



Intelligent Text Processing

ITP

Document Automation Building Intelligent Systems

2004-2-15

This paper introduces the concepts of Document Automation, the confluence of work and result as implemented using ITP.

1.0 Your Business' Processes and Documents	3
1.1 What is a Business?	3
1.2. Document Automation	4
1.3 Document Automation Technology	4
1.3.1 Re-inventing the Wheel	5
1.3.2 A New Kind of Wheel	5
2.0 Intelligent Text Processing (ITP)	5
2.1 What is ITP?	6
2.1.1 Core ITP Value	6
2.1.2 How Does ITP Work?	6
2.1.2.1 Document Model Constituents	6
3.0 Line of Business (LOB) Integration	11
3.1 Document Output	11
4.0 Where to Use ITP	11
5.0 ITP's Business Value	12

Your Business' Processes and Documents

All companies and organizations of any size generate information that is used for a variety of purposes. The information that is generated does not simply exist in the ether alone. This information is transformed into “Content.” Content is simply a meaningful expression of information that is relevant to the stated business goal.

However, transforming information into Content is a manual process. Transforming information, which can be simple or complex—is performed by people within a business or organization. These people are called “Knowledge Workers.” Knowledge Workers plan, create, and process information to create Content. Furthermore, one can view transforming information into Content as “Work.”

Therefore, as Knowledge Workers perform Work—they express that Content within a media. The media most often used for expressing Content are “Documents,” or virtual facsimiles thereof—or in simpler terms, electronic versions of Documents.

When this Content is expressed within a Document media, it reveals two distinct facets. The first facet is obvious: Content expressed within Document media is simply the Document itself, which is comprised of a number of different elements. Document elements are things like text—either boilerplate or merged insertions, tables, graphs, and logos. Elements can be inserted on the fly (interactively), can be selected using programmatic instructions, or can be static, in other words—unchanging.

The second facet that is revealed is a bit subtler in that the produced document also reflects the Knowledge Worker's series of actions that have occurred to create the Document. In other words, to create a Document, the Knowledge Worker must execute a “Workflow” to generate a result. With careful post-transformative analysis, one can observe the required actions it took to create the Document, and in turn the Content itself. Thus the document has within it an inherent “Document Flow,” that is, the steps it took to transform information into Content, and Content into the Document(s). Furthermore, a workflow process often has multiple participants in the flow of work across a business process.

Looked at in this light, one sees the confluence of process and result. In other words, Workflow is Document Flow. For further purposes of this discussion, the term “Document Flow” will imply both Workflow and Document Flow.

What is a Business?

Simply said, a business is consists of a series of interconnected Document Flows that combine to achieve a stated goal. These Document, when produced, can number in the hundreds if not thousands and some of these have higher values than others.

Highly valued Document Flows—of course—are only known to the businesses that use them. Generally, however, highly valued Document Flows occur multiple times and

contain varying content, which require personnel and physical resources to execute. Some of these Document Flows spawn other Document Flows. Some terminate others. The inter/intra-modality of the Document Flows can be stunningly subtle, yet powerfully effective. However, all of them result in one thing—the creation of Documents. Generally, increased efficiency, or reduction in cost, in the creating these high value Document flows can produce stunning ROI.

Documents can include many things—some must use data expressed by the Knowledge Worker alone, some Documents need information merged into them from data sources contained within an organization’s database. Some require both. It is these highly valued Document Flows that we look to automate.

The result...

...Big payoffs in productivity, accuracy and ease of transforming information into Content, which in turn is expressed as a Document for achieving your business’ goals.

Document Automation

Highly valued Document Flows are typically performed over and over to produce what can be called “Document Runs.” These Document Runs, many times, require the insertion of data from a data repository automatically, or perhaps interactively where a Knowledge Worker—in real-time—selects, or creates the appropriate data to be inserted into the Document.

In all cases—automatically, interactively, or a combination thereof, the data can take two fundamental types. The fundamental types of information that can be applied to a document are: information that is unique to a particular document, for example name/address. Or the second type: standardized information, which is characterized as “boilerplate” information.

Boilerplate information is things like form-completion instructions, variable disclaimers, and so on. Yet both unique and boilerplate information contains detailed, self describing properties like fonts, their sizes and display characteristics, organizational elements like layout, tables and graphs, as well as purely visual elements like graphics and logos.

It has however; been historically difficult to synthesize all the elements efficiently in which business-to-customer, or business-to-business relationships can be executed in an ever-changing, highly competitive environment.

Document Automation Technology

Document automation technology has in the past been expensive, hard to implement, was rapidly outdated, and tough to maintain. This was due to the combination of the ever-changing need of Knowledge Workers’ information processing, and the limited document quality that the technology produced.

Re-inventing the Wheel

Knowledge Workers transform information into Content, which is expressed in Documents like forms, letters, contracts and so on. However, many times each element of work demanded a new process, and thus an entirely new effort was needed to generate similar or identical results. Re-use of invested work simply did not exist.

Because of this, Knowledge Workers were forced to re-create the wheel many times each day *to their consternation and their manager's frustration*. Additionally, if automated document processing was available—it was limited to the most basic types information expression. Additionally, if data-merge technologies *were* available—they were so cumbersome to implement and took additional technical resources that many times it was just easier to re-invent the wheel each time.

Modern document elements also were simply not available to these nascent technologies. Never could one interleave organizational elements with data—like placing variable information within easy-to-read graphs and charts. Changing a company logo within a document was no small feat even for the technically savvy. One could simply disregard any kind of large document generation without a huge effort within the organization's information technology community.

A New Kind of Wheel

Ideally a Knowledge Worker should be able to transform their information into Content—and thus Documents, in a seamless environment that does not require any specialized tools. They should be able to collect their requirements from disparate data sources, and have the ability to perform—in real-time—adjustments, corrections and introduce new Work into their Document Flow effortlessly.

Additionally, their Content should not be bound to a specific media or limited in the method of delivering that Content to its targeted goal. The resultant Document must employ the best-of-breed elements without incurring additional cost of achieving those elements.

In other words Knowledge Workers are in desperate need of a new kind of wheel. One that will ever be changeable, always up-to-date, and easy to operate.

Intelligent Text Processing (ITP)

Information Technology Sources Group (US distributor) distributes and provides Licenses for Intelligent Text Processing's (ITP) suite of Document automation products. ITSG offers ITP to sub-distributors, ISV's, selected Partners and selected end users. We offer ITP Training, ITP development (professional services), integrating ITP into LOB applications, IT consulting, and ITP Helpdesk.

What is ITP?

ITP is a family of text processing productivity products that include a document processing manager, web services, and interactive/Server Document Run processors that are used to generate and manage Document Flow. These items are married to an integrated software solution for automating Content production using advanced data merge and delivery technologies.

Core ITP Value

Automated Document generation and Document Flow manipulation are achieved through the application of powerful document modeling technologies in which data can be merged from any type of data repository. This results in advanced document automation at the desktop or server levels. The ability to merge data into Content from Knowledge Workers and their other content transformation tools (like a Line-of-business application) from any data source, across disparate IT platforms makes for the easy assembly of complex documents/forms/reports from mixed IT environments. Both small and large-volume Document Flows can be easily generated, managed, and modified.

In addition, ITP can work as a Knowledge Manager. A Knowledge Manager is an assisting technology that operates in concert with Knowledge Workers. By predetermining appropriate content and processing logic, ITP as a Knowledge Manager allows Knowledge Workers to streamline their analysis procedures. This is done by having ITP restrict Content selection, generation and gathering of disparate data to an appropriate scope for a particular Document or group of Documents.

With this, businesses can introduce new Knowledge Workers into their processes with greater ease because there is a flatter learning curve for assessing the subtleties for transforming information into Content, and thus a faster, better result from your Knowledge Worker's output.

How Does ITP Work?

ITP is based on a powerful document modeling technology in which programmatic model instructions are scripted (for example) into a Microsoft Word (actually used as a programming editor). The scripting not only defines and gathers information from disparate data sources, Knowledge Worker input, and so on, but also how the Document itself is laid out, and displayed.

In other words, a Document Model is the blueprint for the final output—and is called a “Result Document.” The Document Model contains layout elements, ITP text generating instructions, how merged data is retrieved from data sources, and how the data is manipulated within the Document.

Document Model Constituents

A Document Model is the blueprint for the Result Document. But along with the Document Model's Content—there is the question of how ITP manipulates the Content, gathers data from data repositories, and combines Content and gathered data.

ITP can work with any modern word processor or standard text files (.txt files). MS Word is used here as an example word processor due to its ubiquitous use in today's businesses.

ITP as a High Level Language (HLL):

ITP provides a fully functional HLL that is expressed within MS Word documents. By interleaving ITP HLL and boilerplate text, ITP is able to gather data, perform calculations and manipulations on that data, which can come from a database or Knowledge Workers, and also produce boilerplate text.

Advanced manipulations can be scripted within a Document Model to provide for simple to complex data manipulations, such as: calculations, conditional constructs, one-to-one, one-to-many, or many-to-many relationships, loops and arrays. These manipulations can act as a programmatic Knowledge Manager for the placement of text based on a "true" programming construct as it follows its instructions within a model as it executes.

As the Document is generated, the Document Model gathers information from data repositories. Data repositories can be any modern database, transformed information provided by the Knowledge Worker, the selection of Content by the Knowledge Worker, or some combination of the three.

A Knowledge Worker's Input:

Gathering Knowledge Worker-based information is provided by an "Interact" construct. The Interact construction enables a Document to interactively ask questions to the user and act upon the answers to those questions. Using this construct one can build a question and answer series after which, for example, text is produced in relation to those answers.

The Interact construct can optionally be made available at Document run time depending on the specific Document needs. In this scenario, at run time—the Knowledge Worker is presented with a Windows-based series of screens that offer any combination of Content selection, real-time Content entry, or the entry of conditional variables, which can then be programmatically processed to generate the desired Document Content.

Merging Database Information:

The methodology in which a Document Model gathers database information is through the use of an embedded interface (post compile) within the Document Model. Called a Data Interface Definition (DID), this interface reflects the superset of all the potential data in a Document Model, or multiple Document Models require for generating a single or multiple Result Documents.

The ITP DID is developed prior to any Document Models. The DID is created using the ITP software development kit (SDK) in which abstracted data definitions and connection methodologies identify the database from which to gather data, or the data retrieval

ITSG

Information Technology Sources Group

programs that perform the access and gathering. DID development provides an integrated GUI for semi-automatic DID generation.

DID Constructs:

A DID has two fundamental data sorting constructs—a “FORALL-DO” construct and a “WITH-DO” construct. These two constructs are virtually self-documenting in that the following pseudo code is easily understood. The following ITP pseudo code accesses a single database where a specified condition is defined, and data is assigned to a variable:

```
FORALL records in a x database WITH y condition

    ASSIGN database_field_last_name =
        GivenDatabase.Table_Customer_Name.Field_Last_Name

    ASSIGN database_field_account_balance =
        GivenDatabase.Table_Customer_Accounts.Field_Account_Balnce
DO
```

The above pseudo code, which is written within an MS Word document, compiled, and when ran would return a joined-table row from the given database in which each of the returned field data is assigned to a variable. The variable in turn would then be placed within the same Document Model where those specific values are required:

Dear Mr. @(database_field_last_name);

Your account balance is \$@(database_field_account_balance).

Sincerely,
Accounts Billing Department

Thus, upon Model Document compilation and a subsequent run of the same would generate the following MS Word document:

Dear Mr. Smith,

Your account balance is \$500.00.

Sincerely,
Accounts Billing Department.

It is the DID that is embedded within the compiled ITP Document Model that provides the global logic for inserting data merges of unique information into the individual Document when ITP executes its RUNMDL.exe. This process occurs whether a single document is required or thousands of documents are needed. **It is ITP's easy-to-implement data merging abilities that is the highest and best use of the product.**

Multi-Platform Interfaces:

Because all modern databases conform to standard connection methods like ODBC, DB2, MQ Series, native Oracle or MS SQL, an ITP DID leverages those standardizations to access any modern database. In this light a Document Model, which can contain different embedded connection methodologies and different data repository definitions, is able to access virtually any combination of databases available to the business community.

True Multi-Platform Processing:

Should a Document require the assimilation of data from a variety of disparate databases, and Knowledge Worker input, ITP provides this powerful multi-mode ability. The following pseudo code access two databases (x and z) via a single DID where the primary key for the z database is available within the x database. In turn x database's key is passed to the z database in order to identify a customer's detailed audit information:

```
FORALL records in a x database WITH y condition
  ASSIGN z_database_primary_key = x.Table_Customer_ID.Field_ID
  ASSIGN x_database_field_last_name = x.Table_Customer_Name.Field_Last_Name

  FORALL records in a z database WHERE z_database_primary_key =
    z.Table_Customer_Audit.Field_Audit_ID

    ASSIGN audit_detail_id = z.Table_Customer.Customer_Detail
    ASSIGN outstanding_bal = z.Table_Customer.Outstanding_Balance
  DO

  INTERACT determine_include_summary
    QUESTION "Do you want to include summary information? (Y/N)"
    ANSWER
    IF QUESTION = "Y" THEN
      ASSIGN summary = run_summary(audit_detail_id)
    ENDIF
  ENDINTERACT
DO
```

The above example includes an Interact construction that—at run time and before a Document is output, will present the user with a Windows screen that allows the entry of a “Y” or “N.” This will include summary audit information on a customer should the Knowledge Worker desire it.

The Windows screen (a form) is generated automatically by ITP and contains all the proper labels and controls to include pull downs, textboxes, checkboxes, and buttons.

The summary information that is generated using the Interact construct is generated using a function (not shown) that is passed the audit details identification number. The function's parameter is obtained from x database higher in the processing. The variables containing run time values are used within the boilerplate text like this:

Customer Audit Report

Customer ID: @(z_database_primary_key)

Summary Information: Mr. @(x_database_field_last_name) outstands a balance of \$@(summary) for all accounts.

Detail Information:

Account @(z_database_primary_key) has @(outstanding_bal) outstanding.
Account @(z_database_primary_key) has @(outstanding_bal) outstanding.

When ran, the above code results in a MS Word report that looks like this:

Customer Audit Report

Customer ID: Smith1

Summary Information: Mr. Smith has an outstanding balance of \$500.00 for all accounts.

Detail Information:

Account 123 has \$250.00 outstanding.
Account 456 has \$250.00 outstanding.

In sum, the above pseudo code will access two different databases that contain customer relevant information (x database), and company management information (z database) that is combined within a single MS Word Document.

Also included in this example is the ability for a Knowledge Worker to include only detailed information or detailed and summary information in the Document. All of which results in a management report for a single customer with multiple accounts.

Within the DID:

During DID development, any database connection methodology can be defined within a single DID using a single or series of INI text files, which are also—in themselves, Documents too; and can be also be manipulated by ITP using meta-coding techniques.

In other words, should x database be a DB2 AS400-centric database, and z database is an MS SQL Server database—the AS400's TCP/IP address and appropriate data file libraries are identified within the INI file(s) for the x database; and a standard ODBC connection string is additionally defined within that same INI file for the y database.

Upon completion of the DID development and subsequent compilation, both connection methodologies are available to the DID and any follow-on Document development. An easy to understand DID mapping GUI is available for DID developers.

Advanced Meta Coding

Because ITP has the ability to build documents—and in turn standard text files, it can also be used to generate any kind of parameter, .ini, or .cfg file that can be processed with any line of business (LOB) application or network processes that accepts those file types.

Line of Business (LOB) Integration

ITP offers API's that can be integrated into existing LOBs and IT infrastructures. At the heart of the APIs is the RUNMDL.exe or DLL, which provides open interfaces for integration into all platforms from mainframe, AS400 to PC-based architectures.

APIs are offered for the following:

- Mainframe & UNIX/LINUX (MQ Series)
- Windows OS (NT, 200 XP)
- Java
- COM (VB)
- IServer (AS400)
- XML Web Connection
- MS InfoPath

Furthermore, because the connection structures are independent of the documents for which to gather data, data repositories like MS SQL Server, Oracle, DB2, any ODBC compliant database, Lotus Notes, SAP, PeopleSoft, Siebel, Tamio, and so on can be used in any combination to produce any Document.

Also other software applications like MS Excel, ERP, CRM, and CMS data can also be extracted into the Result Document. This makes ITP perfect for complex MIS reports, and real time "Data Mining" across disparate IT systems.

Document Output

ITP's suite of products supports a wide range of output methodologies in which Documents—whether eDocuments, hard copy Documents, or some combination—can be delivered to print (high-volume or low) media, Adobe's PDF format, embedded or attached email, TFFs that are used for faxing or content manager input, and so on. Also ITP can serve information to the web via XML or HTTP protocols.

Large formatting selections are also available and easy to implement. Simply create an MS Word document that conforms to the look and feel of your desired Result Document to include colors, tables, graphics/pictures, electronic signatures, logos/letterheads, barcodes (including postal barcodes), data in headers & footers—insert the proper scripting instructions, compile it; and that is all there needs to be done.

Where to Use ITP

Because ITP technology is so flexible, there are many insertion points within existing LOBs (Lines-of-Business software applications). Simply said—if repetitive communication needs to occur, then there is a need for ITP. Whether your business needs to send letters, forms, emails, faxes, serve information across company portals—ITP does this with a simple set of tools.

Here are some examples of our customer's use of ITP:

ITSG

Information Technology Sources Group

- Direct Mailings
- Monthly Statements
- Insurance Policies (ISO)
- HMO Process Docs
- Web Portal Publishing
- Fulfillment Docs
- Mailing to
- Mortgage Contracts
- Medical Coding
- Web Document Stores
- 1-1 Marketing
- Invoicing
- Mortgage Correspondence
- Medical Reporting
- Banking
- Monthly Billing
- Credit Card Stmts
- Insurance Claim Processing
- HIPPA Processing
-

ITP's Business Value

The most successful businesses carefully integrate existing assets with new technologies to demonstrate immediate payback for their technology investments. ITP meets this goal through the thorough integration of existing—even heterogeneous—IT environments. ITP can be a lynchpin linking all your systems together into one modern, fully scalable solution.