Enterprise Systems Migration

IDIOM White Paper

☎ +61 2 9968 4168 | idiomsoftware.com |
✉ Suite 16, Level 2, 357 Military Road, Mosman NSW 2088 Australia |

Table of Contents

INTRODUCTION

Migration Challenges

This paper presents IDIOM's solution to the challenge of migrating enterprise scale systems, particularly in regulated environments like insurance and banking, but extending to any large-scale migration from one or more legacy systems to one or more future systems.

There are numerous challenges to large scale migration. The failure rate of migrations is legendary, despite vast amounts of money being thrown at the problem. Here are some of the issues:

- Data needs to be transformed and normalised through many stages from legacy to future system including:
  - Basic Encoding – EBCDIC, ASCII, big-endian, little-endian
  - Character Representation – converting bit switches, binary, and hexadecimal representations to character equivalents
  - Datatypes – legacy to SQL to XML (all different)
  - Standardization of Logical Representations – standard use of integer vs character vs decimal both within and across legacy systems; consistent and appropriate use of decimal precision; using one representation for Boolean, for example true/false, instead of yes/no, y/n, on/off, 1/0, present/not present, etc.
  - Common Coding – transforming legacy enumerations into future state enumerations, including many-to-many enumeration conversions.
- Many-to-many migrations: Multiple legacy systems may need to be match/merged into one target system; each legacy system may need its portfolio to be split into multiple target systems (for example, to separately manage on-sale and off-sale business).
- All material calculations in each legacy system need to be (re)discovered and extracted in exact detail, and then recreated, and validated in the target system(s). The provenance of these calculations is often lost, leading to uncertainty, errors, and omissions that can be difficult to resolve from first principles, which may themselves be unclear for historical obligations.
- Correcting critical legacy data issues can create challenging process management problems – do we correct in the legacy system, during the migration, or in the future system?
- Duplicate customer records both within and across legacy systems should be identified and resolved.
- Product definitions need to be rationalised and migrated in a way that is consistent with the underlying business entity data, including all calculations as described above, which is itself undergoing substantial and parallel transformations.

Most approaches that we find in market limit their focus to data only. This approach is fatal and delegates the real work to an enormous and poorly understood programming task, which is then often made worse by considering EVTL (extract, validate, transform, load) to be a single process.

IDIOM refers to single process migration as complexity squared! Each layer of complexity in each legacy system is compounded by each layer of complexity in the target system(s).

Instead, IDIOM proposes use of a 'state-full' bridge that provides a resting point between legacy and target systems. The IDIOM Migration Bridge [the Bridge] can be described as follows:

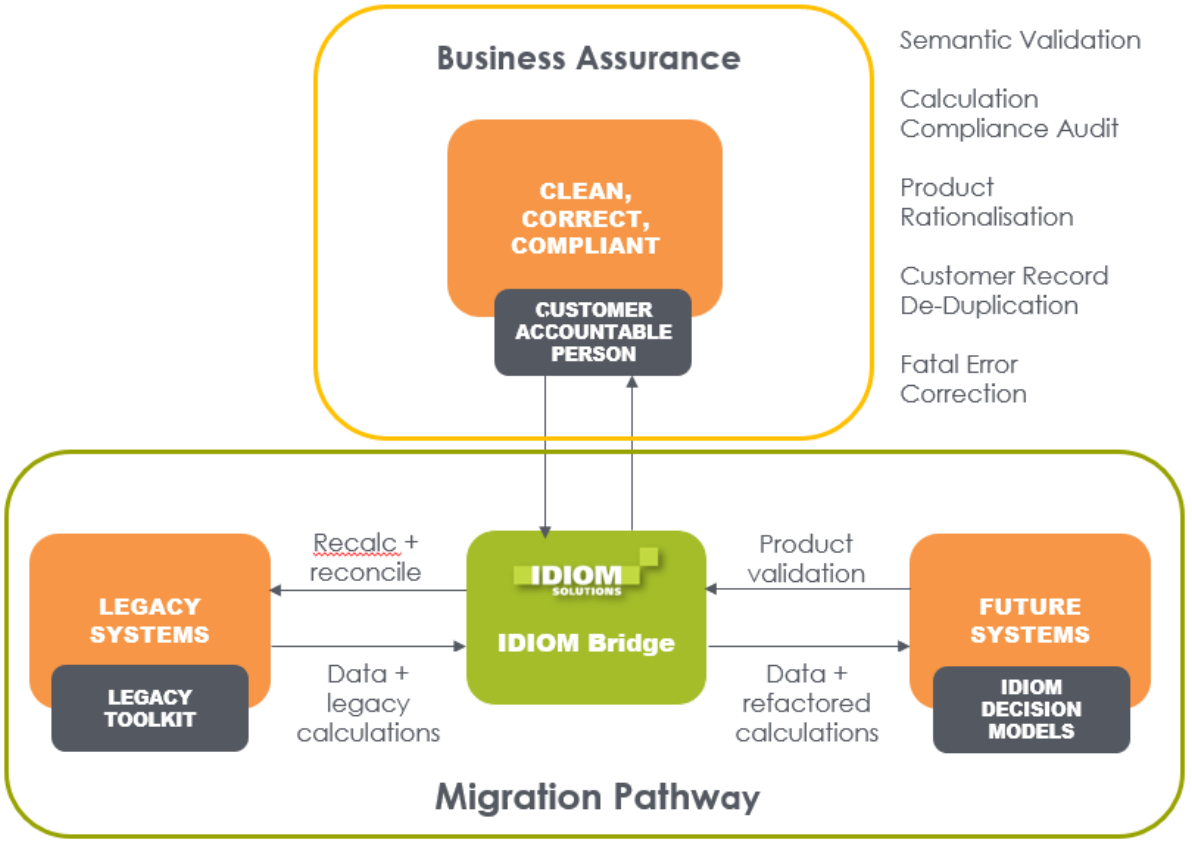

Figure 1

1. **Complete Solution:** The IDIOM Bridge provides a state-full and function rich bridging platform to support value-added migration pathways from 'any/many (legacy) systems' to 'any/many (target) systems'.
2. **Any Legacy System:** With support from our partner PreView (Australia), we have the tools and skills to extract data and calculations from any legacy system into the Bridge; then using IDIOM's proprietary tools and techniques, we can add value to the data in the Bridge, and thence from the Bridge to any target systems for which the Bridge has been configured with product and transactional data definitions. These legacy skills are available on a world-wide basis.
3. **Avoid complexity²** squared, which is our description of the complexity that arises when we try to manage the intricacies of both the legacy and target systems inside one process. In the IDIOM process, each legacy system's extraction and validation steps are constructed and verified independently of all other systems and steps; similarly, each target system's transform and load steps are constructed and verified independently of

all other systems and steps. For the sake of clarity, extract and validate are explicitly disconnected from transform and load, with the bridge providing a safe harbor between these already complex processes.

4. **Significant Value Add within the Bridge:** The bridge provides a valuable and intelligent 'safe harbour' for data and calculations, providing many features to improve and assure data and calculation quality prior to the throw to one or more systems. Value adding features include: semantic validation of data; reconstruction and validation of calculations; de-duplication of customer records; remediation process for fatally bad data; and product rationalisation.
5. **Complete Assurance:** Conversion to the target system's product definition and transactional data formats is built in the Bridge and the results validated against prior images (from both the legacy system and the Bridge itself) to provide complete assurance before the final throw to target systems.
6. **One Step Final Throw:** The final throw executes all steps from each legacy system to target system(s) in one contiguous stream, guaranteed to run between one close-of-business and the next start-of-business over a weekend.

The Bridge described above is not conceptual - it has been built to support the approach outlined herein, and an exemplar migration is available in a demonstration application to make visible the many functions required for a successful migration.

Background and Experience

With our partners Preview, who supply and manage the cohort of specialist systems 'archaeologists', IDIOM's legacy systems knowledge and capabilities currently spans the following systems:

- Life risk / wealth includes CLOAS, Capsil, Life70, Compass, UNISURE, Calibre, SuperB, SAS, Life400, LIFE–MVS, Talisman, Integral & Sonata
- P&C and R/I includes Polisy, COGEN, Polisy400, Huon / TIG, Sirius / PURE, Insure90, Guidewire, SAP & Duck Creek
- Health includes WHICS, HAMBS plus SAP, Amicus, TriZetto & Oracle

The primary data extraction, transformation, and data-type validation is achieved using tools developed by Preview's 'archaeologists' over a period of decades. These tools are industrial strength, agile, and high performance. The tools are used by skilled specialists to deliver the legacy data into the Bridge as XML Schema defined records, one per business entity being migrated.

The IDIOM Migration Bridge itself is built using IDIOM's proven, industrial strength tools, including the IDIOM Decision Manager and the IDIOM Mapper inside the full-cycle, highly scalable applications framework provided by the IDIOM Transaction Engine

The logic requirements generated by semantic validation and replication of legacy calculations fit easily within the operating parameters of the IDIOM Decision Manager, and the Mapper and Transaction Engine are comfortable dealing with migrations that span thousands of tables and millions of migrating entities (customers, accounts, policies, etc).

The IDIOM Decision Manager also plays important roles in Product Rationalisation, Customer Record De-Duplication, and Fatal Error Correction as noted in the above diagram. These optional migration tasks also have specific programmed support inside the Bridge application.

Why Do We Need the Bridge?

The fundamental problem is complexity. Most Clients agree that their legacy platforms are complex. Most also consider that their target systems are complex.

To simultaneously unravel one complex system and refactor it into another unlike and also complex system, in a single logical process compounds the complexity. This complexity exists in the data and its relationships; in the calculations; and in the interplay between calculations and data. All need to be unraveled and refactored together as a single cohesive operational unit.

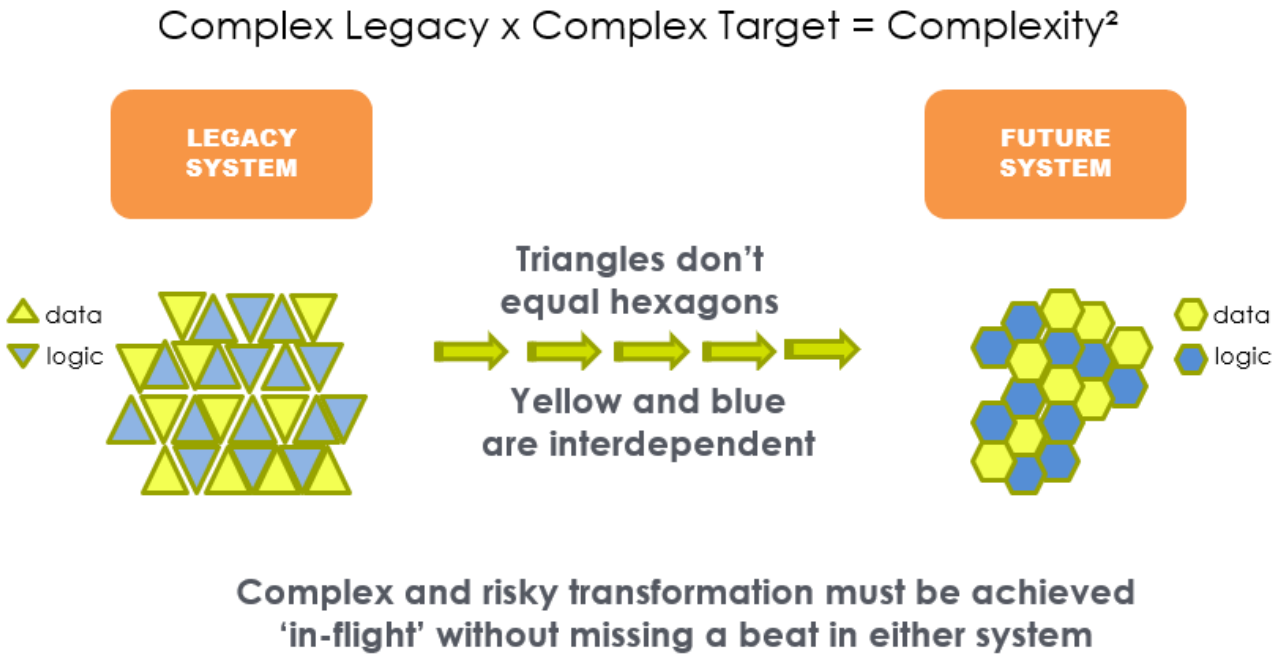

Figure 2

The IDIOM approach is to break the single logical process inferred above into simpler processes, which are run back-to-back for the final throw. And the IDIOM tools separately address the data and the calculations, reducing each to generic intermediate forms in the Bridge.

The initial migration of data and calculations from the legacy system to the Bridge mirrors only the legacy constraints, without consideration of the target system. The throw from the Bridge to the future system is similarly unconstrained by legacy system considerations.

The issue of complexity is resolved in the Bridge itself. With both data and calculations safely contained in a modern, industry standard generic format, we can validate, refactor, and

simplify products and their calculations under the safety net of entry and exit 'control points'. These control points provide assurance that the Bridge contents match the legacy system both before and after refactoring, and they provide a benchmark for future systems use or assurance following the final throw.

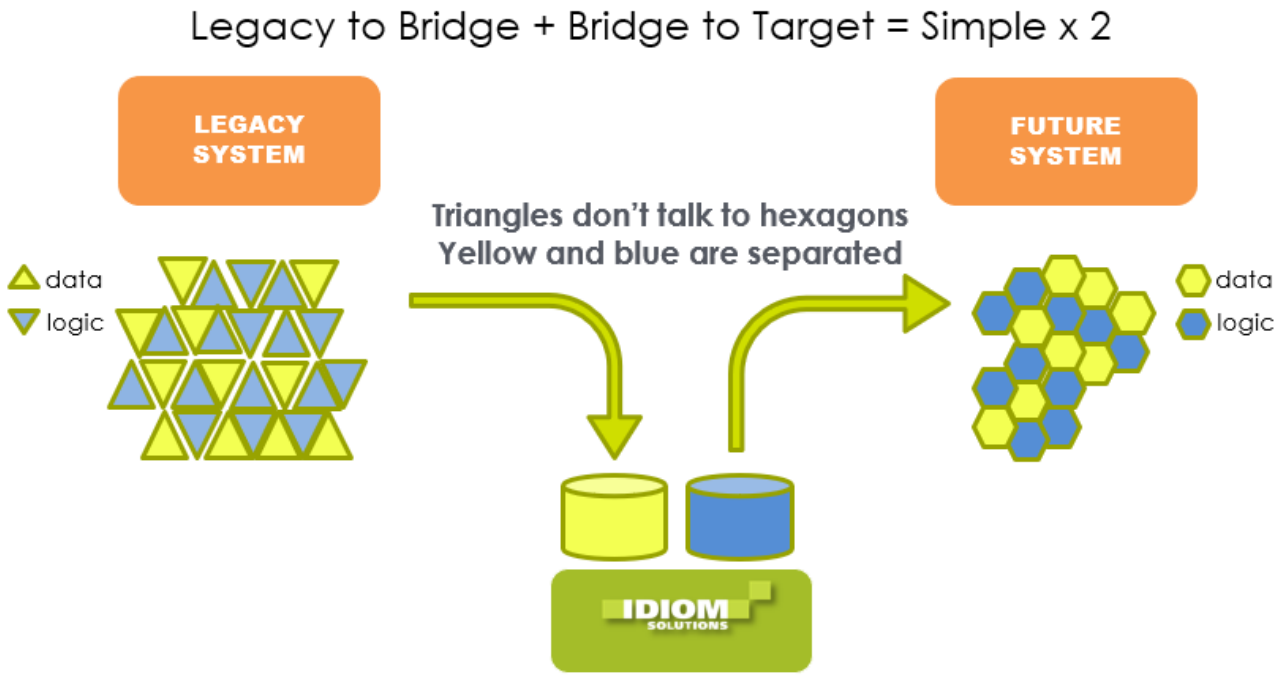

Figure 3

Risk Control

To systematically control risk and assure end-to-end traceability, IDIOM offers a generic framework of five ***Assurance Levels*** that are described later in this paper.

The Bridge consolidates the data and calculations. The Bridge can provide both backward assurance to the legacy system(s) and forward assurance to any future system(s).

The pivotal assurance - and also the most complex - is to be assured that data can be re-used by a new system and get the same result. This is assured by our *Assurance Level Three*, which implements *Calculations Cleansing*. This is more intricate and complex than *Data Cleansing*.

IDIOM's lowest level of assurance, Level One, assures that the legacy data is correctly formatted and compliant with a data dictionary – that is to say, "named data".

IDIOM's experience confirms that named data remains ambiguous unless the processes using it are fully understood; data that only complies with *Assurance Level One* will behave unpredictably once ingested by a different system. The receiving system will face significant obstacles in identifying the cause of anomalies, which requires full traceability and concurrent access to legacy system code.

APRA Requirement to Consider Fitness-for-Use

The Australian Prudential Regulation Authority's [APRA] Prudential Practice Guide CPG235 'Managing Data Risk' places emphasis on the need for data to have 'fitness for use' before acceptance. Clause 51states:

> 51. *Data validation is the assessment of the data against business rules to determine its fitness for use prior to further processing. It constitutes a key set of controls for ensuring that data meets quality requirements*. [**APRA CPG235**]

IDIOM's *Assurance Levels Two and Three* provide Validation and Calculations Cleansing to satisfy CPG235, and more generally induce early detection of issues to do with data and their related calculations. We do not believe that Data Cleansing alone satisfies CPG-235. Further information on IDIOM's approach to managing CPG-235 can be found in the IDIOM whitepaper 'Data Integrity in Financial Services' available here[1]. While APRA is an Australian regulator, the focus of CPG235 is relevant worldwide.

Glossary

Accountable Person means the insurer approved business person or entity that is charged with sanctioning the **Approved Formulas**. IDIOM anticipates that this includes product and process owners, actuaries, tax specialists, and such like.

Alert means a record created by the IDIOM Decision Manager™ that signals a breach of a validation rule (Assurance Level Two), or that a **Controlled Outcome** does not comply with its **Business Policy** (Assurance Level Three).

Assurance Level One is satisfied when the data is valid and complies with a meta dictionary (e.g. a sum insured is a positive decimal amount).

Assurance Level Two is satisfied when the data is valid and also consistent with other data and meta data (e.g. a premium charge can be valid but fails consistency if the policy status – which is a separate datum – is 'out of force').

Assurance Level Three is satisfied when the calculation of critical **Derived Data** is replicable.

Assurance Level Four is satisfied when each customer is represented by one and only one customer entity record in the Bridge, regardless of the number of legacy systems they may exist on.

Assurance Level Five is associated with product and process rationalisation. It ensures that changes that are made to the data after Level Three to support a simplified and rationalised set of data, processes, and products, has not lost any meaning.

Bridge means the intermediate platform which contains (i) customers, policies, claims, and/or other context data, (ii) business/product configuration and reference data, and (iii)

[1] http://idiomsoftware.com/DOCS/Download/ce5e30f0-e6de-4b77-bfe3-766c28f395fc.pdf

the Formulas (i.e. the calculation logic defined within the IDIOM Decision Manager) in the form of a Calculation Engine that enables system outcomes to be recreated and assured.

Business Policy means the business defined and approved set of methods, algorithms, and constraints that govern the calculation of **Controlled Outcomes**.

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

Calculation Engine is an IDIOM term that refers to an IDIOM supplied executable that is used to assimilate and execute any number of Decision Models. The scale and utility of the IDIOM Calculation Engine can extend to include the entire population of calculations for most organisations.

Cleansing means:

- **Data Cleansing** means the data validation achieved via assurance levels one and two, which assures that the existing data is valid and semantically consistent.
- **Calculations Cleansing** means the outcome of assurance level three, which assures that derived data is correctly understood and reproducible.
- **Customer Cleansing** means the outcome of assurance level four, which assures that customers are represented by one and only one record.
- **Product Cleansing** means rationalising the business products, data, and processes without loss of business or customer value.

Client means a party who has engaged IDIOM to assist with a migration. In this Whitepaper, 'Customer' is used to refer to the customers of the Client.

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 a 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 change of state that is controlled by the calculation.

Controlled Outcome is an IDIOM term that refers to a derived datum that the business declares has critical importance. It should include derived data that is essential to the business mission, and all derived outputs that are prescribed by regulation or published obligations. Reconciliation of the set of Controlled Outcomes at various stages of the migration provides the business with end to end assurance. A Controlled Outcome is assigned to/by the Accountable Person, who is the final arbiter of the **Formula** that derives the outcome value. The existence of a Controlled Outcome is enumerated in a register of Controlled Outcomes.

Decision (an IDIOM term) means the result of a Formula (aka a calculation) 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." Because a Decision is produced by a calculation, the terms can often be used interchangeably.

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

Derived Data means data that has its value derived through calculation. Most critical system data is derived data, although not all derived data is critical.

Entity, EntityXML means an XML record that complies with the relevant schema that defines the type of entity that the Entity describes. In business terms it may be a customer, a policy, a loan, a claim, or any other primary business entity that requires migration. See also Context Data, Transaction.

Formula means the specification of the logic, also known as an algorithm, that implements the **Business Policy** governing the derivation of a **Controlled Outcome**. Formulas are a reflection of and a proxy for the Business Policy that governs the outcome value. When defined in IDIOM Decision Manager, formulas are executable across the full extent of data within the Bridge. A Formula is 'as-built' when it represents a legacy system, and 'approved' when it has been sanctioned for use in a future system. If there is any variation between as-built and approved versions, then appropriate compliance documentation will be created by the Accountable Person for audit and traceability.

- **As-Built Formula** means a faithful transcription of the Formula that is derived from the existing legacy system.
- **Approved Formula** means a Formula that has been approved by the Accountable Person(s) and is the approved version of the Formula for all future derivations of the relevant Controlled Outcome. Only Approved Formulas are used in the Product Rationalisation process.

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, and is used to codify and test the business logic that implements business policy (for instance, product rules). Further information about the IDIOM Decision Manager™ and other IDIOM products can be found in the document 'IDIOM Transaction Engine: An Application to Manage Complex Business Entities' available here[2].

Schema means an XML document that complies with the W3C schema definition (https://www.w3.org/XML/Schema). A schema is used to define XML Records, and to provide enhanced meta data that describes them.

System means a policy administration system, claim system, commissions system, reinsurance system, or any other system that the context requires, for insurance, wealth, or superannuation. It may be either a legacy or a future system as the context requires.

Transaction, Transaction Entity, Transaction EntityXML means an XML record that complies with the relevant schema that defines the type of entity that the Transaction Entity describes. See also Context Data, Entity.

[2] http://www.idiomsoftware.com/DOCS/Download/e07606ea-5cd1-460b-bf64-569270cc03bd.pdf

THE BRIDGE

The key to successful migration is the IDIOM Bridge, which provides a safe-harbor between the potentially 40 plus year old technology used by legacy systems and their modern counterparts.

The Bridge is a complete and stand-alone system that is comprised primarily of a generic database and a calculation engine that is delivered inside a highly scalable lightweight application framework in a Virtual Machine (for use either on premise or in the cloud). It is created per Client engagement to ingest and normalize the legacy system data and processes.

The Bridge is 'state-full', meaning that it has persistent data and process in its own right. The layers of 'state' that reside in the Bridge database include:

- Converted legacy data
- Data standardization and normalization rules
- Semantic data validation rules
- Product configurations and essential reference data
- Calculations for all Controlled Outcomes
- 'Matching Customer' matrices
- Data corrections
- Future state data

The various layers are built up incrementally more or less in the order shown above. As each layer is populated (not necessarily completed), it can be executed and tested in the Bridge. This iterative development process is likely to take months per legacy system.

Note that the Bridge application considers the IDIOM Decision Models to be 'data' (they actually reside in SQL tables) and as such, they hold state. In fact, the majority of knowledge in the Bridge will be contained in Decision Models, which can be applied quickly and accurately to the data as it flows through from legacy to future state in the 'Production Mainline' (see following diagram).

It is important to note that all tools used in the process to be described herein are current, fully functional, and self-contained tools that can perform their respective tasks in isolation. The point of this comment is that fast, iterative, and very agile development processes will be undertaken by specialists outside of the Bridge to develop and prove the various components prior to committing them to the Bridge. The Bridge is not the build platform, it is a platform to collate these rapidly evolving components and to prove them in a single contiguous process. When all layers have been completed and tested to satisfaction, the entire process can be run in one contiguous stream shown as the 'Production Mainline', which is the ultimate objective of the Bridge.

In the following diagram, the IDIOM icons are used to indicate process steps that are codified exclusively in IDIOM Decision Models. The 'head' icon indicates a manual process with Decision Model support.

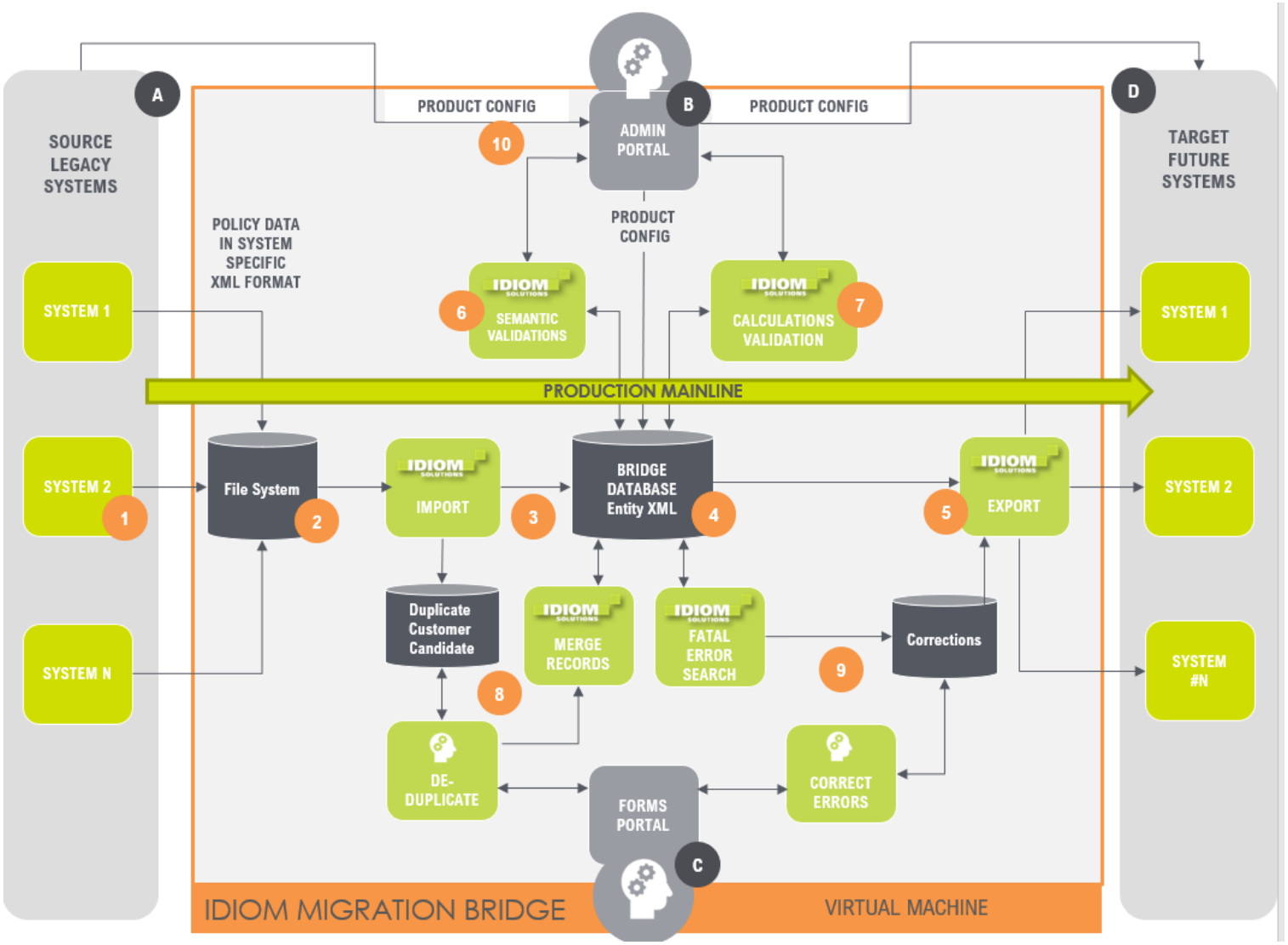

Figure 4

The Bridge has 4 major functional components that are user driven, shown as A, B, C, D (all in black). These are discussed first.

User Managed Processes

A: Data Extraction and Conversion

This phase is technically driven (c.f. business driven) and occurs for each legacy system. It requires skilled systems 'archaeologists' to extract and convert the data and to process it through a series of steps until it is in a consistent, normalised, schema defined format, with one XML record per source entity sitting in the file system of the Bridge.

An entity in this context is as defined by the relevant schema. In business terms it may be a customer, a policy, a loan, a claim, or any other primary business entity that requires migration.

Note that this process usually requires physical changes in the data representation, including some or all of:

- ✓ EBCDIC to ASCII
- ✓ big-endian/little-endian
- ✓ binary and hexadecimal representations to character strings
- ✓ and changes to comply with new XML datatypes.

This process is automated using IDIOM's proprietary extraction toolset (which is not shown in the Bridge diagram). This toolset is proven over many migrations and when used by skilled specialists (the archaeologists) is able to quickly acquire and transform legacy data from most legacy technologies.

As part of this process an intermediate SQL database might be used to provide visibility of data, metrics, and to allow some early data standardisation.

B: Accessing the Bridge Functionality

The IDIOM Transaction Engine 'Admin Portal' is used to provide authenticated and authorised access to the standard Bridge functions, including:

- Load Decision Models from the IDIOM Decision Manager development environment
- Execute batch jobs including:
  - Import
  - Run semantic validations
  - Run calculations
  - Export
  - Others as required
- Access the IDIOM Forms functions [refer C next]

C: IDIOM Forms Functions

There are two standard forms functions:

- Inspect duplicate records and resolve
- Correct fatal errors

Other inspection and calculation functions can be easily added as required.

D: Load the Future Systems

Loading the future systems is dependent on the future systems platform and is usually a handover point to either the Client and/or the Client's implementation partner for each future system.

IDIOM will prepare the data in accordance with schemas provided for each future system and can supply calculation logic for injection into the future system, however the actual load of data and injection of logic is likely to be performed by the Client and/or partner.

IDIOM recommends following up with a subsequent reapplication of the calculation logic externally to the future system, to provide an independent audit that the migration is complete, correct, and compliant.

IDIOM Bridge Functionality

Refer Fig 4 above.

Step 1: Prepare Data

The IDIOM 'archaeologists' are responsible for locating and pre-processing the legacy data, system by system. Multiple systems can be processed in parallel, subject only to the availability of resources.

Step 2: File System

The standard file system of the Bridge is used as a data store for the substantial XML records that are produced by the step 1 extraction and pre-processing. For the sake of clarity, the XML records in the file system will contain ALL data that describes the context entity, with one record per entity. Millions of entities can be managed, with different classes or types of entities located in different folders. Each entity record will comfortably contain thousands of data fields per XML record (the practical upper limit is in the millions of fields).

Customer data will always form one entity type, however it is plausible that some major entities below customer may be managed as distinct types, for instance, policies, loans, claims, etc. Each type of entity will be described by a single schema, to which all records of that type will comply.

At this point the data is still recognisable as being from its legacy source i.e. naming conventions will reflect the legacy system data names to assist with backwards reconciliation.

A reconciliation with the source legacy system is included as part of the handover between the archaeologists and the Bridge.

Step 3: Import

The Import migrates the in-bound, legacy specific XML data into a standard schema defined format that is legacy system agnostic. This standard format may be ACORD, or a preferred proprietary format this is more closely aligned with the targeted future systems.

This transformation is performed by IDIOM Decision Models on an entity by entity basis.

The transformation may be substantial, but each record will exit the process with the same information content as per the in-bound record.

Transformations may include:

- ✓ Standardise enumerations from multiple source systems and source fields as required.
- ✓ Standardise Boolean representations to true/false only.
- ✓ Normalise the data if not already so.
- ✓ Map fields to new locations and formats.
- ✓ Split and merge field values as may be required.

When the Import is complete each record will generally share a common schema definition regardless of its source legacy system, and regardless of its future target system. This is a

preference rather than a technical constraint, so that multiple intermediate schemas could be used.

The new standardised XML is written in to the Bridge database using the ['Source System| |Source System Identifier'] as the key. For the sake of clarity, this important feature means that the import can be rerun as frequently as required without sacrificing any value that has already been added to the Bridge – specifically, customer de-duplication, and fatal error correction (both to be discussed later).

At the same time, the Decision Model will fabricate a Duplicate Customer Candidate key value according to the requirements of the Client. This key value will provide the entry point to identify duplicate candidates. Note that links to third-party duplicate search are planned.

Step 4: The Bridge Database

The Bridge database extends the standard IDIOM Transaction Engine database. It stores each XML Entity record in a single column in the Transaction Entity table.

All subsequent processing will be based around this record. As noted in Step 3, the XML content for each entity can be refreshed as required without sacrificing other state values that are supporting the XML.

The De-Duplication process described in Step 8 will result in merging of some records. When this occurs, the parent merged record is processed and the child records that contributed to the merge are ignored (but not deleted).

If the child record is subsequently updated, it will be remerged on to the parent. If the parent is deleted, the child will become a primary record again.

Semantic Errors will be notified in the database as Alerts and the Duplicate Customer Candidate and Corrections data stores will be logical extensions of it.

Step 5: Export

The Export will run Decision Models to transform the Bridge neutral format data into the schema required for the future target system(s). The actual format and mechanism for the import into the future system determines how this process is managed, and so the specification of the final 'throw' process is not included in this outline.

Step 6: Semantic Validation

Semantic Validation is a batch process that is run on demand from the Admin Portal. Analysis of the results can be visualised using Microsoft's PowerBI.

IDIOM promotes the concept of semantic[3] assurance, which in this case means that the data is understood and assured in terms of its use in systems.

[3] Semantic Defn: Relating to meaning in language or logic

Semantic assurance requires that the data complies with semantic constraints. Most data are subject to constraint rules that can only be assessed by viewing the data in context using a logic engine.

For instance, a start date cannot be after the end date if both dates share the same context (e.g. a policy); each date may be technically valid when viewed individually (i.e. its representation, value range, and location comply with its data dictionary), however the presence of the other date imposes a validity constraint that must be obeyed. The relationship between the two dates cannot be described using any existing data definition language or approach and cannot be captured using a data dictionary – the relationship is a mutual constraint on allowed values that is context dependent. Any given field may be subject to many such constraints at the same time (e.g. the start date also cannot precede the birth-date of the life insured, etc).

The context is an important concept and recognises that many values may affect the subject value being validated. Before Semantic Validation is performed, any existing Corrections are applied to the XML data by the Decision Models. This means that validation is a progressive process that takes into account any corrections which may have been made within the Bridge.

Semantic validation is performed by IDIOM Decision Models and generates 'Alerts' that are inserted into the database for subsequent analysis and follow up.

Whenever Semantic Validation is executed, all prior Alerts are deleted and then recreated as required.

Step 7: Calculations Validation

Calculations Validation is a batch process that is run on demand from the Admin Portal. Analysis of the results can be visualised using Microsoft's PowerBI.

Calculations validation uses a specific complex-type in the entity schema that allows additional meta data to be recorded for any value that is subject to calculations validation (i.e. it is a Controlled Outcome), including the new derived value and an assessment of the match.

During the process, the calculation will calculate a new value for the controlled outcome and assess the match. An Alert may be generated by the decision model for notification via the database in the event of no match, or no match within certain parameters.

Step 8: De Duplication

The IDIOM process supports de-duplication of customer data.

The IDIOM rules are used to fabricate an initial key for duplicate matching.

Any duplicate candidates, which are records sharing the same match key, are available for inspection via the IDIOM Forms Portal. The Form will present selected match data in addition to the key for manual determination of a match. If the user determines that a match exists, IDIOM rules will perform a controlled merge of matched records.

Un-merging is supported by simply deleting the parent merged record and the links to its children.

Step 9: Validation Corrections

Validation Corrections are persisted changes to the transformed source record.

The scope of allowed Corrections is determined by an XML schema that includes an element that describes a subset of the relevant EntityXML (described in the following schema diagram as the 'TransactionEntityXMLSnippet'). The snippet will only include those (relatively few) nodes that are deemed to be both necessary and available for update by users during the migration period.

Entities that require correction will be will be flagged with an Alert by the Semantic Validation rules. The list of outstanding fatal errors will be available to operators using the Admin Portal. Correction of the nodes in error will be achieved using an IDIOM Form built over the full schema, with only the schema nodes in the snippet being input-enabled to allow correction.

When the user submits the form:

- ✓ The form's XML is 'differenced' by rules to identify the changed elements in the corrections snippet; these elements are copied to the Snippet.
- ✓ The corrected new errors (list) are moved to the corrected errors element (list).
- ✓ The Corrections record is updated.

The XML that complies with this schema is persisted in the Corrections table as an extension to the standard Entity XML. Any time the entity XML is subsequently retrieved, the Correction is reapplied (if present).

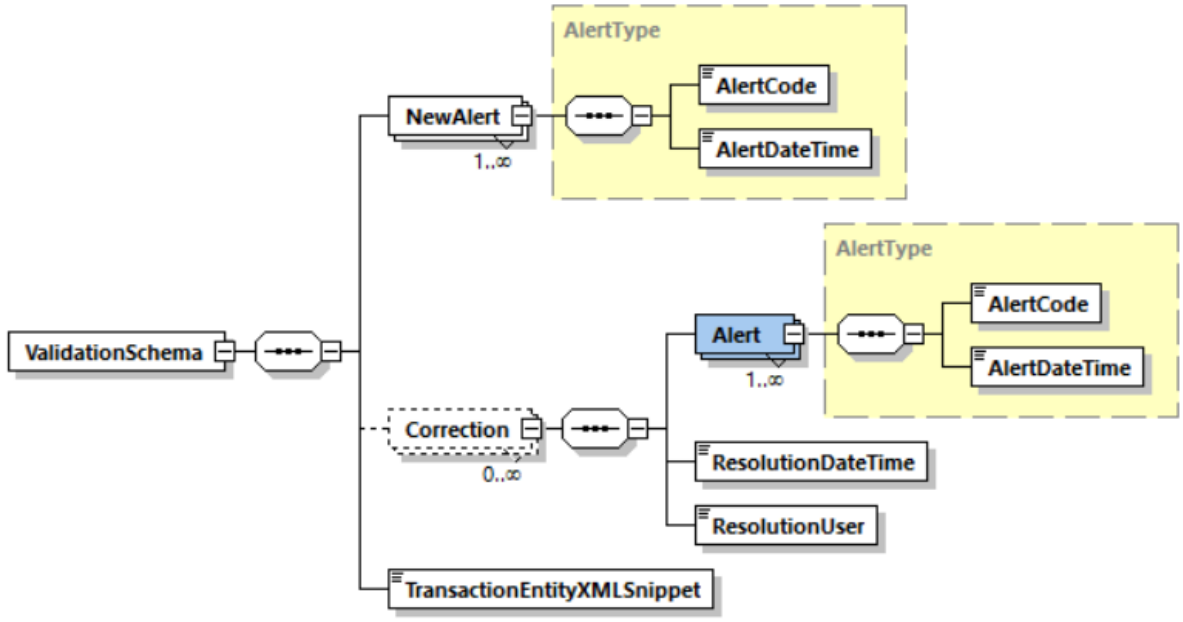

Figure 5

Step 10: Product Rationalisation

Product rationalisation requires manual reconstruction of the product configuration documents using the pathways described as 'Product Config' in the diagram. It is likely that product configuration changes will require changes in the Controlled Outcome calculations, and potentially the schema that defines the EntityXML. Both will need to be maintained in parallel with any changes to the Product Configuration.

Note that Calculations Validation must postdate the Product Configuration change, so that Product rationalisation via product configuration changes is inherently validated by the validation process. At this stage, we do not contemplate running multiple product rationalisation scenarios in parallel, however, a continuous change process can be supported to allow ongoing refinement of the preferred Product Rationalisation strategy.

THE IDIOM APPROACH

The IDIOM approach to data migration is multi-dimensional:

- The primary objective is to mitigate risk by fully understanding the data and how it is used in (possibly multiple) legacy system's and to simplify this via four distinct cleansing steps:
  - data cleansing – to ensure the data is useable
  - calculations cleansing – to ensure the system behaviour in response to the data is understood and replicable
  - customer cleansing – to identify and resolve duplicate customer records.
  - product cleansing – to rationalise products, data, and processes for a simpler end state in a future system(s).
- These are assured by five levels of assurance.
- The process of execution has three distinct phases:
  - abstraction of legacy assets - which is technically driven
  - business reconciliation and normalisation – which is technically and business driven
  - the migration itself – which is relatively mechanical if the above is already completed.

The Major Phases for Delivery

Phase 1: Technical Abstraction

This phase is technical and occurs for each legacy system. It is not a requirement of this phase for the business to clarify, correct or align business rules (which happens in phase 2).

The objective is to obtain and create an abstracted view of the legacy system that includes 100% of the data, and the calculations as described by the Controlled Outcomes. Recalculation of the Controlled Outcomes using 'As-Built Formulas' provides assurance that the extracted data is fit for purpose.

The data is subject to assurance levels one, two and three as described in the next section before the abstracted model is considered for advancement to the next phase.

At the conclusion of the Abstraction Phase the legacy system is fully described and proven in the abstracted model, which can now be used without recourse to the legacy system.

Phase 2a: Customer De-Duplication and Fatal Error Correction

This phase is business driven. The primary objective is to:

- ✓ Normalise the customer records, and
- ✓ To update any relevant data to meet minimum required standards.

This phase is entirely business driven using the Bridge as an application platform and can occur in parallel with Phase 2b (next).

Phase 2b: Business Reconciliation and Normalisation

This phase is business driven. The primary objective is to normalise and approve the calculations, the data, and ultimately, the products themselves into the smallest number of components that can correctly recreate all Controlled Outcomes and deliver the required customer commitments.

Before either data or products can be normalised, we must normalise the calculations, which ultimately determine the meaning of the data (the meaning of the data is the key to the data normalisation process).

Therefore, the first objective of this phase is to get a business validated and approved set of Formulas (see Glossary). Given the importance of these calculations to the subsequent processes and to the business itself, this approval must be authoritative, hence the concept of the Accountable Person (see Glossary).

The Accountable Person must formally approve every Formula that progresses into the Bridge as an Approved Formula. A Compliance Report provides input to this process. Compliant outcomes can be passed with some confidence. Non-compliant outcomes will require some compliance related action on the part of the Accountable Person – to modify the Formula, to accept that the Formula is in error, or some other corrective or mitigating action.

When the Formulas deriving the Controlled Outcomes have been approved, IDIOM and/or insurer technical staff can normalise the calculations and then the data.

At the same time, the Accountable Person can be engaged in developing preferred product rationalisation strategies.

This process will deliver the preferred candidate product taxonomy. A product taxonomy is a division of products between and within future System. While a single future system is plausible, there is often a requirement to accommodate more than one future system, in which case each future system will have its own product structure supported by Product Configurations.

The Bridge contains a set of Product Configurations and a matching set of Data (in XML form). Each future system is likely to require a different set of configurations, and by extension there will be changes to both data and Formulas.

This phase concludes when there is exactly one candidate product taxonomy per future system, and the data and Formulas have been aligned with that taxonomy.

Phase 3: Migration to new System

The prior phases provide the necessary methods and controls to allow this phase to be relatively mechanical, albeit with strong dependency on the future system vendor, and the Client and/or their implementation partner. The data is again transformed to match the future system schema, and the data prepared and thrown.

Following ingestion by the new system, there should be a final rules-managed verification to ensure that the throw is complete, consistent, and correct.

The Assurance Levels

Table 1

LEVEL 1 – DATA FORMAT CLEANSING	LEVEL 2 – DATA CLEANSING	LEVEL 3 – CALCULATION CLEANSING	LEVEL 4 – CUSTOMER CLEANSING	LEVEL 5 – PRODUCT CLEANSING
Extract and Clean ***Data is Dictionary Defined and Compliant with that Dictionary***	**Semantic Validation** ***Data is Semantically Validated against Source System Constraints***	**Assess for Compliance** ***Derived Data is Validated against Source System equivalents***	**De-Duplicate** ***De-Duplicate the Customer Entities; one Customer = one Entity XML***	**Rationalize and Simplify** ***Rationalise Products and Calculations and Remove Source System Constraints***
Data complies with XML Schema definitions	Data values are tested against other data values using business rules inferred from the legacy system	Key values are recalculated and measured against existing system values	Customer demographic data is standardized and analyzed to identify duplicate customers, which are resolved	Data and calculations are transformed, merged, standardized, normalized, and simplified to meet future needs
Technically led, no business input required	Technically led, no business input required	Technically led, business signoff required	Business Led – including user approval of customer record merges	Business Led – including policy decisions to alter products and processes

IDIOM promotes 5 levels of assurance for a complete migration. The IDIOM assurance levels are designed to provide a quantitative and qualitative improvement in data quality as described by each respective Assurance Level in the adjacent table [Table 1]. A notional 6th level of Assurance occurs when the same levels of Assurance can be verified in the Future System(s).

Calculation Engine

Semantic validation can only be achieved programmatically using logic[4], which is why the IDIOM Decision Manager™ plays a central role in the IDIOM data migration solution.

Building the logic requires us to work at a meta level, using the names and definitions of the data to build the semantic assurance 'idiom[5]' – this idiom is a set of rules that describe the constraints and calculations that determine the correctness of the data in the fields that are described by the dictionary.

The Assurance Level Two and Level Three assessments require development and deployment of a calculation engine to evaluate the transaction data. The IDIOM Decision Manager™ is a graphical modelling tool that allows SME's to graphically model and test all required logic; it then generates high performance, native code implementations of the logic that can be executed at scale to evaluate all transactional data, both pre and post migration.

When the rules are executed, they generate new data outcomes (which IDIOM calls 'Decision's, hence IDIOM Decision Manager). These outcomes include new calculated values, and Boolean interpretations of the comparisons of the values. Ultimately, when a breach of any rule is identified, the Decision Models also create an Alert for that breach. An Alert is a specific decision outcome that identifies which rule was breached by the transaction in focus.

This new 'hard data' is captured and stored in the Bridge, from which assurance reporting data can be derived and audited.

A Roadmap

In order to move forward using the IDIOM approach, the following steps need to be taken.

Phase 1 – Parallel Abstraction

The Abstraction Phase can be executed independently for each legacy system. Allow 2-6months per system, depending on the age and complexity of the system. Systems can be done in parallel if multiple systems.

Note that there is an end-to-end dependency between the completion of Phase 1 for all systems, and the completion of Phase 2, which includes the product rationalisation process if any.

Phase 3, the final throw(s), cannot be contemplated until Phase 2 is completed. By inference, the Phase 1 abstractions are initially on the critical path.

[4] For a background on the logic used see https://en.wikipedia.org/wiki/First-order_logic
[5] A speech form or an expression of a given language that is peculiar to itself grammatically. http://www.thefreedictionary.com/idiom. It is this characteristic that gave rise to the word IDIOM in our company name.

Phase 2 – Business Rationalisation

The product rationalisation phase as proposed is driven by the Accountable Person. Given the Phase 1 activities above, we expect to have a stream of issues that require Accountable Person adjudication.

An early start on Phase 2 will help ensure that the time that Phase 2 is on the critical path is kept to the minimum.

Phase 3 – Final Throw

We expect that the final throw can be prepared and executed in a timely manner when the future system is ready. The lead time for future system preparation should ensure that the final throw is not on the critical path.

Production Use

All of the preceding steps must be able to execute in a single contiguous stream, to be repeated at will for testing purposes, culminating in the final, single-step production throw(s).

IDIOM experience to date suggests that this can be usually be achieved system by system within a 48hour window per (i.e. over a weekend) without requiring disruption to business-as-usual that is attributable to the migration process itself.

Product Rationalisation and Analysis

Product rationalisation and analysis will be driven by and coordinated by insurer business leaders and the Accountable Person.

The Bridge is a sandpit where preferred product taxonomies can be developed and tested. One final taxonomy will survive in the form of a preferred Product structure for each (if more than one) future system.

The data in the Bridge is a combination of product reference data [Product Configuration] and entity transaction data. The Bridge Product Configuration data will be synthesised from all of the products that are to be included in that candidate's product family and may be held (in the Bridge) as simple tables, relational tables, or XML as convenience requires.

The Bridge transaction data will be held as xml in a format that is consistent with the requirements of the Product Configuration. It is a design objective (but not a technical imperative) that there be a single omnibus transaction entity schema for all transaction entity types.

Each future system candidate will be supported by a series of IDIOM Decision Models that implement the Controlled Calculations, and all transformations and reconciliations.

Selection of the Release Candidate(s)

There is the potential to throw to multiple future systems. This could be desirable if the customisation cost for the preferred System is excessive for the strategic value of the

products being thrown. A more flexible, lower cost System option may be preferred for these products, with IDIOM offering its technology as a candidate.

It is highly desirable that the Product Configuration data design that is assumed by the future System is used as the starting point for the candidate (product taxonomy) selection process. The rationalisation objective would then be to reduce the number of defined products to the smallest (or otherwise optimal) set that can describe the largest set of transaction instances, with priority given to the on-sale products, followed by the in-force products.

The IDIOM based re-creation of the Controlled Outcomes will then empirically confirm whether or not the existing future system Product Configuration can adequately describe the candidate product set, and/or what customisation and additional features and functions may be required. This in turn may be valuable input to finally confirming any demarcation of products for each future system.

This may be an iterative process, leading to an eventual agreed product taxonomy, which may include different Product Configuration approaches for different product hierarchies if the complete portfolio is to be allocated across more than one System.

When agreement is reached on the allocation of Products to one or more future systems, then the next task will occur for each target system.

Prepare for Future System

When the Product Structure Release Candidate(s) have been selected, further decision model development should occur for each as follows.

- Normalise the idealised calculations by an iterative process of aggregating and conditioning calculation components and adjusting both the Product Configuration and the Transaction EntityXML as required for maximum simplicity and performance. This process must not materially change any calculated value.
- Produce an extract of the finished database and reconcile to previous control points. This reconciliation must be exact.

Map to the Future System

The process is now ready to throw to the future System. The Product Configuration is now in a tested state and is known to work in the context of the Transaction EntityXML. Both the Product Configuration and the EntityXML schema should now be familiar to the target system vendor and can be validated in advance. Mapping to the specific schema (DDL or xsd, including ACORD) that is used by the future system should be mechanical and low risk. By definition, any mapping conflict will be known in advance and should have been resolved through customisation of the target system.

The actual throw can then occur repeatedly as required until a cutover is made.

FURTHER OPTIONS

Calculation Support to the Future System

The process as described above will by necessity have recreated all of the critical calculations (the Controlled Outcomes) for each of the legacy systems. These calculations will have been aggregated, normalised, optimised and validated for the Candidate Release for it to have been successfully validated and promoted.

These calculations will exist as IDIOM Decision Models, which can be natively executed via the Calculation Engine (either JAVA or Microsoft C#). There are several ways that the Calculation Engine can be used directly by a future System.

For the sake of clarity, the following suggestions do not require any further work on, or development of, the validated calculations.

Direct Call

If the platform allows, the Calculation Engine can be called directly in the address space of the calling function. If this approach is used, the calling system would be required to instantiate a DOM (or JDOM) that matches the Transaction EntityXML and pass it into the Calculation Engine by reference.

Service Call – Caller Supplies Data

The service call would wrap the Calculation Engine in a service of the vendor's preference, which may be a queued service, a web service, or any other form of asynchronous or synchronous call to an external process.

The data would need to be provided by the caller in a format which matches the Transaction EntityXML. The results are returned to the caller in the Transaction EntityXML.

Service Call – Service Acquires the Data

The service call would wrap the Calculation Engine in a service of the vendor's preference, which may be a queued service, a web service, or any other form of asynchronous or synchronous call to an external process.

In this approach, the service would receive a Transaction Entity key and a service request. The service would read the data from the System database and perform the calculation. The results could either be posted back to the database, and/or returned to the caller.

Further Benefits

Given any of the approaches above, it is plausible, even desirable, that the Accountable Person should directly own and manage the calculations on a go forward basis. IDIOM provides extensive support for SME driven unit and regression testing at scale, automated documentation, and audit and control of released artefacts, so that deployment of SME defined calculations directly into the system can be made both risk averse, and transparent

and auditable, as prescribed by CPG235. This process is described by one IDIOM customer as 'nimble, continuous, perpetual'.

Audit and Remediation

Even given a perfect throw it is plausible that data will be found to be unfit for purpose in its new environment, or the new environment generates unexpected outcomes. In either case, regular (daily) 'business as usual' data validation and recalculation can provide peace of mind. This process can be implemented on behalf of any future system re-using the IDIOM calculation assets already described.

The IDIOM approach generates a register of precise item by item issues at Assurance Levels Two and Three that can be used to drive further analysis and remediation. It is a feature of the IDIOM approach that the identification and analysis of Alerts as described in this document is a precursor step that is intended to progress seamlessly into analysis of remediation options, thence actions, and outcomes, both during the migration, and thereafter into production as required.

The IDIOM approach fully supports live system remediation as follows.

When reviewed by SME's, the Alerts are recognised as situations that then require situation analysis as to cause and consequence, followed by analysis of remediation options. The options are subject to governance decision making by the Accountable Person to determine subsequent actions to achieve the remediation outcome.

The IDIOM Decision Manager Workbench™ is purpose built to support this data quality assurance cycle, which is summarised as follows:

- ✓ Identify the Situation;
- ✓ Situation Analysis (cause and consequence) of the Situation;
- ✓ Analyse Options;
- ✓ Governance Decision Point: Accountable Person to select and approve the preferred Option;
- ✓ Perform the Action as required by the Option;
- ✓ Confirm and document the Outcome.

IDIOM asserts that this process will help the insurer to meet its obligations under CPG235. Further information on IDIOM's approach to managing CPG235 can be found in the IDIOM whitepaper 'Data Integrity in Financial Services' available here.

Finis

John Salamito | CEO & Director | IDIOM Solutions

☎ +61 2 9968 4168 | +61 421 587 100 | ✉ Suite 16, Level Two, 357 Military Road, Mosman NSW 2088 Australia | @ john.salamito@idiom-solutions.com | idiomsoftware.com

Mark Norton | CEO & Founder | IDIOM Ltd

☎ +61 2 9968 4168 | +64 21 434 669 | ✉ Suite 2-2 93 Dominion Road Mt Eden, Auckland 1024 New Zealand | @ mark.norton@idiomsoftware.com | idiomsoftware.com

Mark Midwinter | Director | IDIOM Solutions & IDIOM Ltd

☎ +61 2 9968 4168 | +61 419 211 192 | ✉ Suite 16, Level Two, 357 Military Road, Mosman NSW 2088 Australia | @ mark.midwinter@idiom-solutions.com | idiomsoftware.com

Jeremy Grigg | SVP and Regional Head | IDIOM North America

☎ +1 877 835 0004 | +1 512 758 3641 | @ jeremy.grigg@idiomsoftware.com | idiomsoftware.com

☎ +61 2 9968 4168 | idiomsoftware.com |
✉ Suite 16, Level 2, 357 Military Road, Mosman NSW 2088 Australia |