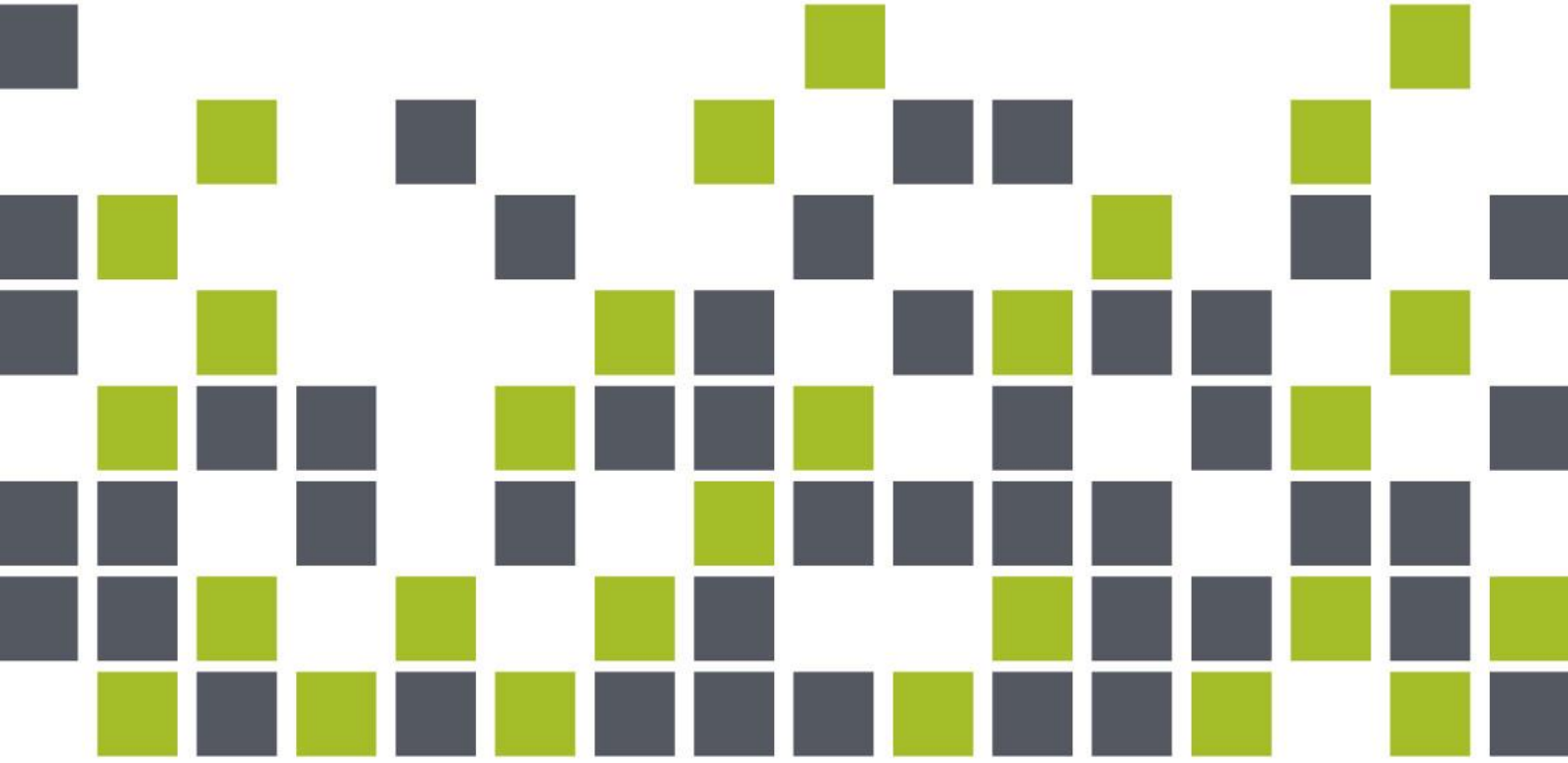
IDIOM
PURE DECISIONING

Excel and IDIOM

A Comparison

Whitepaper October 2017



☎ +64 9 6308950 | @ idiomsales@idiomsoftware.com | idiomsoftware.com
✉ 2-2, 93 Dominion Road, Mount Eden, Auckland 1024

Excel and IDIOM

Executive Summary 4
Background 5
Glossary 5
Fundamentals 7
Product Introduction 7
Excel 7
IDIOM Decision Manager 8
Summary of Product Positioning 11
The Building Blocks 11
Excel 11
IDIOM Decision Manager 13
Formulas 14
Excel 14
IDIOM Decision Manager 15
General Comparisons 15
Processing Models 16
Execution Model 16
Excel 16
IDIOM Decision Manager 17
Versioning 17
Polymorphism 18
Testing 18
Scalability 19
Calculation Speed 19
Data 20
Transactions versus On-Board Data 20
Data Structure 21
Data Types 22
Security of Intellectual Property 22

Excel 22
IDIOM Decision Manager 23
Summary 23

Mark Norton | CEO and Founder | Idiom Limited

☎ +64 9 630 8950 | +64 21 434 669 | Australia free call 1800 049 004
✉ 2-2, 93 Dominion Road, Mount Eden, Auckland 1024 | PO Box 60101, Titirangi, Auckland 0642, New Zealand
@ mark.norton@idiomsoftware.com | idiomsoftware.com | Skype Mark.Norton

EXECUTIVE SUMMARY

Excel and IDIOM Decision Manager share a similar concept of Formulas (see Glossary), which are developed en masse to collectively build-out potentially large-scale 'calculation models'. In each case, the calculation models can include many primary calculations supported by a large number of sub calculations.

From this point of overlap, the products diverge into contrasting areas – Excel as a personal modelling tool with extensive control over input and output formats; and IDIOM as a tool for injecting substantial multi-use calculation models into front-line Transactions (see Glossary).

Excel is a personal productivity tool that seamlessly blends on-board data with its Formulas in one self-contained calculation model (viz. a spreadsheet). The on-board data includes configuration and reference data, as well as the critical calculation 'context data'. The context data is the data that varies for each execution of the calculation model, and is both the subject of, and the purpose of, the model (e.g. an insurance policy, claim, loan, etc).

Excel's very basic data representation (rows/columns), and its limited control over calculation process flows, together limit the range of use cases that can be managed in a single spreadsheet, so that there are limits on the range of calculation events that can be supported in a single spreadsheet.

IDIOM uses multi-dimensional data schemas to specify its context data; and its calculation engine simultaneously supports many calculation models and process flows across multiple calculation events. Together these features allow one IDIOM calculation engine to process an unlimited number of context data entities for any number of discrete calculation events. That is, IDIOM explicitly supports a multi-purpose/multi use calculation paradigm.

The IT concept of a Transaction is the basis for most commercial processing systems. A Transaction makes a state change to its context data by applying all relevant calculation models to it. The purpose of the IDIOM Decision Manager is to independently build and test these calculation models, and to then inject them into production system Transactions as a single executable that supports multiple calculation models per transaction event. This omnibus executable is the IDIOM Calculation Engine (see Glossary).

The appeal of Excel is that it is very accessible, and it can rapidly evolve calculation models of substantial scale, including all relevant input/output formats. However, these models are ultimately limited in calculation scope; and they cannot be deployed in a transactional context unless refactored and re-written in a more appropriate computer language.

If the ultimate purpose of a calculation model is to manage real-world data in a transactional context, then an early transition from the very agile but ultimately limited Excel approach to an IDIOM Decision Manager multi-purpose/multi use approach should be considered: To increase run-time scalability, the calculation scope of the models, and the transparency and auditability of the modelling process; To allow automated testing and version management; To protect calculation IP; And to deploy calculation IP 'without fingerprints' into any production application, for use at any scale.

Excel and IDIOM

BACKGROUND

IDIOM is regularly involved in converting Excel spreadsheets into IDIOM Decision Models, as a means to productionising the Excel based algorithms. This indicates a degree of equivalence in the two products.

This equivalence can be confusing and we are sometimes asked when one or other should be used. This Whitepaper attempts to shed light on this question.

The generalised comments about Excel in this document reflect what IDIOM generally sees as 'normal' Excel usage in the third-party spreadsheets that we are asked to convert. In some cases, an Excel 'power-user' could extend the capabilities of the standard spreadsheet beyond this norm. Where we are aware of these extended capabilities, we have noted them in footnotes. However, it is our view that the presence of extended capabilities such as Power Pivot and Power Query do not materially change our assessments in this document.

IDIOM is very respectful of Excel as a product and sometimes competitor! In our view, Excel is unparalleled as an initial and personal prototyping tool for the inception of industrial scale calculations. If those calculations are intended to remain in the laboratory, to be initiated manually when and as required, then Excel is an appropriate choice.

IDIOM, on the other hand, is focused on managing its calculations inside production system Transactions at an industrial scale. If the intended purpose of the calculations is for use inside business transactions in a production setting, then IDIOM should be a preferred choice.

Therefore, the end-use of the calculations is an important factor in assessing which tool to use. This assessment should be made early in the calculation development life-cycle, because an extended Excel based calculation development cycle is likely to constrain the calculation design in multiple and sometimes subtle ways – in particular with regards to normalisation of calculations and sub-calculations, which is partially a consequence of Excel's limited two-dimensional data representation.

Glossary

In this document:

Algorithm[1] means the combination of algebra[2], first order logic[3], and business rules[4] that together specify precisely how the answer to a specific class of problem is determined. For the purpose of this document, we assert that both Excel and IDIOM can define more or less equivalent algorithms.

Author means the person writing an Excel Spreadsheet, or building an IDIOM Decision Model, as the context requires. The skillsets are not dissimilar.

Calculation means either a) an Algorithm, or b) a Formula, or c) the result of an Algorithm or Formula as the context requires. See also Decision.

Calculation Engine is an IDIOM term that means the IDIOM supplied executable, which can ingest, manage, and execute any number of Decision Models, each of which could equate to an Excel spreadsheet. The scale and utility of the IDIOM calculation engine can extend to the entire population of calculations for most organisations.

Calculation Model means slightly different things in Excel vs IDIOM Decision Manager. In Excel, the calculation model more or less equates to a single spreadsheet instance. In IDIOM, the calculation model is represented by the Calculation Engine, which can embrace an entire community of Excel scale calculation models.

Context Data means the subject data that is supplied to a Calculation Engine to provide the real-world subject matter for any given invocation of the calculation. The context data provides the purpose of the calculation, and is the proximate target for the calculation outcomes (aka 'decisions'). Context data excludes reference data, configuration data, and other passive data that are not subject to a state change controlled by the calculation.

Decision (an IDIOM proper-term) means a calculation outcome that is persisted because of its inherent value to the business. IDIOM formally defines a Decision as "A single definitive datum that is derived by applying business knowledge to relevant data for the purpose of positively supporting or directing the activity of the business."

Decision Model (also an IDIOM proper-term) means "an ordered assembly of decisions that creates new and proprietary information to further the mission of the business".

Formula means the specification of the Algorithm for a Calculation

IDIOM Decision Manager is the proprietary IDIOM software that enables business Subject Matter Experts [SMEs] to build and test the Formulas that implement business policy. The IDIOM Decision Manager builds and manages Formulas, Decisions, and Decision Models.

Logic means the assembly of operator 'primitives' that are required for data acquisition and manipulation, predicate logic, and numerical processing, and which are assembled into

---

[1] https://en.wikipedia.org/wiki/Algorithm

[2] https://en.wikipedia.org/wiki/Algebra

[3] https://en.wikipedia.org/wiki/First-order_logic

[4] https://en.wikipedia.org/wiki/Business_rule

Formulas. These primitive building blocks are universal, and are more or less equivalent building blocks common to both Excel and IDIOM Decision Manager. Examples include 'if/then/else', all, not, any, plus, minus, multiply, divide, get, put, etc.

Normalisation is used with its 'information management' meaning, wherein complex forms are reduced to simpler forms by removing redundancy in the definition thereof, while at the same time increasing re-use of shared elements. Something can be described as fully normalised when it requires the smallest number of elements to fully describe the subject matter, which by definition infers zero redundancy in the definition. The objective of normalisation is to simplify complex definitions, increase their utility and agility (through reuse and reduced time and cost to change), and reduce potential update anomalies.

Transaction means the activity that is initiated by an event that changes the state of a business entity, and which when complete leaves all systems compliant with all business policies that are relevant for that entity in that state.

FUNDAMENTALS

Product Introduction

Excel

From Wikipedia[5]:

"Microsoft Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. It has been a very widely applied spreadsheet for these platforms, especially since version 5 in 1993, and it has replaced Lotus 1-2-3 as the industry standard for spreadsheets.

Microsoft Excel has the basic features of all spreadsheets using a grid of cells arranged in numbered rows and letter-named columns to organize data manipulations like arithmetic operations. It has a battery of supplied functions to answer statistical, engineering and financial needs. In addition, it can display data as line graphs, histograms and charts, and with a very limited three-dimensional graphical display. It allows sectioning of data to view its dependencies on various factors for different perspectives (using pivot tables and the scenario manager). It has a programming aspect, Visual Basic for Applications, allowing the user to employ a wide variety of numerical methods, for example, for solving differential equations of mathematical physics, and then reporting the results back to the spreadsheet. It also has a variety of interactive features allowing user interfaces that can completely hide the spreadsheet from the user, so the spreadsheet presents itself as a so-called application, or decision support system (DSS), via a custom-designed user interface, for example, a stock analyzer, or in general, as a design tool that asks the user questions and provides answers

[5] https://en.wikipedia.org/wiki/Microsoft_Excel

and reports. In a more elaborate realization, an Excel application can automatically poll external databases and measuring instruments using an update schedule, analyze the results, make a Word report or PowerPoint slide show, and e-mail these presentations on a regular basis to a list of participants. Excel was not designed to be used as a database."

IDIOM Decision Manager

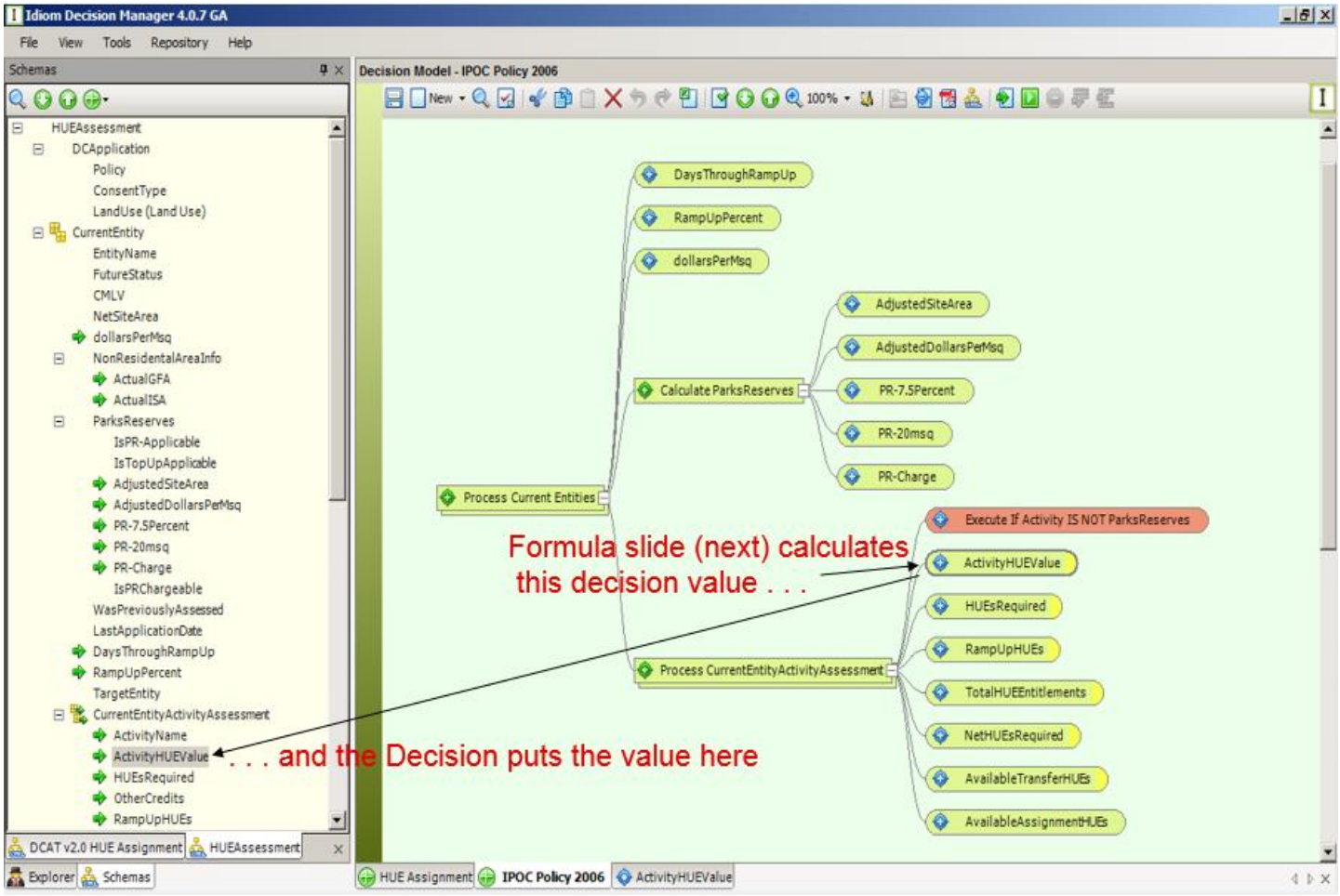
A contrasting summary for IDIOM Decision Manager[6] is as follows.

- ✓ IDIOM Decision Manager is a tool for graphically modelling, testing, and deploying business decision-making logic – without programming!
- ✓ A tool for the policy maker, not the programmer.
- ✓ IDIOM automates complex decision-making at the enterprise level, deployed as industrial strength stand-alone software components.
- ✓ In day-to-day practice, it is used by SMEs, or analysts working closely with them.
- ✓ Together they model the business policies in terms of both data and decisions (see Decision Model below) before moving on to define the underlying logic that binds them together (see Formula below).
- ✓ The decision logic and data are usually modelled together in a single combined process of analysis and definition.
- ✓ The data model and the decision model share the same 'context awareness', with current-context and context boundaries visually highlighted within the development palettes.
- ✓ Testing of the models is available at all times within the development palettes themselves; extensive regression testing (incl. 'model answer' differencing) is available in the on-board 'Test Executive' (not shown).
- ✓ Deployment as 100% generated software components is fully automated and 'without fingerprints'.
- ✓ Example below is a small but real model drawn from a city council implementation of policy that calculates financial contributions to be paid by property developers.
- ✓ The problem domain is decomposed using a 'mind mapping' approach until we reach the atomic units that we call decisions (rounded boxes).
- ✓ This 'decision model' and the adjacent data model (left hand panel below) are demonstrably aligned and integrated through shared context – validating and strengthening both.
- ✓ The data model defines the entity at rest; the decision model defines the valid state transitions. Together they completely define the entity life-cycle and required business policy.
- ✓ The atomic 'decisions' provide an easy entry point for specification of the underlying rule details via the Formulas (see next).
- ✓ The underlying rule details are easily captured using a 'Lego like' drag-and-drop approach that is 'more fun than playing golf' according to the CEO of one of our largest customers – there is no scripting or coding required to build Formulas.
- ✓ The rules can be tested immediately within the IDIOM Decision Manager palettes.

[6] http://www.idiomsoftware.com/Pages/Products/IDIOMDecisionManager.aspx

- ✓ When finished, IDIOM Decision Manager generates computer source code (C# or Java) with a single button click, to be called by any application at run-time using any of a wide variety of supplied interfaces and wrappers (in-line, dll, web service, queue service, and many more).
- ✓ And at the same time, it generates the models into business readable PDF documentation.

Figure 1 IDIOM Decision Manager - Decision palette showing a Decision Model and Schema



Key points of difference

- ✓ IDIOM's decision models do for calculations what data models do for data – a powerful abstraction that makes the underlying complexity visible and manageable.
- ✓ The models allow data transformations and more traditional business rules to be intermingled. Business rules acting alone are severely limited in their ability to fully resolve complex commercial problems – invariably, in-line data transformations are necessary to facilitate the calculate/adjudicate/act behaviour of business rules.
- ✓ Context is continuously displayed and actively managed.
- ✓ Decision models that incorporate both data and rules behaviour enable a further critical capability that is unique to IDIOM Decision Manager – the models can be fully tested using real-world cases directly in the builder palettes. No external technology or

application support is required to empirically prove the completeness, consistency, and correctness of the models.

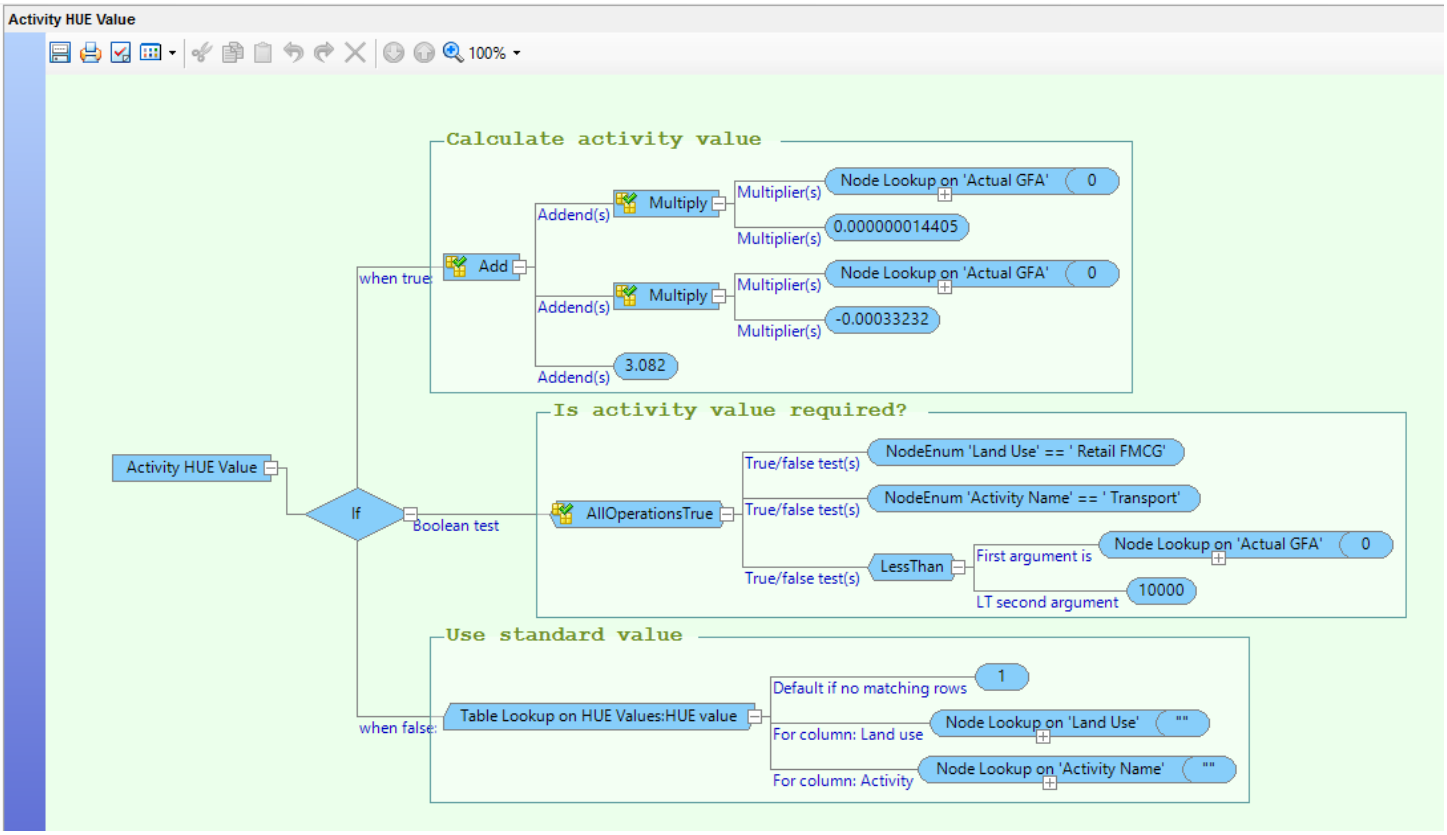
Key points of innovation

- ✓ Fundamental redesign of the traditional SDLC by fully separating the development and automation of business policy (deciding) from development of the system's activities that support it (doing).
- ✓ Use of IDIOM is effective in spawning a 'Calculations Development Life Cycle' that is managed independently of and alongside the traditional Systems Development Life Cycle.

Key value propositions

- ✓ 100% alignment of systems based decision making with business policy, because the business owners have hands-on custody and control of the policy definitions actually used by the system.
- ✓ Increased agility with reduced business risk through business modelling and empirical testing of policy definitions prior to automated generation and implementation.
- ✓ Significant reduction in the business cost of developing and implementing automated business policy.
- ✓ Further reduction in software development time, cost, and risk through reduced system complexity, fewer moving parts, & clear separation of concerns.

Figure 2 IDIOM Decision Manager - showing the Formula Palette



Further benefits

- ✓ Full auditability of policy changes, and visibility of policy implementation through the graphical models and logical English PDF generation.
- ✓ Decision model artefacts can be traced to/from 'sources of truth' in underlying business policy documents (Word, Excel) using the IDIOM Tracker for requirements traceability.
- ✓ The IDIOM Decision Manager Workbench is available to experiment with and further develop policy independently of the underlying systems and of the SDLC.
- ✓ Automated, robust, industrial strength deployment on any scale that can be supported by the 'host application' and its underlying platform.
- ✓ Simple injection into legacy systems leading to eventual legacy replacement

Summary of Product Positioning

IDIOM summarises Excel as follows:

A scalable, self-contained, and robust personal tool for modelling algorithms with primary focus on algebraic algorithms, and with a secondary focus on the input/output, presentation, and other data manipulation techniques as required to support the algebra.

IDIOM summarises IDIOM Decision Manager as follows:

IDIOM Decision Manager is an SME driven development tool for modelling, testing, and documenting business logic (including business rules, data manipulation, and numerical operations) independently of the operational systems that will apply that logic; and then delivering self-contained black-box executables to those systems for local execution of the business logic – that is, a calculations engine.

There are areas where IDIOM does not compete with Excel. In a reflection of its provenance in prototyping and financial modelling, Excel has evolved a substantial applied mathematics capability that extends to Operations Research. IDIOM does not offer a comparable capability in this area. Similarly, with on-board reporting and graphical presentation of results.

This is because the primary objective of IDIOM Decision Manager is the deployment of black-box executables into core systems as a proxy for the business owners of the algorithms contained therein. Operations Research and graphical reporting capabilities are not normally required in standard commercial computer Transactions, and are therefore not a focus of IDIOM.

The Building Blocks

Excel

Excel is comprised of cells, which are addressed by their row/column location within worksheets.

In normal use, there is minimal meta data describing the cell, limited more or less to formatting. There are options for naming and categorizing, and adding other metadata to cells especially with the XML tables that were added in Excel 2007, but it is unusual for these options to be used.

All cells are equal in the eyes of consuming cells. In particular, a cell may be a Formula, or a data value, including reference data and/or context data. Because the spreadsheet usually has minimal meta data describing its cells, a Formula cannot address other cells by their meta data; it must explicitly address any cell or cells that it requires a value from[7].

This means that a Formula which needs to repeatedly process a thousand other cells must itself exist a thousand times, so that it can address each of the thousand cells individually[8]. This explosion of Formulas on each row of data substantially de-normalises the calculation model in Excel, which creates the potential for update anomalies when the redundant copies are not all updated to the exact same value simultaneously. In fact, the data in Excel can also suffer from this form of duplication, so that both data and Formula cell values are often repeated across rows.

The effects of the resulting de-normalised structure are aggravated by the fact that the multiple copies of the same value, whether it be data or Formula, are not managed by any form of meta-dictionary. There is no way for the spreadsheet to record the fact that replicated cell values are all logically the same value; this knowledge must be carried externally to the spreadsheet and/or be inferred by an Author.

In summary, the following are fundamental constraints that, while allowing substantial agility, limit Excel's utility as a core calculations development technology:

- Two-dimensional row/column structure limits the complexity of data that can be addressed, thereby limiting the Excel problem domain to simple data structures[9]
- Using the cell as the common and fundamental building block for both data and Formulas, and allowing them to be interchangeable
- Lack of an integrated meta-level dictionary – there is no 'map' or meta level documentation for an Excel spreadsheet
- Excel will only reliably function on a document up to a certain size; approx. 100MB[10].

Because Excel is inherently not transactional in nature, it is usually necessary to manually rebuild spreadsheet logic in toto in order to deploy it into multi-user commercial applications. To rebuild a spreadsheet requires end-to-end technical analysis of the spreadsheet (including linked spreadsheets), which can be technically difficult and error prone, and which in IDIOM's experience also regularly uncovers pre-existing errors in the spreadsheet itself. The larger and more complex the spreadsheet, the more likely this is.

---

[7] This is not true of Excel XML tables, which have structured references referring to columns, however many users are unaware of this functionality, and most legacy spreadsheets do not utilize it.

[8] Excel XML tables have only one formula for each column, but are not much used.

[9] The addition of Power Pivot and Power Query to Excel 2010/2013 have gone a long way to addressing this issue, but they require Excel users to learn new techniques for data loading and manipulation. IDIOM has not yet seen this in a commercially supplied spreadsheet.

[10] Power Pivot can work with much larger documents, however in doing so it requires large amounts of memory. Power Query can work with even larger documents, but tends to be very slow in doing so.

IDIOM Decision Manager

The IDIOM Decision Manager takes a different approach, and the building blocks are fundamentally different. As the following schema diagram shows, the data (Schema), Decision Models and Decisions, Formulas, and reference data Tables are all first order artefacts in an IDIOM Repository. The very existence of this Repository schema in the public domain is a substantial difference when compared with Excel – this is a meta-model, a 'map' of the superstructure for all IDIOM calculation models.

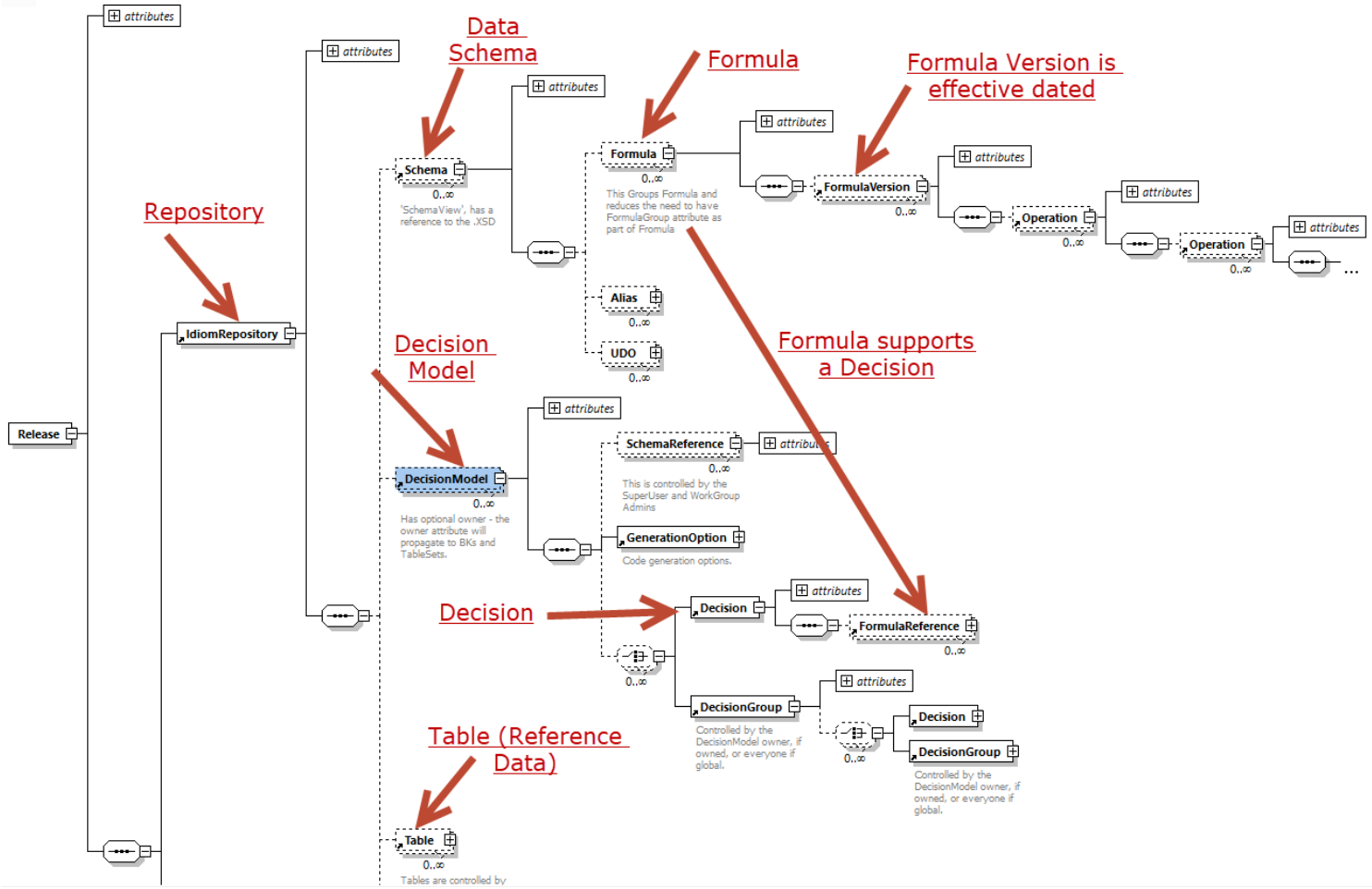


Figure 3 IDIOM Repository Schema

- **Data:** data is described by one or more XML Schemas (.xsd), which supports large and complex multi-dimensional data constructs that are defined independently of the Formulas that use them. This single factor significantly expands the scope of calculations that can be built, and the ability to normalise them.
- **Decision:** A decision is an output value that is calculated by a Formula, and which will be placed in the decision's context node in a schema.
- **Decision Model:** A decision model is a graphically displayed tree structure that provides a visual process map of the decisions and decision groups, and which ultimately controls the sequence of execution of the Formulas.

- **Formula:** Formulas are also graphically displayed tree structures that nest an unlimited number of operations – this is not dissimilar to Excel in approach. A Formula is associated with a node on a schema to provide it with a known data context, which is required to provide data navigation context at runtime. This is essential because of the complex, multi-dimensional nature of the data being managed. A Formula can be used by multiple Decisions, and a Decision can use multiple Formulas over time.
- **Table**: Tables are traditional row/column format reference data, which can be held on-board or linked to external sources. Again, the table's definition (meta data) and values are separated, and exist as different artefacts in the model.
- **Repository:** A Repository is the container for a set of related Decision Models. Any number of Decision Models from multiple Repositories can co-exist in an IDIOM Calculation Engine, allowing Calculation Models of virtually unlimited scale in a single Calculation Engine.

Using the IDIOM building blocks, a Calculation Model can be built that is fully normalised – that is, all data is normalised within the schema definitions; Formulas and their components (aka calculations and sub-calculations) are uniquely defined, named, and located on their context nodes; and Decision Models provide a normalised flow of Decision execution that binds the Formulas to the data in a predictable, transparent, and auditable manner.

Formulas

Formulas are the raison d'être for both Excel and IDIOM Decision Manager. They are the critical artefacts that define the purpose of the spreadsheet or decision model respectively.

Formulas are a capability that is relatively similar in terms of intent and efficacy across the two products. Both products use the term Formula more or less equally. IDIOM additionally refers to the output of the execution of a Formula as a Decision, hence IDIOM Decision Manager. The use of the term 'decision' reflects a focus on more general business logic outcomes in IDIOM rather than the relatively more specific numeric calculations in Excel.

The size and complexity of logic of Formulas in both products is, for all practical purposes, unbounded. It is likely that an Author will be trying to reduce Formula size because of their own limitations before either tool's limits are reached.

Excel

Formulas in Excel are hand scripted into a single line of text that is built up from a palette of available operators. The Author of the Formula text can select available operators from the palette, which then assists the Author by injecting a template for that operator's parameters into the text line. Formula syntax validation is immediate.

The visual presentation of the Formula is as a continuous line of formally structured text. The following is an example of Excel Formula code from a third-party spreadsheet that IDIOM is currently converting – this example is one line of text describing the Formula for a single cell:

=IFERROR(IF(OR(AK23="LINEAR",AK23="PLS"),IF(AJ49>0,MAX(AH3+AI56,AI3-AI3/AI$49*$AJ$49),MIN(AH3+AS56,MAX(MAX(AH3+AI56,IFERROR(TREND(AA3:AH3,OFFSET(INDIRECT(AP19),AJ3,0)*{1;0;0;0}+OFFSET(INDIRECT(AO19),AJ3,0)*{0;1;0;0}+OFFSET(INDIRECT(AN19),AJ3,0)*{0;0;1;0}+OFFSET(INDIRECT(AM19),AJ3,0)*{0;0;0;1},OFFSET(INDIRECT(AP19),AJ3,8,1,1)*{1;0;0;0}+OFFSET(INDIRECT(AO19),AJ3,8,1,1)*{0;1;0;0}+OFFSET(INDIRECT(AN19),AJ3,8,1,1)*{0;0;1;0}+OFFSET(INDIRECT(AM19),AJ3,8,1,1)*{0;0;0;1}),0)),AI3-AI3/AI$49*$AJ$49))),IF($AK$23="LN(Y)",IF($AJ$49>0,MAX(AH3+AI56,AI3-AI3/AI$49*$AJ$49),MIN(AH3+AS56,MAX(MAX(AH3+AI56,IFERROR(GROWTH(AA3:AH3,OFFSET(INDIRECT(AP19),AJ3,0)*{1;0;0;0}+OFFSET(INDIRECT(AO19),AJ3,0)*{0;1;0;0}+OFFSET(INDIRECT(AN19),AJ3,0)*{0;0;1;0}+OFFSET(INDIRECT(AM19),AJ3,0)*{0;0;0;1},OFFSET(INDIRECT(AP19),AJ3,8,1,1)*{1;0;0;0}+OFFSET(INDIRECT(AO19),AJ3,8,1,1)*{0;1;0;0}+OFFSET(INDIRECT(AN19),AJ3,8,1,1)*{0;0;1;0}+OFFSET(INDIRECT(AM19),AJ3,8,1,1)*{0;0;0;1}),0)),AI3-AI3/AI$49*$AJ$49))),IF($AK$23="LN(X)",IF($AJ$49>0,MAX(AH3+AI56,AI3-AI3/AI$49*$AJ$49),MIN(AH3+AS56,MAX(MAX(AH3+AI56,IFERROR(TREND(AA3:AH3,IFERROR(LN(OFFSET(INDIRECT(AP19),AJ3,0)),0)*{1;0;0;0}+IFERROR(LN(OFFSET(INDIRECT(AO19),AJ3,0)),0)*{0;1;0;0}+IFERROR(LN(OFFSET(INDIRECT(AN19),AJ3,0)),0)*{0;0;1;0}+IFERROR(LN(OFFSET(INDIRECT(AM19),AJ3,0)),0)*{0;0;0;1},IFERROR(LN(OFFSET(INDIRECT(AP19),AJ3,8,1,1)),0)*{1;0;0;0}+IFERROR(LN(OFFSET(INDIRECT(AO19),AJ3,8,1,1)),0)*{0;1;0;0}+IFERROR(LN(OFFSET(INDIRECT(AN19),AJ3,8,1,1)),0)*{0;0;1;0}+IFERROR(LN(OFFSET(INDIRECT(AM19),AJ3,8,1,1)),0)*{0;0;0;1}),0)),AI3-AI3/AI$49*$AJ$49))),IF($AJ$49>0,MAX(AH3+AI56,AI3-AI3/AI$49*$AJ$49),MIN(AH3+AS56,MAX(MAX(AH3+AI56,EXP(IFERROR(TREND(AA3:AH3,IFERROR(LN(OFFSET(INDIRECT(AP19),AJ3,0)),0)*{1;0;0;0}+IFERROR(LN(OFFSET(INDIRECT(AO19),AJ3,0)),0)*{0;1;0;0}+IFERROR(LN(OFFSET(INDIRECT(AN19),AJ3,0)),0)*{0;0;1;0}+IFERROR(LN(OFFSET(INDIRECT(AM19),AJ3,0)),0)*{0;0;0;1},IFERROR(LN(OFFSET(INDIRECT(AP19),AJ3,8,1,1)),0)*{1;0;0;0}+IFERROR(LN(OFFSET(INDIRECT(AO19),AJ3,8,1,1)),0)*{0;1;0;0}+IFERROR(LN(OFFSET(INDIRECT(AN19),AJ3,8,1,1)),0)*{0;0;1;0}+IFERROR(LN(OFFSET(INDIRECT(AM19),AJ3,8,1,1)),0)*{0;0;0;1}),0))),AI3-AI3/AI$49*$AJ$49)))))),0.001)

Excel has a library of several hundred operators in its palette, with some individual numerical functions representing substantial capability (e.g. goal seek etc).

IDIOM Decision Manager

IDIOM Decision Manager has a smaller library of available operators, but one which includes all of the data manipulation, business logic, and numerical operators that are required for normal commercial applications processing. The library is easily extended.

The IDIOM build approach for Formulas is entirely drag+drop – there is no scripting at all, therefore all Formulas are syntactically correct when built (you can't build an invalid Formula).

The visual presentation metaphor is that of a tree, where the root is the Formula result (a 'decision') and the leaves are primitives (i.e. a singular datum acquired by the Formula at runtime). See the Formula palette example shown earlier for an example.

General Comparisons

Some other comparisons:

Feature	Excel	IDIOM
Formula evaluation	The formula evaluation dialog box is modal and doesn't allow concurrent inspection of other elements.	Formula evaluation is modeless. Any repository, decision model component and property can be accessed via the meta model.
Documentation	No documentation features other than the spreadsheet itself.	Built in 'Logical English' PDF documentation generator. User can maintain Business and Technical descriptions and can document Formula components.
References	Formula references to data and other formulas are abstracted so that the meaning is difficult to follow.	IDIOM Decision Models and Formulas use graphical presentation and fully qualified names throughout.
Debugging	Excel's Formula evaluation doesn't have a breakpoint feature. The Author can only evaluate one Formula at a time and only step into one reference at a time, and cannot evaluate Formulas that are (re)calculated as the result of other formulas.	IDIOM's debugging features include: Step into and step over decision (group)s; set breakpoints for any Formula; set a trace on any Formula operation; store the results of Formula operations in IDIOM's internal xml file.
Cross references	Cell Precedents can only be displayed for cells with references; Cell Dependents can only be displayed for cells with Formulas.	Comprehensive 'Show where used' feature available for every IDIOM component: xml nodes, tables, decision and groups, formulas.

PROCESSING MODELS

Execution Model

The execution model describes how each Formula is selected and executed during an execution cycle of the spreadsheet or Decision Model respectively.

Excel

The Excel Formulas are assembled into Worksheets, which are in turn aggregated into Workbooks, also known as Spreadsheets. Spreadsheets can then be linked together into loosely coupled groups of linked spreadsheets.

The Excel execution cycle is triggered by changing the value of any cell. In normal use, when the value is changed, all cells that reference the triggering cell's data are recalculated, and so on until all dependent cells have been recalculated across all Worksheets and linked Spreadsheets (if they are available). The processing cycle can start with any cell, and can cause a change in any other cell that is linked by this inferred dependency chain.

The Author does not have to declare any form of processing model because it is inferred directly from the dependencies between cells that are implicitly declared within the Formulas. The downside is that the Author cannot easily visualise or control the execution model. This a an important reason why it is difficult to build Excel models that can respond to multiple and varied use cases, versioning, and polymorphism (all discussed later in this document).

IDIOM Decision Manager

IDIOM Formulas are grouped into Repositories through association with their context Schemas. The Repository will also contain Decision Models, which provide the process flow models for execution of the Formulas, which are linked to the Decisions in the Model. The Decision Models can also use Formulas to manage their internal workflow. There is a default workflow, but not an inferred workflow. This ability to control Decision Model execution through explicitly defined logic blocks in purpose specific process control Formulas is important to managing multi-use, Versioning, Polymorphism, and Testing.

In a deployed application, the Transaction that is responding to a run-time event must supply that Transaction's context data as XML. This data is passed to the IDIOM Calculation Engine, which will execute all required Decision Models until all required Decisions/Formulas have been executed in the order specified by the models, so that the context data is once again complete, consistent, and correct in its new state, at which time it is returned to the caller to close the Transaction cycle. All IDIOM Transactions are stateless and execute independently of each other, so that any number of Transactions can run in parallel to achieve unlimited throughput.

Versioning

Versioning is a generic business requirement inherent to all systems that contain long-lived calculation models. In general, Excel does not provide any mechanisms to support versioning. Some basic versioning can be attempted using date based lookups on cell data and/or date switching inside individual Formulas etc. However, this is complex and limited in depth and scope, so that spreadsheet versioning almost always means a new version of the spreadsheet file itself.

For production spreadsheets that are used by more than one person, managing versions poses significant operational problems in Excel.

IDIOM Decision Manager has explicit support for versioning at all levels of the repository. Formulas have multiple versions, which are effective dated and automatically selected by their effective date to match the effective date of the context data. Similarly, Tables have multiple versions of reference datasets that are effective dated and automatically selected

at runtime. Effective dating of individual rows of reference data is also supported by explicit date 'ranging' operators that are automatically applied at runtime.

Decision Models manage date sanctioned branches within the models, and entire decision models can be effective dated through use of the IDIOM Control Model concept. The consequence is that an IDIOM calculation model can be versioned indefinitely at every level, so that it is 'nimble, continuous, and perpetual' in the words of one IDIOM customer.

Polymorphism

Polymorphism is a topic related to versioning. The term refers to the ability of a single calculation model to perform many different calculations depending on the context data at run-time (as does versioning).

Calculation design patterns for most commercial problems are quite finite, with a limited et of different approaches and algorithms being reused in different combinations across different contexts.

When the individual components of a calculation model are built and tested as separately addressable artefacts, as they are in an IDIOM Repository, then they can be easily reused. This means that an IDIOM Decision Model can address multiple problem classes by re-ordering and reusing its internal logic components; that is, it is polymorphic.

What is required is a dynamic process model that can orchestrate the calculation components in response to the given context. The IDIOM Decision Model provides this degree of dynamic process control with a separate layer of workflow control Formulas that explicitly address process flow.

Without an active meta model, this capability is not feasible with Excel.

Testing

As a general comment, good testing benefits from a range of features:

- **Step-by-step testing:** As development proceeds, the Author must build up a library of tested calculation components. Step-by-step testing allows the Author to test and review every Formula element working up from the primitive operations that comprise each Formula, through to the Formula itself and eventually the entire calculation model.
- **Separation of calculation components:** As the calculation components are built out using the above approach, they are promoted into a tested status. Building out a large calculation model (e.g. hundreds of sub-calculations nested inside dozens of primary calculations, all effective dated) requires the Author to build, test, and promote selected calculation components individually, without needing to execute all aspects of the calculation at all times.
- **Testing at scale:** The ability to pass a wide range of test cases through the calculation model including boundary testing and regression testing. For production acceptance testing, millions of test cases may be required (for instance, all production use cases).

- **Testing against known true outcomes (regression testing)**: When this form of regression testing is used, the context data must execute in matched pairs through a single execution model.
- **Champion/Challenger testing**: In contrast with the point above, champion/challenger testing requires testing the same context data with two variants of a calculation model. In each case, the results must be kept for comparison purposes at every level.

For each of the above testing capabilities Excel provides no automated assistance, so the Author needs to perform substantial manual testing. Even then, testing at scale is simply not feasible.

Furthermore, there is substantial risk of inadvertently changing an Excel model during the testing process itself. The question for the Author then becomes – how would you know, and how would you verify that this has not happened? The answer is, you don't and you can't.

The IDIOM Decision Manager and its associated suite of tools provides automated testing assistance across all of the above testing categories.

Final note: In converting many spreadsheets ranging up to 100 megabytes in size, IDIOM has always found errors in the spreadsheets. Errors are very difficult to eradicate in an Excel environment, and any proprietor of Excel spreadsheets should plan for this accordingly.

Scalability

As the size and complexity of the transactions and throughput requirements increase, Excel will eventually run into performance and scalability issues that are not easily resolved. IDIOM Decision Manager is, on the other hand, a true server application that can be scaled vertically. The problem is dealt with by addressing each use-case individually in a separate Transaction, rather than all at once in a worksheet.

Calculation Speed

Calculation speed in Excel has the advantage for a number of reasons. The regular and limited data format and dynamically inferred process flows together mean that the calculation algorithm can be highly optimised, including processing multiple calculation streams at once. There is only one underlying 'process flow', which the Excel developers have tuned to high performance levels.

Where the spreadsheet needs to process multiple rows of unrelated data, this ability to multi-stream the calculation process provides a distinct speed advantage.

The IDIOM calculation flow is sequential in nature, which allows increased control and normalisation of calculation components, but which nonetheless must be evaluated step-by-step at runtime. Notwithstanding that the IDIOM Calculation Engine, including each Decision Model within it, is fully compiled, this calculation speed will not match the optimised Excel equivalent in those cases where Excel's parallel processing advantage can be used.

In most commercial transactions, this is not an issue, and calculation speed would not be visible to the transaction user. The typical IDIOM Decision Model will execute in a few milliseconds, with even large Calculation Engines usually completing in fractions of a second.

The difference could become visible in those cases where large datasets are processed inside individual IDIOM transactions. In these cases, the IDIOM approach will process each element in sequence; the Excel approach would likely process these in parallel. For the sake of clarity, using multi-record transaction in this way is not considered to be good practice in transaction design.

DATA

Transactions versus On-Board Data

The Wikipedia Quote above stated that "Excel was not designed to be used as a database". Nonetheless, a single Excel spreadsheet can hold a substantial volume of data[11]. This ability to quickly amass data and logic at scale is profoundly useful to a calculation Author during the R&D phase of a calculation, and is useful under laboratory conditions that do not need to respond to external events in a production context.

However, the implication of processing data 'on-board' the spreadsheet is that while Excel is not designed to be used as a database, it is by definition a database from the perspective of the calculation model itself. All data used by the calculation model must be accessed in or through individually addressed cells in the spreadsheet.

At the same time, there is no practical ability to process context data that is supplied transactionally at runtime. This means that Excel based calculations are not a candidate for traditional transaction processing and/or database processing; the substantial benefit that Excel offers during laboratory development of calculations becomes a substantial barrier to deployment in a commercial, multi-user setting. Deployment of multiple copies of an Excel spreadsheet for production use carries significant operational risk – they are opaque, difficult to test and audit, difficult to version, and difficult to protect the IP therein.

IDIOM Decision Manager on the other hand, defines all of its calculation logic against a data meta-model (one or more XML Schemas), and then generates an executable that will substitute real transaction or database data for the meta-model at runtime.

The meta model always describes its subject entity in a known state, and the IDIOM calculation model is responsible for transitioning that state to a new state. Because the IDIOM model executes a state change, it is by definition stateless, and any number of concurrent run-time instances can be used to achieve unlimited throughput, so that an IDIOM deployed Calculation Engine can execute at any scale.

[11] 1,048,576 rows and 16,384 columns (A1 through XFD1048576). Each cell can hold a maximum of 32,767 characters. This capability is significantly extended through Power Pivot, which has a maximum capacity of nearly 2 billion rows.

The practical scale of individual XML records approaches but does not match that of Excel, so that the XML 'on-board' data is usually limited to hundreds of thousands of nodes in total, rather than the millions/billions that Excel can process. This is not a limitation of XML; rather, it is a physical limitation on the runtime resources required to process very large schema defined objects.

One way to sum up the difference in approach is to say that with Excel, the Formula goes to the user where the data is as an xls/xlsx file; with an IDIOM deployed solution, the data is sent by the user to the Formula as a transaction context.

To get an Excel Formula to the data usually means shipping an .xls file to the user who will process the data. In a commercial applications context, the processing of individual transactions is normally a distributed function, so that if Excel is used as a calculation engine for production data, then the .xls files are also likely to be distributed. This means that all intellectual property in the file is exposed to all points of representation where transactions need to be processed. It is worth noting that, perhaps perversely, the IP in a spreadsheet can be opaque to people trying to maintain the spreadsheet, but transparent to those trying to steal it.

By way of contrast, the IDIOM Decision Manager approach uses completely standard Transaction processing technologies. Getting the data to the Formula can be achieved in real-time using any secure messaging approach (secure webservice, queues, batch files etc.), or by processing database transactions in batch. Regardless of how the data transfer is achieved, the IP in the IDIOM calculation engines is never visible or accessible to third-parties.

Data Structure

Transactional data has a natural structure that is typically a series of overlapping hierarchies, with the root node of the primary hierarchy presenting the transactional context[12]. This data structure is an important factor in defining Formulas and the process flows that execute them. This data structure cannot be represented in rows and columns without an externally defined entity/relationship model to describe it (at least!). This E/R model does not exist in Excel.

It is for this reason that IDIOM Decision Manager uses the XML Schema standard to define its data elements. The XML Schema standard (.xsd) can define complex data structures, which IDIOM leverages to navigate up and down complex relationships in both the Formulas and the process flow that is executing the Formulas.

Implementing an equivalent process would be a substantial challenge in Excel, and could only be achieved in even a simplistic form with a disproportionate amount of manual coding.

[12] For an extended discussion on this topic see our article on Modern Analyst "Requirements and the Beast of Complexity".
http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/ArticleID/1354/PageID/1369/Default.aspx

Data Types

Excel is not strongly typed, which is consistent with providing an easy to use, and very forgiving tool. Strong typing means that the data type must be declared at a meta-level, to be enforced by the tool when the data is entered.

Excel does allow the Author to specify a range of data types (as formats) however these are not enforced, because entering a non-complying data type generally causes Excel to automatically convert the datatype[13]. This is an additional risk to be managed when testing.

SECURITY OF INTELLECTUAL PROPERTY

The Intellectual Property [IP] in many calculation models is fundamental to the success of the owning organisation. Protecting this IP should be an important criterion in tool selection.

Excel

Excel Workbooks are by definition 'personal' IT artefacts. This often means that they are subject to the following risks:

- They are not easily managed under standard IT source code management practices, so that they often allow users to bypass company procedures and policies.
- They often reside on c:\ drives and may not be backed up.
- The versioning is usually done with a somewhat blunt approach: "… Copy.xlsb", "… Copy (1).xlsb", …, "Copy (6).xlsb".
- The workbooks can be shared by Email and file transfer with colleagues, which creates multiple instances of the 'same' data and logic, and potentially exposes IP.
- The Excel workbook can connect to backend databases using connections that may be set up during the installation of other business applications. These connections can have a 'live link' or 'copy' property and the IT department may not be aware that the connections have been persisted. Refreshing such a connection will invoke queries that can download entire database tables.
- Once access is granted, spreadsheet users can write their own database queries, which may be executed against potentially production databases. These users can learn how to write Excel queries that update and delete data simply by watching YouTube videos[14].
- They are often undocumented.

Excel workbooks are like icebergs; a user can only 'see' exposed parts of its data and logic, and may not realize that the view could be restricted. Excel has several standard features that can submerge data and logic:

- Hidden worksheets, hidden rows and hidden columns.

[13] Note that Power Pivot and Power Query are more strongly typed.

[14] https://www.mssqltips.com/sqlservertip/1540/insert-update-or-delete-data-in-sql-server-from-excel/

- Hidden (protected) worksheets (which can be unprotected with a US$20 password breaker).
- Data in white font.
- Connections to backend databases complete with login credentials including the password.

IDIOM Decision Manager

Decision Models in IDIOM Decision Manager are managed on centrally managed servers, subject to standard IT management practices including access control, backup, and change logging.

Deployment of an IDIOM Decision Model is required to navigate an IT managed process that is likely to include generation of comprehensive end-to-end documentation, version control of the models, automated generation of executables, and injection of all relevant artefacts into the organisation's ultimate source code repository.

Provided that the executables operate within a secure operating environment (and if this is not the case, then the organisation is inherently insecure), then there is zero security exposure of the IP contained therein.

SUMMARY

The power and scale of Excel as a calculations modelling tool is well known and respected. Excel provides a rapid prototyping capability that is accessible to many people with a need for defining calculation models.

Excel can be used to provide an effective production solution when all of the following are true: The entire range of required use cases can be processed on-board a single Excel spreadsheet; The spreadsheet does not need to automatically respond to external events; And neither the data nor the function need to be distributed (i.e. the spreadsheet will remain in a laboratory setting).

Excel is also an excellent tool to present data to the end-user, and the final output from an IDIOM Decision Model often takes the form of an Excel Spreadsheet.

However, when the calculation model is required to process online transactions or database records, and/or to automatically respond to external events across a network, then another solution is required. IDIOM provides this solution.

Mark Norton | CEO and Founder | IDIOM Limited

☎ +64 9 630 8950 | +64 21 434 669 | Australia free call 1800 049 004
✉ 2-2, 93 Dominion Road, Mount Eden, Auckland 1024 | PO Box 60101, Titirangi, Auckland 0642, New Zealand
@ mark.norton@idiomsoftware.com | idiomsoftware.com | Skype Mark.Norton

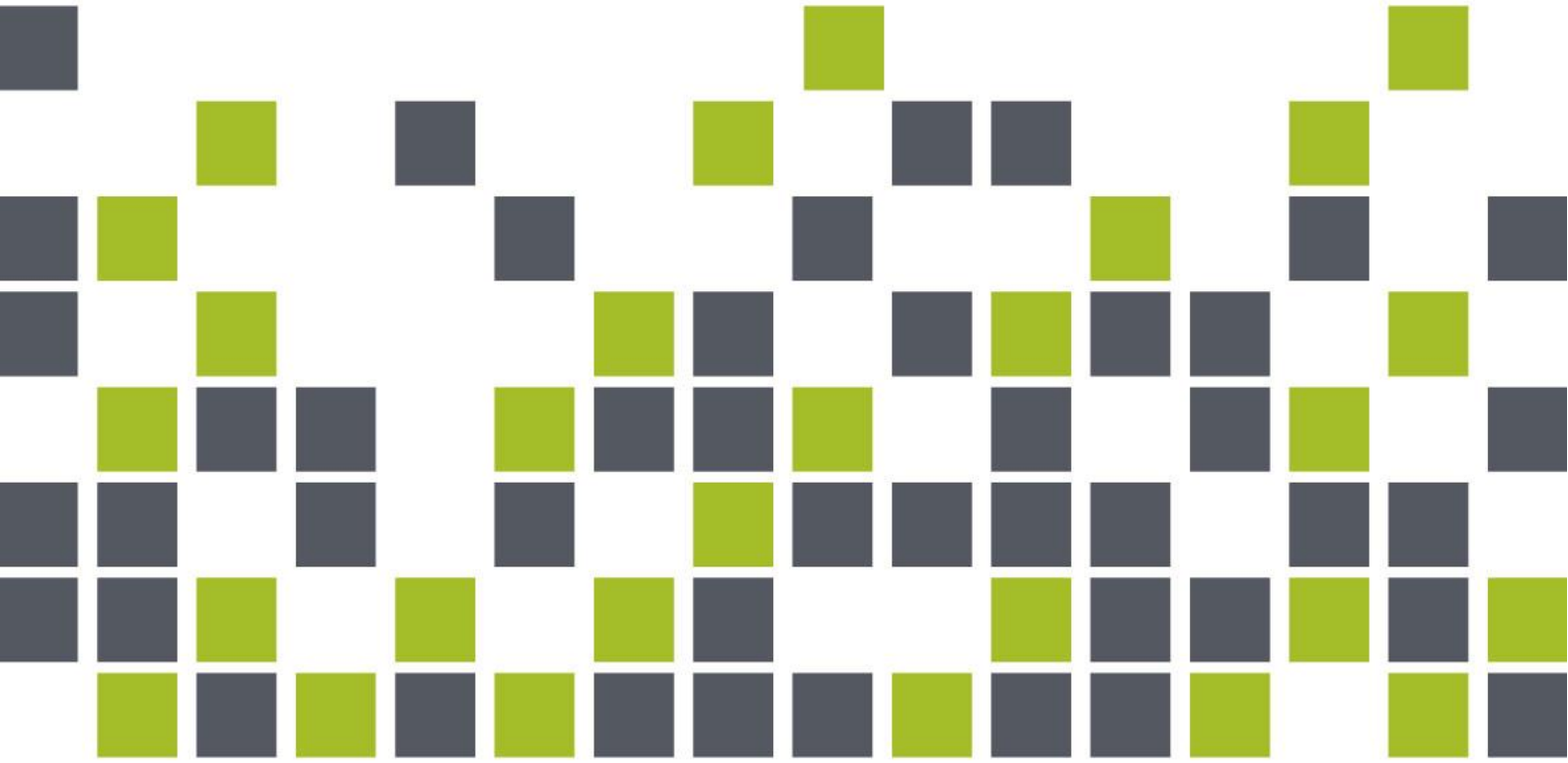
About IDIOM

Established in 2001, IDIOM Limited is a private company based in Auckland, New Zealand.

IDIOM develops and licenses decision-making software that automates business policy on a large scale, making systems more transparent, more agile, and more durable, while reducing development time, cost, and risk.

IDIOM's innovative business oriented software is used by business users to graphically define, document, and verify corporate decision-making and related business rules; it then auto-generates these into small footprint, non-intrusive software components for use in systems of any type or scale. IDIOM is a pioneer in the development and use of decision automation concepts, and has applied these concepts to develop and automate business policy for Partners around the world in local/state/central government, health admin/clinical health, insurance/superannuation/finance, telecoms, logistics, and utilities.

IDIOM develops and licenses: IDIOM Decision Manager™, IDIOM Forms™, IDIOM Document Generator™, IDIOM Dialog Manager™, IDIOM Mapper™, IDIOM Tracker™, IDIOM Decision Manager Workbench™ and the IDIOM Transaction Engine™



☎ +64 9 6308950 | @ idiomsales@idiomsoftware.com | idiomsoftware.com
✉ 2-2, 93 Dominion Road, Mount Eden, Auckland 1024