Pacioli – Concept Paper

Platform for interacting with XBRL-based digital financial reports logically

By

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

November 16, 2020 (DRAFT)

> "The difficulty lies not so much in developing new ideas as in escaping from old ones." *John Maynard Keynes*[1]
>
> "If I had asked people what they wanted, they would have said faster horses." Henry *Ford*[2]

Executive summary:

- Today the processes of financial accounting, financial reporting, the audit of financial reports, and financial analysis are processes that contain many automated and manual tasks and processes.
- Increasing complexity and volume of information is putting stress on current accounting, reporting, auditing, and analysis processes.
- New tools and capabilities are being provided such as structured information, artificial intelligence, digital distributed ledgers.
- These new tools and capabilities are changing the dynamics of what tasks and processes of financial accounting, reporting, auditing, and analysis must be performed manually and what might be automatable.
- Pacioli is a logic programming based, enterprise ready, scalable "platform" or "toolkit" or "infrastructure" for effectively and reliably interacting with machine readable digital financial reports such as XBRL-based digital financial reports.
- The first user application to be created using Pacioli will be a tool for auditors.

[1] Good Reads, https://www.goodreads.com/quotes/5099424-the-difficulty-lies-not-so-much-in-developing-new-ideas

[2] Good Reads, https://www.goodreads.com/quotes/15297-if-i-had-asked-people-what-they-wanted-they-would

Copyright (full and complete release of copyright)

Automation is about removing friction, driving costs down, speeding processes up, and improving efficiency and productivity. Automation is about improving processes in order to deliver goods and services that are better for less cost in a timely manner.

Today the processes of financial accounting, financial reporting, the audit of financial reports, and financial analysis are processes that contain many automated and manual tasks and processes. These tasks and processes tend to be accomplished using highly skilled, highly trained accountants and tend to, in most cases, yield acceptable results. But the cost of achieving those results is high.

Information is increasing in both complexity and volume, stressing existing processes and technologies. Most finance processes tend to be fed by many dissimilar systems making information integration complex particularly as the volume of automation increases.

Older technologies are making it increasingly difficult to keep up with today's fast paced information exchange. New technologies such as structured information, artificial intelligence, digital distributed ledgers and others offer significant and compelling opportunities to make accounting, reporting, auditing, and analysis tasks and processes more efficient and effective. A Forbes article[3] points out that organizations are already using artificial intelligence to create more intelligent products, create more intelligent services, and improve internal business processes.

But how exactly is "automation" actually achieved? What is necessary to replace existing processes is ongoing repeatable processes that are reliable because the processes can be controlled thus achieving verifiable high-quality. This results in effective automation. Saying this another way, if a process cannot be controlled then the process simply cannot repeatedly and reliably output high-quality. If output is not high-quality, automation cannot possibly be effective.

So, control of a process in necessary in order for the process to be effective. How to you control a process? You control a process using rules. Manual processes are controlled by rule that are read by humans. Automated processes are controlled by rules that are readable by both machines (i.e. to execute the process) and humans (i.e. to make sure the rules are right).

Who creates these machine-rules that are used to control processes that yield effective automation? Accountants must create these rules because the rules tend to be accounting

[3] Bernard Marr, Forbes, *3 Important Ways Artificial Intelligence Will Transform Your Business And Turbocharge Success*, https://www.forbes.com/sites/bernardmarr/2020/08/03/3-important-ways-artificial-intelligence-will-transform-your-business-and-turbocharge-success/

oriented. Technical rules tend to relate to rule syntax and such technical rules can be hidden from business professionals. What is left is the business logic and accounting rules that are used to control information and control process workflow. As such, the creation of machine readable rules must be "self-service". Business professionals must be empowered to create, adjust, maintain, and otherwise manage the rules that are used to control and therefor effectively automate processes.

With the implementation of self-service rules, business professionals can control their processes. Rules provide control; control leads to high-quality; high-quality enables effective, reliable automation.

Financial accounting related tasks and processes including reporting, auditing, and analysis are particularly automatable because they adhere to mathematical models such as the double entry accounting model and the accounting equation. These and other details are provided in the document *Essence of Accounting*[4].

There are many additional details related to effectively automating processes. Those details are discussed in the document *Computational Professional Services*[5].

This paper provides a conceptual overview and articulates the value proposition of Pacioli[6] which is a scalable, enterprise platform for working with digital financial reports effectively. Pacioli can be seen as a "logical toolkit" and a "new software infrastructure" which leverages artificial intelligence problem solving capabilities to effectively automate accounting, reporting, auditing, and analysis tasks and processes.

Primary Problem-Solving Paradigms

First, in order to evaluate the value proposition of Pacioli, one needs to understand that there are three primary problem-solving logic paradigms[7]:

1. **Knowledge Graphs** (i.e. the W3C semantic web stack; RDF, N3, OWL, SHACL, SPARQL, RDF triple stores)
2. **Graph Databases** (i.e. Neo4j and other labeled property graphs, Graph Query Language or GQL, graph databases)

[4] Charles Hoffman, CPA, *Essence of Accounting*, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf
[5] Charles Hoffman, CPA, *Computational Professional Services*, http://xbrlsite.azurewebsites.net/2020/library/ComputationalProfessionalServices.pdf
[6] Logical Contracts, asdf, http://logicalcontracts.com/xbrl-and-sbrm-reporting/
[7] Primary Problem Solving Logic Paradigms, http://xbrl.squarespace.com/journal/2020/9/15/primary-problem-solving-logic-paradigms.html

3. **Logic Programming** (i.e. Prolog, SQL, relational databases)

Fundamentally, to get a computer to do work you in essence write "code" or create machine-readable information that such a computer can interact with in the form of what amounts to a series of "if…then" statements that get the computer to perform the work you desire to be performed (automated).

This "code" can be broken down into helpful "buckets" to help with the process of creating the code, maintaining the code, and managing to tasks and processes the computer performs. Again, these details are critically important to grasp. If they are not grasped, ineffective or inefficient results could be the result. These details are provided in the document Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements[8].

Pacioli Platform

Pacioli is a cloud-based "platform" or "toolkit" or "infrastructure for working effectively with XBRL-based digital financial reports. The first application we will build using Pacioli is a tool for external financial reporting managers, internal auditors, and external auditors to determine if an XBRL-based financial report is a properly functioning logically (consistent, complete, precise) and help those and other business professionals understand if information being conveyed by such reports is what they intended.

Pacioli is a platform that can scale and otherwise meet the needs of the enterprise. Pacioli is not designed to "meet a regulator mandate"; it is a tool for effectively automating tasks and processes involved with accounting, reporting, auditing, and analysis.

While these capabilities can also be leveraged for general business reporting; the focus of Pacioli is financial reporting. Simply by stripping out the double entry accounting model, the accounting equation, and any other financial reporting scheme specific logic; Pacioli can be configured for general business reporting.

While a "tool for auditors" might not be the most sexy software application imaginable; a tool for auditors must be the first tool created because if such a tool for auditors does not exist, how can an accountant understand that an XBRL-based financial report that they have created is properly created, how can an internal auditor sign off on such reports and be right, or how can

[8] Charles Hoffman, CPA, *Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements*, http://xbrlsite.azurewebsites.net/2019/Library/SpecialTheoryOfSemanticCommunicationOfFinancialInformation.pdf

an external third-party auditor be able to communicate issues that they detect with a reporting economic entity and for an auditor an and an external financial reporting manager have a meaningful discussion as to whether an XBRL-based financial report that they have created is or is not a properly functioning digital financial report that conveys the meaning that was intended to be conveyed by the reporting economic entity and properly interpreted by financial analysts that use information from such a report?

SWI-Prolog

The Pacioli platform is built on top of the SWI-Prolog platform[9]. SWI-Prolog is a mature, freely available, open source, comprehensive commercial Prolog environment. "Out of the box", SWI-Prolog provides key features and libraries including:

- Basic logic engine already built-in which covers Part 1 of the ISO Prolog standard, the de-facto Edinburgh Prolog standard, and important parts of Quintus and SICStus Prolog and good compatibility with other version of Prolog including Ciao, YAP and GNU-Prolog. Although the aim is to maintain compatibility with standard ISO Prolog wherever possible, SWI-Prolog deliberately deviates from the ISO standard to accommodate additional functionality and synchronize with modern logic programming languages.
- Extensive meta programming capabilities, handy for formula evaluation
- Flexible parser, facilitating friendlier syntaxes
- Built-in search and matching
- Very efficient C-based engine, with RAM usage an order of magnitude under other currently existing alternatives
- Vast open source API libraries, including XML processing, JSON processing, web services, etc.
- Mature implementation, consolidating over 3 decades of optimization and refinement in the fields of Logic Programming and Cognitive AI.

Most of the existing applications for processing XBRL don't even leverage basic XBRL processors; rather software vendors typically tend to choose to create all necessary functionality "from scratch". Creating a logic engine is a non-trivial task.

Rather than repeat this mistake, Pacioli leverages one of the three primary problem-solving logic paradigms.

[9] SWI-Prolog, https://www.swi-prolog.org/

SBRM

No one really argues that a thick layer of machine-readable metadata is necessary in order to automate processes and perform work. The more machine-readable metadata, the more work can be performed. The question is: who will create that thick layer of metadata.

OMG's forthcoming Standard Business Report Model (SBRM)[10] provides a global standard logical conceptualization of a business report. The SBRM logical conceptualization is based on the *Logical Theory Describing Financial Report*[11].

The *Logical Theory Describing Financial Report* and Standard Business Report Model were created by reverse-engineering XBRL-based financial reports submitted to the U.S. Securities and Exchange Commission (SEC) using the US GAAP financial reporting scheme and the IFRS financial reporting scheme. The European Single Electronic Format (ESEF)[12] effectively follows the same or extremely similar architecture as the SEC creating, in essence, a de facto standard approach to creating XBRL-based digital financial reports.

Pacioli leverages a proven, reliable, standards-based, best practice based method[13] for creating XBRL-based financial reports. Pacioli enables enterprise-class quality not simply the steps to "meet a regulatory mandate". Critical to understanding the value of Pacioli is having an understanding of the method[14] used by Pacioli to control XBRL-based report quality.

XBRL Format Plus Other Syntax

Currently, Pacioli imports XBRL-based information, CSV-based information using an SBRM format. Ultimately, other technical syntax formats will be supported including Excel, JSON, RDF/OWL/SHACL, Cypher (Neo4j), the forthcoming Graph Query Language (GQL)[15] format which is expected to become an ISO standard, SQL, and other relevant formats used by enterprises.

[10] OMG, Standard Business Report Model (SBRM), https://www.omg.org/intro/SBRM.pdf
[11] Charles Hoffman, *Logical Theory Describing Financial Report*, http://xbrl.squarespace.com/logical-theory-financial-rep/
[12] ESMA, *European Single Electronic Format*, https://www.esma.europa.eu/sections/european-single-electronic-format
[13] Proven, Reliable, Standards-based Best Practice Method for Automating, http://www.xbrlsite.com/2020/Theory/MethodTwoPager.pdf
[14] Understanding Method (Abridged), http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingMethod_Abridged.pdf
[15] Graph Query Language Standard (GQL), https://www.gqlstandards.org/

Current Pacioli Implementation

The current implementation of Pacioli is a lean cloud server assembly accessible either by a plain web form or using about 15 interactive SWI Prolog "notebooks"[16] that is used for communication, learning, training, testing, debugging, and other engineering tasks necessary to construct the underlying infrastructure for Pacioli. Pacioli is not currently a product or even resemble what the product will ultimately look like.

Testing of Pacioli[17] has been rigorous, methodical, detailed, and extensive.

Technical

Just to give you some perspective on the present implementation, Pacioli is currently 5,000 lines of code, comments included, all Prolog except 100 lines of embedded Javascript, implementing everything you see online and behind it, running on a single Docker container on a tiny (1Gb ram + 20Gb disk) Amazon Web Services (AWS) instance. As the UI grows there will probably be more Javascript. Pacioli uses also a number of standard Javascript open source libraries, and the SWI-Prolog built-in libraries (XML parsing, CSV parsing, HTTP support, and HTML support). The current implementation was created in about 500 hours of developer time by a software engineer with 30+ years of Prolog expertise.

Why so little code and so little time to develop a robust set of functionality? Three reasons. First, Prolog is a high-level language so less code is necessary to implement functionality as contrast to a lower level functional programming language. Second, an engine + model + rules approach is used. Third, only the subset of the XBRL syntax was is used has been implemented as contrast to implementing 100% of XBRL but then not using 80% of the XBRL technical syntax that you implemented. This approach has significant positive impact on code maintenance and business rule maintenance.

Note that the software engineer creating Pacioli had ZERO experience with XBRL before creating what he created.

Pacioli has an XBRL processor and XBRL Formula processor that is partially implemented but already processes 98+% of all the XBRL-based financial reports submitted to the SEC using US GAAP and IFRS. Inline XBRL is not yet supported, but the raw XBRL instance and XBRL taxonomy schemas including linkbases are processed. Full support for all the XBRL Formulas used by the SBRM is currently working. This API is 100% Prolog and will be exposed as a web

[16] Logical Contracts, XBRL-based Financial Reporting, http://logicalcontracts.com/xbrl-and-sbrm-reporting/
[17] Testing of Pacioli, http://xbrlsite.azurewebsites.net/2020/master/Pacioli.html

service. Further, Pacioli provides missing capabilities not provided by XBRL Formula[18] that are necessary to fully process XBRL-based financial reports and determine if such reports are properly functioning logical systems[19] that are complete, consistent, and precise.

The current toolkit started with the conversion of XBRL instances and their full DTS into a compact Prolog representation that leverages the SBRM. The following examples help you get a sense for Pacioli:

- Notebooks: http://demo.logicalcontracts.com/example/PROOF.swinb
- Importer: http://demo.logicalcontracts.com/reportImporter
- Altering XBRL: http://demo.logicalcontracts.com/example/AlteringXBRL.swinb
- Fundamental accounting concept validation: (older example, this is deprecated) http://demo.logicalcontracts.com/example/XBRLlab.swinb
- Other: http://xbrl.squarespace.com/journal/2020/9/15/pacioli.html

Formula evaluation, including variable binding, is done by the straight execution of a Prolog goal (actually, a reasoning derivation) generated from the formula; this follows naturally from accumulated know-how in meta interpreters and partial evaluation

XPath and Prolog search combined navigate the XBRL documents, which Prolog relations accumulate an equivalent but more amenable representation, ready to fuel reasoning upon it.

After this first step, now upper layers (model patterns, more complex rules) can be added.

Pacioli Financial Report Web Service

Pacioli will be a web service used by other software vendors creating software applications related to financial reporting that wish to "buy" robust functionality rather than "build" the functionality themselves. For example, consider the Luca[20] software application for creating XBRL-based reports. Ultimately, Luca will call a web service provided by Pacioli for verifying that an XBRL-based report is correctly created and for information used to review the report. Similarly, other software vendors can use the Pacioli web service to verify that their XBRL-based reports are correctly created, expert system functionality to drive accounting and reporting tasks and processes, Excel interfaces for creating or reviewing financial reports, analysis tools that query and extract information from XBRL-based financial reports.

[18] Specific Deficiencies in Capabilities of Existing XBRL Formula Processors, http://xbrl.squarespace.com/journal/2016/9/26/specific-deficiencies-in-capabilities-of-existing-xbrl-formu.html
[19] Understanding the Financial Report Logical System, https://www.youtube.com/playlist?list=PLqMZRUzQ64B7EWamzDP-WaYbS_W0RL9nt
[20] Luca, http://xbrl.squarespace.com/journal/2020/9/15/luca.html

The Pacioli web service provides a complete framework which includes the logic processing capabilities of Prolog, the logic expressed as machine-readable rules using the XBRL syntax or perhaps other syntax, the financial report model, the logical theory which describes how everything interacts, and a method that provides control over automation because of the high-quality resulting from a robust set of rules.