The utility-based agent will rate each scenario to see how well it achieves certain criteria with regard to the production of a good outcome. Things like the probability of success, the resources needed to execute the scenario, the importance of the goal to be achieved, the time it will take, might all be factored in to the utility function calculations.

³² Wikipedia, *Intelligent Agents, Utility-based Agents*, retrieved August 14, 2016, https://en.wikipedia.org/wiki/Intelligent_agent#Utility-based_agents

Utility-based Agent



A utility-based agent is the same as a goal-based reflex agent but adds additional functionality of wondering what the environment will be like if a specific action is taken, evaluating if that state is desirable or undesirable given specific goals and desires given that state, and evaluating how happy the agent will be within that state. Such agents might even be allowed to change its goals.

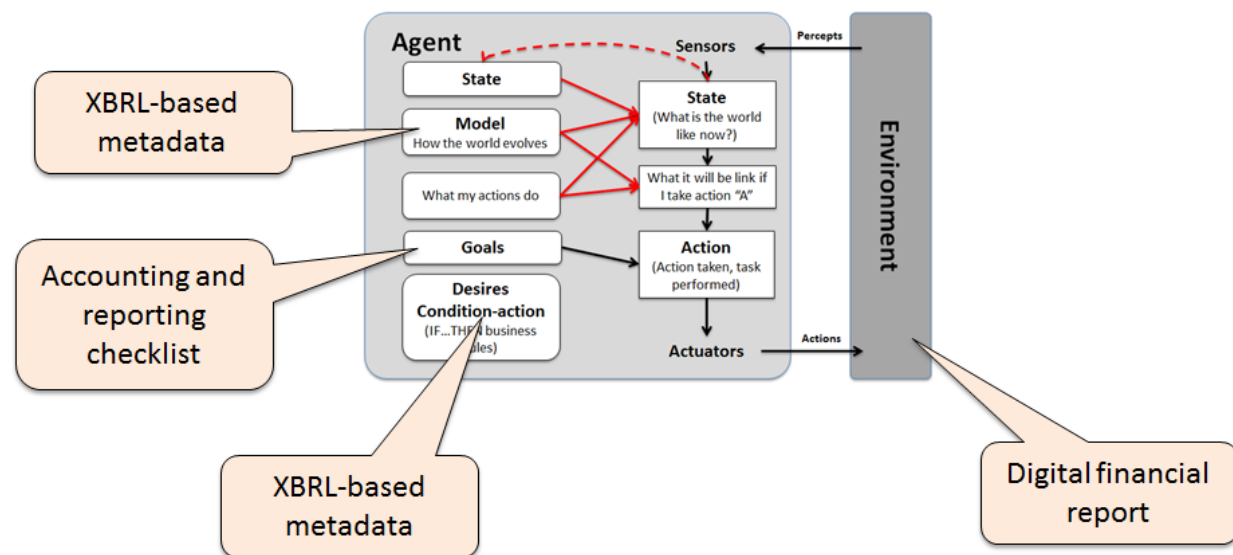
1.6. Using Intelligent Software Agents in Financial Reporting

So what is the best type of agent? That depends. What type of agents could be useful in the process of creating a financial report?

Why would you even want to create an agent for financial reporting³³?

1.6.1. Starting simple; example of one type of agent

The following is the architecture of an agent that could be helpful in the process of creating a financial report³⁴. I believe that such an agent would be classified as a global standard rational, deliberative, non-learning, goal-based agent.



1.6.2. Automating accounting and reporting checklists

1.7. Additional Resources for Getting Started

The following is helpful background material which will help you during your process of getting up to speed with model-based digital financial reporting including its core component XBRL. None of this material is required in order to fully use the material in this document. However, in order to understand the reasoning behind many decisions, to understand the XBRL technical syntax which lies behind this model, this information can be helpful.

The point of this document is to keep XBRL behind the scenes as much as possible. Most of this will be achieved by implementing software applications which absorbed

³³ *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>

³⁴ *Automating Accounting and Reporting Checklists*, Retrieved July 24, 2016; <http://xbrl.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html>

the complexity of the XBRL technical syntax, allowing the business user to achieve what they need to achieve safely, robustly, reliably, and consistently; never having to understand the details of XBRL.

That said, many people like to “dink” around with their cars in their garage. If you don’t know what you are doing, this can be dangerous. But, with good training, resources, or other help it can be useful to “get under the hood” so to speak. This is likewise true of XBRL.

1.7.1.Digital financial reporting and XBRL, the big picture

A good source for background information about XBRL-based digital reporting, what it is, what it does, what it means, problem it is trying to solve, how it is being used by others and other such background information read *XBRL for Dummies*. This is not a required step, but this step can be helpful if you want to get a complete picture of the landscape surrounding model-based digital financial reporting. The following URL points you to information on where to obtain this resource, details of what it contains, and other helpful information:

<http://xbrl.squarespace.com/xbrl-for-dummies/>

1.7.2.Hello world! example

If you don’t know what a hello world example is, you probably should not even bother with this. If you do know what a hello world example, here is a hello world example of XBRL:

<http://xbrl.squarespace.com/journal/2008/12/18/hello-world-xbrl-example.html>

Again, this level of understanding can be helpful, but it is optional.

1.7.3.An XBRL technical syntax primer

Again, while not necessary, an understanding of the basics of the XBRL technical syntax can be helpful. Chapter 4 of *XBRL for Dummies* (see above) has a concise, but more importantly correct, primer of the XBRL technical syntax. Weighing in at a mere 28 pages, for those who desire or need this level of detail this is a great place to start. The XBRL technical specification is like reading the owner’s manual. Even if you want to wade through that 158 page document, the primer will help you grasp the bigger picture framework into which to fit the details you will collect about XBRL.

1.7.4.Explore public company financial information

Ultimately, one of the primary reasons for taking the time and effort to structure financial information is so that the information can be extracted and used by automated machine-based processes. One very good example of this is the XBRL-based financial information reported by public companies to the U.S. Securities and Exchange Commission (SEC) and made available in their EDGAR system.

All that XBRL-based financial information is there and free to use.

The problem with that is that the information is made available in the raw XBRL technical syntax.

A software vendor, 28msec, has created a free publically available repository of public company financial information and provides access to machine-readable and human-readable information. That repository is called SECXBRL.info and is

available here: <http://app.secxbml.info/>. Information for the DOW 30 is available for free.

If you make your way to the list of the companies which make up the DOW 30, you can navigate through the information repository and explore. All the information came from XBRL-based digital financial reports.

ID	Archives	Profile	EntityName	CompanyType	SIC	SICDescription	SICGroup	Sector
http://www.sec.gov/CIK/0000058740	Link	SEC	3M CO	Corporation	3841	SURGICAL & MEDICAL INSTRUMENTS & APPARATUS	10	Measuring/Analyzing/Photographic/Control/Medical Instruments
http://www.sec.gov/CIK/000004952	Link	SEC	AMERICAN EXPRESS CO	Corporation	6199	FINANCE SERVICES	7	Nondepository Credit Institutions
http://www.sec.gov/CIK/0000732717	Link	SEC	AT&T INC.	Corporation	4813	TELEPHONE COMMUNICATIONS (NO RADIOTELEPHONE)	11	Communications
http://www.sec.gov/CIK/0000112627	Link	SEC	BOEING CO	Corporation	3721	AIRCRAFT	5	Transportation Equipment
http://www.sec.gov/CIK/0000116230	Link	SEC	CATERPILLAR INC	Corporation	3531	CONSTRUCTION MACHINERY & EQUIP	10	Industrial/Commercial Machinery/Computer Equipment
http://www.sec.gov/CIK/0000094110	Link	SEC	CHEVRON CORP	Corporation	2911	PETROLEUM REFINING	4	Petroleum Refining and Related Industries
http://www.sec.gov/CIK/000085877	Link	SEC	CISCO SYSTEMS, INC.	Corporation	3578	COMPUTER COMMUNICATIONS EQUIPMENT	3	Industrial/Commercial Machinery/Computer Equipment
http://www.sec.gov/CIK/0000211344	Link	SEC	COCA COLA CO	Corporation	2080	BEVERAGES	9	Food/Kindred Products
http://www.sec.gov/CIK/0000330554	Link	SEC	DUPONT E I DE NEMOURS & CO	Corporation	2820	PLASTIC MATERIAL, SYNTH RESIN/RUBBER, CELLULOS (NO GLASS)	6	Chemicals/Alled Products
http://www.sec.gov/CIK/0000314088	Link	SEC	EXXON MOBIL CORP	Corporation	2911	PETROLEUM REFINING	4	Petroleum Refining and Related Industries
http://www.sec.gov/CIK/000040545	Link	SEC	GENERAL ELECTRIC CO	Corporation	3500	ELECTRONIC & OTHER ELECTRICAL EQUIPMENT (NO COMPUTER EQUIP)	10	Electronic/Other Electrical Equipment/Components
http://www.sec.gov/CIK/000085862	Link	SEC	GOLDMAN SACHS GROUP INC	Corporation	6211	SECURITY BROKERS, DEALERS & FLOTATION COMPANIES	8	Securities and Commodities Brokers/Dealers/Exchanges/Services
http://www.sec.gov/CIK/0000354950	Link	SEC	HOME DEPOT INC	Corporation	5211	RETAIL-LUMBER & OTHER BUILDING MATERIALS DEALERS	6	Building Materials, Hardware, Garden Supply
http://www.sec.gov/CIK/0000050863	Link	SEC	INTEL CORP	Corporation	3674	SEMICONDUCTORS & RELATED DEVICES	10	Electronic/Other Electrical Equipment/Components
http://www.sec.gov/CIK/0000261143	Link	SEC	INTERNATIONAL BUSINESS MACHINES CORP	Corporation	3570	COMPUTER & OFFICE EQUIPMENT	3	Industrial/Commercial Machinery/Computer Equipment

Other sources are available, but what 28msec provides is one of the best organized financial information repositories. If you are really ambitious, you may want to explore the API reference and create something using Excel.

1.7.5.Putting the pieces together

Trying to understand calculus without knowing algebra is pointless. Often when you need to learn something it is important to breakdown what you want to learn into steps. You learn what you need to in step one, then you go to step two. You learn step two, then go to step three. And so on. Trying to skip steps causes frustration and other problems.

Learning about XBRL-based digital financial reporting requires you to progress through a series of steps. There are no short cuts. Master of model-based digital financial reporting will take an investment in time and effort, particularly today when software is not at the maturity level that it needs to be. If you don't want to put in this time and effort, wait for software to mature.

If you do want to make this investment, this material is laid out the way it is precisely to help you through these necessary steps. They will not turn you into a master, but they will set you on the path to mastery. Mastery takes even more time and effort. Again, there are no short cuts.