# Contents

**About IDIOM** ..................................................................................................................................................................... 3

**INTRODUCTION** ......................................................................................................................................................... 4

**Automating Business Policy in a Customer Platform** ................................................................................................... 4  
  Further Effects of the Approach ....................................................................................................................................... 6

**IDIOM Customers** .............................................................................................................................................................. 7

**IDIOM PROPOSED SOLUTION** ......................................................................................................................................... 8

**The Approach** .................................................................................................................................................................... 8

**Loan Origination and Management Application** ............................................................................................................ 9  
  The Database ...................................................................................................................................................................... 10  
  Financial Products Builder Platform .................................................................................................................................. 12  
  Credit Checker .................................................................................................................................................................... 12  
  Other System Interfaces ...................................................................................................................................................... 12  
  Auto Migration .................................................................................................................................................................... 12  
  Versioning ........................................................................................................................................................................... 13  
  Consumer Use of LOMA ................................................................................................................................................... 13  
  IDIOM Forms ....................................................................................................................................................................... 13  
  Workflow ............................................................................................................................................................................ 13  
  Document Generation ........................................................................................................................................................ 14

**MAKING IT WORK WITH IDIOM** ................................................................................................................................. 15

**IDIOM Tools** .................................................................................................................................................................... 15  
  The IDIOM Decision Manager ........................................................................................................................................... 15  
  IDIOM Forms....................................................................................................................................................................... 18  
  IDIOM Decision Tracker ...................................................................................................................................................... 19  
  IDIOM Mapper .................................................................................................................................................................... 19  
  IDIOM Decision Manager Workbench ................................................................................................................................ 19  
  IDIOM Document Generator ............................................................................................................................................... 21
About IDIOM

IDIOM was established in 2001 by NZ private interests specifically to enable development and deployment of financial products by business specialists without recourse to IT technical experts. IDIOM’s ultimate goal is for “game-changing agility for IDIOM’s customers in their chosen markets”, with appropriate consideration also given to quality, reliability, scalability, compliance, security, efficiency, and transparency.

In achieving this goal, IDIOM pioneered the development of decision automation concepts and approaches, and their practical application to the design and development of systems of all sizes. IDIOM has applied its “pure decisioning” tools and approaches to improve business agility, and systems development performance, in projects in Europe, Asia, North America and Australasia, across such diverse domains as insurance/finance, health admin and clinical health, municipal, state, and central government, telecoms, utilities and logistics.

IDIOM leverages this experience in developing and marketing the “IDIOM Decision Manager”, a purpose built decision automation tool. IDIOM Decision Manager is a proven, pragmatic and cost effective tool for capturing, managing, automating and deploying business decision making know-how. For a general background to this concept, please see the IDIOM authored article “Requirements and the Beast of Complexity”, published on Modern Analyst.

The following tools are currently in the IDIOM portfolio and all may have a role in this proposal (each is further explained later in this document):

- **IDIOM Decision Manager** is a tool for the SME and/or analyst to graphically model, test, document and deploy complex business decision-making as fully executable, high performance ‘decision models’ – without programming!

- **IDIOM Forms** is a tool to define and deploy large, complex Web2.0 forms that are tightly bound to IDIOM decision models at execution time, field by field. The decision models are used to apply all business logic, to control workflow, and to dynamically control the form's look and feel.

- **IDIOM Decision Tracker** is a tool to map MS Word and MS Excel documents to IDIOM decision models for full bi-directional traceability between corporate policy definitions and their implementation as IDIOM generated decision models.

- **IDIOM Document Builder** is a tool to define, test, and deploy MS Word document specifications, and a document generator that uses those specifications to build transactional MS Word documents under the control of IDIOM decision models.

- **IDIOM Decision Manager Workbench** is a user operable application to acquire decision models and other process components, assemble them into processes, and to run them on a large scale. It also collects outcomes and supporting information per transaction and/or per process for subsequent analysis and/or action.
INTRODUCTION

This document describes a generic loan origination and management application that is built around a core of credit and workflow decisioning. The solution is designed to allow integration with existing back-end applications, as well as external credit scoring agencies.

The application is built using IDIOM’s Agile Decisioning Approach. This proven and standardised design approach assumes a small bespoke framework that provides an application container for multiple IDIOM tools that define and execute financial product definitions. The IDIOM products are used to configure and execute financial products within the framework, and a configuration platform is provided for this purpose. Configuration of financial products is quality assured using the extensive regression testing and product simulation capabilities provided by the IDIOM Decision Manager Workbench.

In this document we traverse several related concepts: business policies, decisions and decisioning, and financial products. In IDIOM’s view, business policies define how the business will respond to external events to generate value for the business; decisions are the means by which the policies are codified, tested, deployed, and ultimately executed at all points of representation. Financial products are an important and publicly visible manifestation of business policies.

Product specific policies within financial products determine how the business will accept applications for each product, at what price, and under what terms and conditions; additional, more general operational policies dictate workflow, product lifecycle management, and document generation. IDIOM addresses all of these policy automation needs using the declarative, business operable IDIOM Decision Manager within the bespoke application proposed herein.

Automating Business Policy in a Customer Platform

IDIOM’s ‘policy as content’ metaphor is original and we believe remains unmatched in delivery and execution. In support of this we direct your attention to our article ‘Simplify Systems with Decision Centric Transactions’ – this article is substantial and provides a full outline of the ‘policy as content’ approach, including through reference to other IDIOM published articles.

IDIOM’s ‘model driven development’ ancestry has allowed IDIOM to fully automate business policy on a large scale, with further and significant application architecture and systems development benefits. Using the IDIOM tools, the complete ‘business policy life cycle’ (including policy development and authoring, codifying, testing, generating and deploying, and operational execution) can be delegated back to the true business owners, leaving the underlying application framework to focus on the technology dependent functions, including:

- Database: persistence, search and find, audit history and archive;
- Integration: content based routing, validations and transformations, access to integration connectors and end points;
- User access: authentication and authorization, session management, work-lists;
And provision of a ‘virtual environment’ that supports the full scope of the product life-cycle – in this case, loan origination and management.

The separation of policy (aka product) implementation from the underlying systems infrastructure is logical – they have distinct and independent drivers and development life cycles. The drivers for changes in the systems infrastructure are externally defined by changes in technology. The drivers for policy changes on the other hand, can range across routine business activities (e.g. contract negotiations); changes in the regulatory environment; and/or other market driven or internal policy changes. Regardless of the trigger, policy changes can be overlapping both in terms of scope and time, and can be both urgent and complex relative to the technology driven changes.

In the traditional systems development worldview, a single system image is shared by a lifetime of overlapping policy driven and technology driven changes – the result is increasing complexity that progressively increases cost, time, and risk to implement new changes. Because of this complex intertwining of technology and business policy in a single system image, with compounding complexity and opaqueness over time, any substantial stepwise change in underlying technology usually implies a rewrite of the business policies therein – avoidance of this significant, risky and expensive operation is the primary reason for the existence of ‘legacy systems’. Separation of the independent technology and business policy life-cycles is both logical and beneficial, allowing each to remain independent of the other, able to be replaced at will – there need never be another legacy system.

One of the greatest benefits of the IDIOM approach is this separation of concerns between these competing change drivers. And it works - substantial regional and global organisations have successfully implemented systems based on these approaches on a large scale, allowing policy changes to be implemented on a daily basis if necessary, independently of the much less frequent but higher risk host application upgrades.

IDIOM is now seeing further benefits of the approach. When business policy is able to be generated into self-contained executables, it is a simple matter to build a small, tightly engineered generic framework that houses an independent and propitious ‘policy development cycle’ as proposed herein, including policy definition, testing, simulation and what-if analysis. This gives rise to a learning organisation, itself an important additional benefit of IDIOM’s approaches.

In a customer setting, this mutually beneficial separation of responsibility provides the business product owner with outstanding agility with regards to the management and implementation of their products, while the underlying system remains robust and unaffected by changes in business policy.

IDIOM is a tool designed to empower business owners in the management of business policy, including the policies that define financial products. These business owners become the hands-on users of IDIOM, and are able to take full control of the product development and deployment life cycle, often unassisted by IT. Of course, IDIOM is happy to provide ongoing services in support of business policy rules, but it is an ongoing objective of IDIOM that these services should be limited to implementation assistance and then infrequent periodic support as required – our objective is to empower the business product owner, not make them dependent!
Further Effects of the Approach

Building a complex product life cycle process using the above approach has other positive downstream effects.

1. The ‘decision model’ is tightly structured, and able to be rendered into many formats, including logical English, computer source code, and XML. It is therefore transparent, auditable, and reliable as a specification of the executable code and the business policies that it automates.

2. This is a tool for the SME. The high degree of automated assistance and reduced development complexity means that the SME is only contributing what they already know – their own proprietary business knowledge. Productivity is quickly and reliably achieved; arcane IT skills are not needed and offer no advantage.

3. Generation of computer source code means that the code per se does not need to be tested; only the logic supplied by the SME needs to be tested. This means a substantial reduction in testing effort.

4. The automated tool assistance and reduced human input reduces the potential for errors. This gain is compounded by the extensive on board testing support already described, so that it is rare for a decision model to be anything other than complete, correct, and consistent when initially released into a formal black-box testing process.

5. The decision model itself, and all the artefacts that support it, fully support perpetual, continuous versioning. When the models are implemented as ‘content’ in computer systems, they provide a practical and robust mechanism for business SMEs to independently develop, manage, and deploy the organisation’s codified knowledge (including complex calculations, workflow, and other business policies) on a daily basis – continuously and perpetually.

6. The decision models are technology agnostic and technology independent, forming a complete historical record and the ultimate ‘source of truth’ for the organisation’s proprietary knowledge; extracted, tested, confirmed, and documented by that organisation’s SMEs in the normal course of business.

7. A tool assisted approach as described allows SME’s to capture, test, and deploy this knowledge extremely quickly. In a 100 developer year project, 80% of the system code was generated from decision models that were built in less than 20% of the total development hours – all of which were contributed by business analysts drawn from business branches.

8. This development efficiency is sustainable over the long term, offering huge business agility with reduced cost, time, and risk

9. A decision model is a durable life-long artefact that defines the business in perpetuity. As a complete record of organisation knowledge, and with multiple and extensible source code generation options, there should never be another ‘legacy system’.
IDIOM Customers

IDIOM has more than 50% of its business in Australia, with a majority coming from the financial sector. Especially relevant customers include business units within each of the four largest Australasian insurers, including QBE-LMI. IDIOM replaced the LMI decision engine (Decision Point) at QBE-LMI (then PMI) in 2005.

Other financial sector customers include Sydney based Toyota Financial Services; multiple Australian superannuation fund administrators; and two of the largest life insurers in NZ. Looking further afield we include Cicada, a data validation service for European stock exchange transaction tickers; Counterparty Link, a UK based ‘know your client’ data specialist; K&P, an Italian technology provider for loan brokers; and MicroEnsure, a UK based low cost insurance channel with 10’s millions of lives insured.

IDIOM is also one of the largest health billing engines in the world through our partnership with vendor PowerHealth Solutions, and is present in most hospitals in Australia as well as around the world. IDIOM assisted with the development of billing rules for the initial Power Billing and Revenue Collection customer (the Hong Kong Hospital Authority, with 47 hospitals and 121 outpatient clinics, generating 1 million invoices a week).

Other IDIOM users include multiple NZ Government Ministries, the Queensland State government, multiple municipal authorities, an award winning airline, and other miscellaneous utility and service providers.
IDIOM PROPOSED SOLUTION

The Approach

The IDIOM approach leverages the IDIOM toolset, all of which builds on our flagship product, the IDIOM Decision Manager. The IDIOM Decision Manager is an ideal solution to the core decisioning requirements of the Loan Origination and Management Application [LOMA] including credit decisioning, workflow decisioning, and external systems related data transformations.

We would like to reaffirm that our ultimate goal is for “game-changing agility for the business in its chosen market”, with appropriate consideration also given to quality, reliability, scalability, compliance, security, efficiency, and transparency. Achieving agility while sacrificing any of these qualifiers is an ‘own goal’ to be avoided.

To ensure that the systems that are delivered meet all of these qualifiers at the time of deployment requires a comprehensive quality assurance process. This process is discussed more fully in the ‘Simplify Systems with Decision Centric Transactions’ article. It is important to note that while the quality assurance process is comprehensive, it still allows for daily deployment if this is required.

Agility may infer a high rate of change, however, that is not the intent here. What we mean by agility is a lack of constraint on the business; its systems should support new products and services in market time, not in systems development time. Some markets may demand a higher rate of change; some may not. Either way, we need to be mindful that the system is conceptually in a permanent state of continual change. In order for this to not become a constraint, we need to actively consider “continuous, perpetual versioning” of the entire system image.

IDIOM’s products, approaches, and market positioning are all about empowering our customer’s business product owners and SMEs to be able to operate independently of the traditional SDLC – this is the essential IDIOM value proposition! To facilitate this, the rules development exercise could and should start with IDIOM resources ‘buddied’ with the Lender’s SME’s, so that leadership in the codifying of rules can transition from IDIOM to the Lender during the initial tranche of development, to be managed entirely in-house thereafter.

Our industry experience suggests that credit decisioning rules development is likely to be in the range of a one month effort, give or take a couple of weeks, depending on how well articulated these rules are at the beginning. It is both plausible and desirable to start this effort immediately, prior to or in parallel with development of the proposed core LOMA platform. If minimal early phase deliverables are selected as the initial development task, this could be implemented and delivering value in as little as 3 months elapsed.

As you might expect, IDIOM is comfortable building a platform around the IDIOM products to achieve an optimal, cost effective and risk averse bespoke business outcome. The IDIOM approach is a well-developed and proven approach for just-in-time, low cost, low risk development. The very nature of the IDIOM products significantly reduces the size of the bespoke parts of the overall application for a much reduced overall development effort. By way of example, we have an Australian insurance customer wherein ~80% of all executable code is IDIOM auto-generated – and this generated code cost 1/20th of the cost of the same number of lines of code built by hand in the framework application.
For the sake of clarity, the massively reduced scale of the underlying framework application means a small build size for reduced time, cost and risk. With this in mind, we always assume that this part of the LOMA is developed bespoke for each Lender.

Accordingly, IDIOM proposes to build out the bespoke framework (we will call this the ‘LOMA Framework’) that will contain and orchestrate the artefacts generated by the IDIOM toolset, including decision models, forms, configurations and documents, using the IDIOM ‘Agile Decisioning Approach’. A critical technique used by this approach is store the credit applications and all other loan context data natively as XML (in an XML data-type column in the database) – this data will not (and must not) be described in the native database definition language [DDL]. This simple design technique reduces build time, cost and risk by substantial margins; it is the key to business agility, and will also improve application run-time performance.

While IDIOM is comfortable building this application to any commercial platform or architecture, our proposal assumes a Microsoft stack. Our experience is that this will significantly reduce development effort, and it provides useful capabilities ‘out-of-the-box’. For instance, with respect to the native XML data stored within the database, Microsoft’s SQL Server provides query and report writing tools that seamlessly span DDL and XML defined data; and if SQL Server’s native full text search is enabled, Google-like search can be easily provided to LOMA’s users across the full extent of the loan data.

**Loan Origination and Management Application**

The Loan Origination and Management Application delivers the following capabilities in line with the adjacent diagram [Figure 1]:

- A ‘Decision Engine’ core to support an IDIOM Forms enabled loan origination and management user interface. It will also be callable by any other system if needed for reuse of credit decisioning rules and related capabilities.
- A database to hold all loan applications, as well as the product and policy configuration artefacts, including Decision Models.
- A financial products ‘builder’ platform, to include IDIOM Decision Manager [I], IDIOM Forms Builder [F], IDIOM Document Builder [D], and the IDIOM Decision Manager Workbench [W]. The IDIOM Decision Manager Workbench is used to aid product regression and scenario testing.

---

1 If the Lender would prefer to build to a Java, open source, and/or proprietary framework, we can do so. Cost is likely to increase.
The Database

The database will be more or less as per the following schema [Figure 2].
Note several features of this data architecture:

- All Green entities are part of the ‘Product’ definition.
- A Product is ultimately manifested by its current set of decision models and its ProductConfigXML, which is a set of product parameters held in XML format that modify the rules being executed by the decision models. Together they fully implement the credit policy for each and every product.
- Each Loan (or credit application or other product) instance belongs to a Product Class.
- The actual Loan data is held in the LoanDataXML column of the Loan entity. Every version of every loan XML is retained in the Loan History table.
- Each Loan can be associated with a distinct Mortgage Partner which also has a set of loan configuration parameters as described in the PartnerConfigXML,
- The ProductConfigXML and PartnerConfigXML will be maintained by purpose built IDIOM Forms.
Financial Products Builder Platform

The Financial Products Builder Platform (see Figure 1) is a platform to support the lending ‘policy development life cycle’. It includes two IDIOM Forms within a simple list based application built over the Product Class and Mortgage Partner tables, to manage their respective configuration documents [styled in Figure 2 as ‘ProductConfigXML’ and ‘PartnerConfigXML’ respectively]. The credit policy manager (SME) will use these forms to update these configuration documents directly.

The primary product within the platform is the IDIOM Decision Manager, which is used to define, test, document, and deploy the underlying credit policy, workflow rules, data mappings and transformations, and other miscellaneous decisioning tasks. Development of these rules is aided by the regression testing and simulation capabilities of the IDIOM Decision Manager Workbench. The Workbench is capable of running multiple versions of all decision models in series and in parallel across the entire production portfolio (and/or on a copy thereof) to facilitate this regression testing (backward looking) and policy simulation (forward looking). It can also provide full data masking as a product option.

The platform provides support to generate the validated and tested ‘production release’ decision model binaries directly into the Decision Model Binary table. Note that the runtime binaries are loaded by the LOMA Framework directly from the database, taking into account state and version attributes.

Credit Checker

We suggest putting the use of vendor credit data under the control of a decision model (see ‘Credit Checker’, Figure 1). That is, the system will run the decision model to determine which vendor data should be requested for each credit assessment. This puts the control of these relatively expensive services under the control of the product SMEs via the Financial Product Builder Platform.

Other System Interfaces

A generic process will be used to connect each ‘other system’ interface, whether inbound or outbound. From that point, development of each unique system interface will require a similar work effort (1-2 weeks) that cannot be made generic.

Each outbound interface will be supported by a decision model that maps the LoanDataXML onto the schema defined XML that is expected by the target system; the resulting XML document will then be committed to the LOMA database and passed asynchronously to the target system via an integration service (Web Service, ESB, simple queueing service, etc). The reverse process will be used for inbound services.

In the first instance, the target outbound system is the ‘back-end’ system. The target in-bound systems are selected credit scoring vendors.

Auto Migration

Auto migration of loans is built into the Application, supported by decision models.

In this case, for each forward looking change in a Product definition (for instance a new field is added, or a new value calculated), a decision model is created that will update the LoanDataXML to match the new product schema and rules. LOMA will inspect the product version of the inbound loan documents.
against the product version of the current configuration document for the relevant product. If the product configuration indicates a later version AND if the product indicates that the new rules apply to existing transactions, then LOMA will call the version migration decision models in succession until the current product version is reached.

Note that this is optional, and can also be conditioned – for instance, product changes might only be applied after the 2nd anniversary of a loan, or when some other threshold is reached, etc.

**Versioning**

All elements of the IDIOM platform are effective-date versioned.

**Consumer Use of LOMA**

There will be additional security and authentication/authorization needs for direct consumer use, however, as we have no knowledge of the Lender’s existing capabilities in this area, we assume that a ‘consumer ready’ portal is available.

For our purposes, we assume that any user accessing the loan origination forms is properly authenticated and that their authority credentials are available to the LOMA when the request arrives. The ID of the consumer and the user are added to the database to record this information.

This being the case, there is no system difference between internal operation and consumer operation – that is, direct business. In reality, we would expect to see some changes in the decisioning process based on the internal/external status of the user, and also the authority level of the user. This differentiation is accommodated by adding the user’s authority and credentials to the Loan context data and using it within the decision models to guide the decisioning process as required.

**IDIOM Forms**

Please see the section on IDIOM Forms later in this document for further background on this product. For the purpose of LOMA, reflexive IDIOM Forms will be built that will operate under the control of special ‘(Forms) Session’ decision models, while at the same time being tightly bound to the full set of ‘credit decisioning’ decision models. By tightly bound, we mean the rules are fully integrated with the form and are executing potentially field by field. An example of this approach is where the business decision models identify a business error, and the session model turns on the error display to present it immediately.

This will provide the maximum reflexive experience (meaning that the least possible user input is required) for the user, tailored to the user’s individual circumstance.

**Workflow**

A loan will go through multiple stages in its life-cycle, for instance periodic reviews, interest charges and repayments, and possible other ad-hoc changes and interventions.

These changes are triggered by one of three events:

- **DateTime is reached** (e.g. a review date or interest charge date is reached);
✓ Manual intervention (e.g. a payment date has passed without payment);
✓ An external event is raised (e.g. a payment is received)

Sometimes, an automatic bring-up will trigger a manual intervention. All of these circumstances are accommodated by the data design [Figure 2], with the orange entities the focus for this discussion.

Each time the LoanDataXML is processed through the decision engine, the ‘Workflow: Actions, and BringUps Data Model’ (see Figure 1) will run. This data model will inspect the final state of the LoanDataXML and determine if any manual alerts need to be raised and when (for instance, it is being referred and refer processing is required); and/or if any future events need to be recorded as a BringUp for automated processing (for instance, interest is to be charged). Negative BringUps are also used – for instance, if an external event like a payment is due on (say) the 15th, then a BringUp on the 16th may be set to raise an alert Action that a payment is missed.

In this case, if the payment is received, then the LoanDataXML would be updated with a payment transaction, in which case the BringUps are cleared and reset so that the above intervention would not be triggered.

This is done when the LoanDataXML is saved – the write process itself strips out the Actions and BringUps and re-applies them to the database. Unprocessed BringUps are first cleared and replaced ‘in toto’; unassigned Actions are cleared and all new actions also inserted. In this way all life cycle activities can be managed for the full life of any entity.

**Document Generation**

We note that the IDIOM Forms are routinely used to file-pick and upload existing documents via the browser. Monitoring the timing of receipt of these documents (and escalation if they are not received) can be managed by the Workflow as described above.

When received, the documents can be inserted into the database by the Application.

For the generation of loan documentation, we propose the IDIOM Document Generator. This IDIOM product provides a comprehensive capability to generate OpenXML (Word) documents under the control of decision models, to be supplied as Word or PDF documents. It has a purpose built document design tool and database, and requires its own runtime process within the Application.

Using this tool, the Lender is able to generate contract and compliance documents for publication. Naturally, these generated documents will also be stored in the Application database.
MAKING IT WORK WITH IDIOM

IDIOM Tools

As you might expect, IDIOM has been fine-tuning its flagship IDIOM Decision Manager product since its launch in 2001 to fully support the concepts outlined in our series of articles.

In fact, the name of our company is a reflection of the importance of the business ‘idiom’, and of its relationship to both business rules and data. The ‘idiom’ is the proprietary language that is used to describe the essence of any business – it is used to describe its business policies, its decision making, and its business rules. And in so doing, the idiom refers to the data ‘in context’. The challenge lies in mirroring this context within the various executables that implement it.

Defining and automating the ‘idiom’ is the essence of our business and of our IDIOM Decision Manager product.

A transaction often requires dialog with a human actor. A dialog is easily managed by building an IDIOM Form over the transaction data – the same data as is available to the decision models that interpret and maintain it. An IDIOM Form includes the ability to execute IDIOM Decision Models inside the Forms experience on a large scale, field by field if necessary, to ensure a fully reflexive user experience.

Finally, the transaction end state might need to be reported via one or more business documents. The stand-alone IDIOM Document Generator can be used to generate complex business documents under the control of decision models, using only Word authored text and images as design input.

The IDIOM Decision Manager

✓ IDIOM Decision Manager is a tool for graphically modeling, testing, and deploying business decisions - without programming!
✓ A tool for the policy maker, not the programmer.
✓ IDIOM automates complex decision-making at the enterprise level, deployable as industrial strength stand-alone components.
✓ In day-to-day practice it is usually 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 modeled 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 at all times within the palettes.
✓ Testing of the models (or any part thereof) is available at all times within the development palettes themselves; full regression testing (incl. model answer differencing) is available in the adjacent ‘Test Executive’ (not shown).
✓ Deployment as 100% generated software components is fully automated and ‘without fingerprints’.
Example above 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 above) are demonstrably aligned and integrated through shared context - validating and strengthening both.

The data model defines the problem domain at rest; the decision model defines the valid state transitions. Together they completely define the 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 rules details are easily captured using a ‘Lego’ like drag-and-drop development 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 these formulas.

The rules can be tested immediately within the IDIOM Decision Manager palettes.

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 simple interfaces and wrappers (in-line, dll, web service, queue service, many more).

And at the same time, it generates the model into business readable documentation (PDF).

Key Points of Difference

IDIOM’s decision models do for decisions what data models do for data – a powerful abstraction that makes the underlying complexity visible and manageable.

The models allow internal data transformations and business rules to be intermingled within a single transaction. 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 behavior of business rules.

✔ Context is continuously displayed and actively managed.

✔ Decision models that incorporate both data and rules behavior enables 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 correctness, completeness, and consistency of the models.

**Key Point of Innovation**

✔ Fundamental redesign of the traditional SDLC by fully separating the development and automation of business policy from development of the systems that support it.

✔ Use of IDIOM is effective in spawning a ‘Business Policy Development Life Cycle’ that is managed independently of and alongside the traditional System 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 and reduced business risk through business modeling and empirical testing of actual 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 cost, time, and risk through reduced system complexity and clear separation of concerns.

**Further Benefits**

✔ Full auditability of policy changes, and visibility of policy implementation through the graphical models and logical English PDF generation.

✔ The Decision Manager Workbench is available to experiment with and further develop policy independently of the systems development process.

✔ Automated, robust, industrial strength deployment on any scale supported by the ‘transaction processing application’ and its underlying platform.

✔ Simple injection into legacy systems leading to eventual legacy replacement.

**IDIOM Forms**

IDIOM Forms is a tool for generating web2.0 forms that embed the power of IDIOM decision models into the form itself. As the user navigates through an IDIOM Form, the integrated decision models can be executed on an element by element basis to make various business calculations and assessments; and to modify the form’s meta-data. This allows the decision models to dynamically adjust the visible shape and content of the form on an element by element basis.

A single IDIOM Form can be used to process complex business transactions (e.g. an insurance application, a clinical pathway, a loan application) through to closure, including such functions as validation, transaction acceptance, costing or pricing, up-selling, plus determination of subsequent workflow.
The IDIOM Forms Engine (the runtime component of IDIOM Forms) is production hardened after many years use on thousands of servers throughout the NZ health sector. The deployed IDIOM Forms Engine became the basis for the NZ Health Information Standards Organisation (HISO) forms standard and is currently the most compliant under that standard. The Forms Engine is now also widely used in the finance sector for insurance underwriting and claims management globally.

**IDIOM Decision Tracker**

The IDIOM Decision Tracker is a tool to map MS Word and Excel policy documents to IDIOM decision models for full bi-directional traceability between corporate policy definitions in the MS documents and their actual implementation as IDIOM generated decision engines.

**IDIOM Mapper**

The IDIOM Mapper generates ‘simple SQL’ from an XML configuration document, and then executes a full round trip transaction cycle: From database to XML; then rules execution (multiple decision models); and from XML back to the database. All done extremely quickly, either in single transactions or in batch. The mapper process is thread-safe and can execute in many process streams.

**IDIOM Decision Manager Workbench**

The IDIOM Decision Manager Workbench is a user operable platform for running decision models across enterprise databases on a large scale without the need for IT technical support.

- A generic batch processing platform for use by IT and business operations.
- A platform for the auditor, the business policy manager, the product manager, the corporate business strategy manager.
- An audit tool for identifying, reporting, and managing anomalies, errors, and issues of concern in any database.
- A simulation tool for comparing the performance of current and ‘to be’ versions of any automated policy; and a tool that can modify data on the fly to simulate changes in the make-up of in-bound transactions over time
- A data masking tool, able to replicate large scale databases with all personally identifying details masked from its own library of several million fake identities. Relationships between identities in the underlying data can be maintained, notwithstanding the use of fake identities.
- A conversion tool: able to read data from one system in its proprietary format; intelligently transform it through one or more decision models; and then output it to a new system also in its proprietary format. A scale of millions of entities and hundreds of tables is easily supported.

**Examples of Use**

**Superannuation Fund Administrator:** Perform all period end processing, including fee’s, insurances, member adjustments, entitlements, and reporting, for several hundred thousand fund members.
**Insurer:** Compare the current underwriting policy with a proposed underwriting policy across a recent portfolio of 500,000 insurance policies to determine potential changes in the rate of referral and its attendant costs.

**Superannuation Fund Administrator:** Run 100’s of distinct audit tests across 1,000,000 member accounts on a daily basis to independently verify the production data.

**City Council:** Compare this year’s rating policy with next year’s proposed rating policy for each property in a city of more than 500,000 ratepayers to identify outlier changes in the rates actually charged.

**Insurer:** Dynamically modify key attributes of real transactions to simulate changes in the make-up of the in-bound business, and assess the impact of these changes across an entire portfolio.

---

**Enterprise Class Features**

- Full authorization and audit controls down to the field level for all users of the platform.
- Seamless operation across multiple user definable environments, for instance Development, UAT, Simulation, Production.
- High performance, including parallel processing on a large scale across multiple machines, for instance running up to 24 Processes on each i7 class CPU, with multiple CPU’s possible, each logging back to the Workbench for centralized management.

**Measured Performance**
✓ Daily pass of 1,000,000 pension fund members each comprising of a join of >20 complex member related tables for a total of ~one billion rows.

✓ 10’s of Decision Models implementing hundreds of individual member tests.

✓ Alerts captured and reported daily for start of business.

✓ Run daily in 30 minutes using one i7 class processor, with data drawn from an IBM iSeries.

**IDIOM Document Generator**

The IDIOM Document Generator is a tool to collect and collate Word document templates and many associated text fragments (called blurbs), and to assemble these into complete documents under the control of a decision model. The templates and blurbs are checked for style consistency when sourced.

A schema defined XML document is generated by the Document Generator that lists all of the blurbs required for a document, and any variables highlighted within them. The actual insertion of each blurb is then triggered by a decision model on a case by case basis.

The decision model also populates all of the variables named in the text, so that large and complex documents can be generated automatically under the control of decision models to reflect the exact circumstances of the underlying business record.

The Document Generator also allows for insertion of images and for calling bespoke programs to insert additional text or images.

The Document Generator only requires standard Word authoring skills, and IDIOM Decision Manager.
Finis.

For further information or discussion regarding this proposal, please contact the author:

**Mark Norton** | Executive Director | **IDIOM Limited** | Microsoft GOLD Business Partner |
Office +64 9 630 8950 | Mob +64 21 434669 | After Hrs +64 9 817 7165 |
Aust. Free Call 1 800 049 004 | PO Box 60101, Titirangi 0642 | New Zealand |
Suite 8 Level 1 93 Dominion Road, Mt Eden, Auckland 1024 | New Zealand |
Email mark.norton@idiomsoftware.com | Skype Mark.Norton |

www.idiomsoftware.com | Pure Decisioning