“We know now that the source of wealth is something specifically human: knowledge. If we apply knowledge to tasks we already know how to do, we call it ‘productivity’. If we apply knowledge to tasks that are new and different we call it ‘innovation’. Only knowledge allows us to achieve these two goals.”

—Peter Drucker
INTRODUCTION

Every organization has unique business domain knowledge that provides them their primary source of productivity and innovation. All too often, however, this knowledge is not efficiently captured and used. It is scattered in documents, spreadsheets, models, people’s heads, and buried in software code. What if this knowledge could be more easily captured in your organization and treated as an asset that can be shared, reused, activated and processed to create better business outcomes?

Intentional Software offers a platform that help customers capture and make more efficient use of their business domain knowledge. Knowledge is turned into a processable asset. Similar to other processing technologies - such as word processing, data processing, natural language processing or transaction processing - knowledge processing operates on knowledge as input and transforms it to a useful output, like automating a process, or generation of software and other artifacts.

By representing the knowledge of a business concisely in a structured way, much labor intensive work, including reading documents or models, manually checking them and recoding them into new formats, can be completely automated, thus reducing errors, time and manual work. Instead knowledge is used as processable instructions which can “execute” the knowledge similar to how software programs execute computer instructions.
BUSINESS DOMAIN KNOWLEDGE

Today organizations use a combination of documents, spreadsheets, databases, XML, models and software to describe and automate the knowledge in their business processes. The business professional who has the business domain knowledge - the subject matter expert, or domain expert - records their knowledge in documents, spreadsheets, and models. These specifications are read and manually translated into new artifacts, resulting in more documents, more spreadsheets, more models, more schemas, and more software code. Not only is this a labor intensive, error prone and fragile process, but it also lends itself poorly to repeatability when changes occur.

As an example, using this traditional process, a typical description from a professional using a word processor to document a medical health assessment form could look like the following Word document, which would require manual interpretation and translation to some other form to be ready to automate.

PROCESSING KNOWLEDGE

In contrast to the traditional manual flow, using Intentional’s platform for processing knowledge, the domain expert, as before, records the knowledge - their intentions - in a document, but this time the document is an Intentional structured document. The domain knowledge in the document - medical terms, questions, measurements - is represented in the structured document expressing the knowledge from the domain expert. The new description is very similar to the Word document and looks like this:

In this new document, the content is essentially the same as in the Word document, but has a rich structure which allows processing of the knowledge to analyze, verify, execute and transform it.
The knowledge from the medical domain expert is now used to define how the data gathered when executing the previous health assessment will be used to devise treatment plans for a specific patient. These treatment plans are also recorded by the medical domain expert in a structured document:

**VERIFYING KNOWLEDGE**

These rule sets can be analyzed statically and executed with test data to verify their completeness and correctness. This gives the medical domain expert direct feedback of how correct and complete their knowledge is.

To run the knowledge, we use an execution environment for the medical domain at hand. Below is an example of a test execution of the questionnaire we saw earlier:

The medical domain expert is recording the rule sets using answers to the questions in the health assessment questionnaire by referencing these questions explicitly. A rule set can also create new definitions of calculated values from the patient medical form, for example “Waist to Height Ratio” which is calculated directly from the patient data coming from the patient when the questionnaire is executed.
EXECUTING KNOWLEDGE

We capture the knowledge in a processable for which will allow us to directly execute it, similar to how a spreadsheet computes formulas. The difference to a spreadsheet is that we use the Intentional platform to create a structured document that a domain expert can use that closely matches how people in that domain already communicate, and yet the knowledge is directly accessible to automation.

One application of executing knowledge is to produce standalone artifacts like policy documentation, documentation of processes, forms, call scripts for data centers, or help files.

Another application of executing knowledge is to generate software from it. Rather than directly running the knowledge descriptions as we did above, we can instead transform the knowledge into traditional software code for a different execution environment. We can “compile” it to a different execution environment which uses a traditional computing language.

One example could be a rules engine where knowledge is compiled to rules engine code. The example here shows this compilation. Note how each rule of the domain knowledge (left) has been transformed into executable code in a rules engine programming language (right). Any changes in the domain knowledge can be directly recompiled into new version of the rules engine code.

Note that the domain expert (left) does not need to know anything about the execution environments or computer language (right). The knowledge is distilled in a concise form which does not include any implementation assumptions or details. Indeed it is completely implementation independent and can be directly targeted to multiple execution environments without changing the business domain knowledge itself.
DOMAIN SPECIFIC LANGUAGES

The knowledge description has syntax and semantics which is what is being executed. It is “language” and “code” in the traditional computing sense; “domain language” and “domain code”, i.e. language and code for a specific domain. The domain language and domain code can be interpreted or compiled since they have computable syntax and semantics. This is often called a Domain Specific Language (DSL) to distinguish it from a General Purpose Language (GPL) like Java, C++, C#, Python, COBOL, or FORTRAN.

Traditionally DSLs have been technical in nature targeting technical users. Examples of popular DSLs are:

- SQL – databases
- HTML – web layout
- XML – data encoding
- UML – visual modeling
- SysML – systems modeling
- VHDL – hardware design

Note that all these languages end with an “L”, which stands for “Language”. Here is an example of some SQL code:

```sql
SELECT title,
       COUNT(*) AS Authors
FROM Book
NATURAL JOIN Book _ author
GROUP BY title;
```

DSLs have proven to be very powerful and successful, and there are thousands of DSLs in use today, most of them proprietary. One problem is that these DSLs require a strict syntax and semantics, and therefore have not been so accessible to non-programmers, specifically people with the domain knowledge. Can the power of DSLs be made available to the business domain expert?

DSLs FOR BUSINESS USERS

The Intentional Platform uses DSLs at its foundation. As we saw in the SQL example, DSLs tend to be very technical and hence not suitable for business users. Therefore the Intentional Platform projects the DSL into a user interface experience that the business user is more comfortable with. The Intentional Platform uses a projectional editor that transforms DSLs into a user interface accessible to business users.

As we saw in the medical domain knowledge example in the previous pages, the projections of the DSLs can be tuned to the exact need of the business domain expert - the person with the business domain knowledge.

Many business users are used to an interface like Microsoft Office® – Word®, Excel®, PowerPoint®. These systems, underneath the user interface, also have a rich structure which is being presented as an interactive environment for the end user. Similarly, the Intentional Platform presents an interface for DSLs that is similar to a Microsoft Office style application. Below is an example from a Financial Document Workbench to produce monthly financial reports from Excel data, and other data sources:
APPLICATIONS

The previous example illustrates a typical application of processing knowledge in the medical domain – the creation and running of software directly from medical knowledge descriptions. Intentional customers use their own unique business domain knowledge for processing in several applications; indeed there is an endless opportunity of applications by encoding the specialized knowledge of domain experts. Here are a few more applications from projects.

DATA MAPPING AND TRANSACTIONS

The mapping of data elements to databases and transactions is a tedious, repetitive and error prone process that requires not only deep technical skills, but also a deep understanding of the business process to be automated. Recording the domain and technical knowledge in a processable form can automate the integration and verify mappings and corresponding transactions. This is especially beneficial when new or changed data elements and processes are introduced frequently. The example below shows an integrated view of business domain and technical knowledge of a certain data element with its associated business rules.

These data mappings and transactions must also be verified and tested in production systems through a test harness. The example below shows such a test of corresponding transactions.
ANALYTICS AND VISUALIZATIONS

Data analytics either require specialized programming skills, or apply simplified tools that limit what analytics and visualizations can be done. By combining the best of both worlds, the knowledge processing solution uses advanced big data processing back ends and plug and play visualization frameworks for the front end. The domain knowledge here provides a business user interface to control the back end processing as well as the visualizations. The description below is an example of such a set up to analyze a stream of data:

MODEL INTEGRATIONS

Systems engineering includes the use of several different models to describe various aspect of a system design. These models are typically not integrated with each other leaving large amount of manual knowledge work to reconcile these models as changes occur. Using knowledge processing, we define the languages used to describe these models and do correlation and automatic transformations across models in different formats. For example a CAD model can be related and linked to rows in Excel, and a model in SysML. The example below shows a Systems Engineering design where these three types of models have been integrated and linked for use in dependency tracking, consistency checking and impact analysis of a system design.