



# ESG guide

*Authored by Arthur Gingrande, Partner, IMERGE Consulting*

## *Processing Unstructured Documents: Challenges and Solutions*

*An AIIM User Guide  
By Arthur Gingrande*

*This is one in a series of User Guides from AIIM International.  
They are intended to educate and inform readers on a variety of  
enterprise content management topics.*

Copyright © 2004 by:  
AIIM International  
1100 Wayne Avenue, Suite 1100  
Silver Spring, MD 20910 USA  
301-587-8202 / 800-477-2446  
ISBN 0-89258-395-9

Published by:



[www.aiim.org](http://www.aiim.org)

## Processing Unstructured Documents: Challenges and Solutions

*The majority of business documents (e.g., invoices, purchase orders, resumes, work orders) are arbitrarily structured and, as so, cannot be processed using conventional “ICR-friendly” document processing approaches. Alternatively, they may be consistently formatted but contain incremental variations from one document to another (e.g., medical claims, IRS forms) that diminish the effectiveness of using customary, ICR-based forms processing applications to the point where they hit unacceptable cost-justification levels. Over the past few years however, advances in automated forms processing technology have dramatically improved recognition accuracy in processing arbitrarily structured business forms. These advances include exponential increases in computing speed and memory, significant improvements in image processing technology, and innovations in neural network algorithms.*

Clearly, ICR-based forms processing has come a long way since its introduction to the imaging world in the late eighties. In those early days of automated forms processing, automatically identifying a given form type and then setting it up for ICR/OCR taxed the capabilities of users, integrators, and vendors alike. Typically, the only forms eligible for image-based forms automation were those that had been created by the company responsible for processing the form. In fact, the forms in the early days were designed explicitly to be “ICR-friendly;” that is, they were specifically formatted to fit the needs of intelligent character recognition software, in order to obtain the most accurate text recognition and hand-printed recognition results possible.

This meant that, preferably, the form data were printed in “drop-out” ink—carbonless ink designed to be ignored by an imaging camera—so that only the “active data” filled in by the customer was actually detected by the scanner. Furthermore, since ICR engines encounter extreme difficulty when forced to recognize connected characters, the form data fields had to be framed by graphical objects known as “combs”—strings of

boxes that forced a person to separate hand-printed characters when filling out the form.

The user created a software template to match each form type and define the ICR parameters of each of the fields on the form—check box, hand print, machine print, alpha, numeric, number of characters, and so forth. Due to the limitations of form identification technology, forms were processed homogeneously by the batch, carefully sorted and tightly organized by form type: one batch, one form type, so that the form template precisely fit each and every form in the batch. If the layout of any form in the batch differed incrementally from that of the form definition template, the system rejected the form and sent it to a human for manual data entry.

Over the years, form identification and data location algorithms have improved considerably. Setting up a form type—or a number of form types—for automated recognition has become a relatively simple task for users to accomplish with about any given forms processing software system. An end-user can establish form ID and data field parameters in a few simple steps, using a form setup module that creates a mouse-driven, form definition template with the aid of a user-friendly graphical user interface (GUI). Algorithmic software innovations allow ICR and OCR engines to recognize data elements even when the target data falls outside the zones of the form definition template. Taken collectively, these innovations add up to higher recognition accuracy. For all practical purposes, the problem of processing structured data on forms is solved.

Nowadays, however, it is unstructured data that is getting all of the attention. *Unstructured data accounts for nearly 80% of all of the corporate data on record.* This percentage applies across the board to all data types, including email and voice messages, slide presentations, videos, attachments, paper documents, and paper forms. The overwhelming volume of the information involved makes processing unstructured data for storage and consumption a major priority for most corporations. This preoccupation with unstructured documents prevails currently in the world of paper forms, where the hottest document management application is the processing of unstructured forms such as purchase orders, invoices, medical claims, and explanation of benefits (EOB) forms.

## Chapter One:

### Defining the “Unstructured” Document

The term “unstructured” document, of course, is a misnomer when applied to forms. After all, the essence of a form is structure. One could go so far as to say that, in reality, a form is nothing but structure devoid of content—in effect, a form is an empty vessel waiting to be filled with data. Moreover, there is no question that, purchase orders and invoices, for example, are consistently structured from one instance to another—at least from the standpoint of the issuing company. The term “unstructured” becomes meaningful if and only if it is interpreted from the perspective of the company on the receiving end that must process the myriad of purchase orders, invoices, shipping documents, medical claims, or explanation of benefits (EOB) forms that arrive daily at its headquarters in a variety of layouts that differ arbitrarily from one sender to another.

These random differences over a series of functionally similar forms mean that the forms cannot be processed using the traditional template-based approach, in which one software template matches each and every data field on all the forms in a presorted batch. Instead, the data on these forms must be entered manually by human operators, and at considerable cost.

If the operation is automated, then it must be capable of processing unstructured forms “on the fly,” using sophisticated algorithms that locate the data on a diverse array of form layouts within a given form type. Forms that originate from a continually changing number of sources, and that arrive in arbitrary sizes and formats, require a system that employs relatively little explicit, programmatic knowledge about the formats of incoming documents. In other words, accurately automating the processing of unstructured documents is successful under conditions where the forms automation software is indifferent to the targeted form data. Jerry Fisher, Vice President of Symbus Technology at the time, once remarked that, “forms processing will penetrate the mainstream market only when the software that drives it is as indifferent to the form data as the form itself is.”

Accordingly, the following definition represents the consensus of the major forms processing vendors, based upon interviews with them conducted by IMERGE Consulting. It will govern the use of the term “unstructured” document in this paper. An unstructured document is a class of forms in which:

- The individual forms are not designed by the processing agent;

- The forms are received in a variety of layouts that can differ from sender to sender; and
- The forms cannot be processed using the traditional form template that consistently matches up with each and every data field on all the form documents in a presorted batch.

In other words, *the term “unstructured” document refers to a group of functionally alike forms with dissimilar layouts, which are received by the processing agent in collectively high volume and that would normally require manual key entry in order to capture their data.* In reality, a better term, slowly coming into use, would be “semi-structured” forms.

### Major applications

Unstructured document processing technology is just starting to vie for mainstream user adoption. In a survey conducted by IMERGE Consulting, vendors reported that the most popular applications include the following, listed in the order of consumer demand:

- **Invoices and purchase orders**—Every major company processes mountains of invoices every day, not to mention their sister form, the purchase order. Their arbitrary format makes it necessary to enter the data contained manually. Nowadays, every major forms processing vendor offers an invoice solution that can also do purchase orders.
- **Medical claims**—Data on an HCFA-1500 medical claim form is contained within 31 numbered regions, which in turn contain over 100 inconsistently placed, densely packed data fields within them, which can include machine print, mark sense, hand print, bar code, columnar, signature, and, in some instances, bar code symbols. HCFAs, UB-82s and UB-92s, and dental claim forms are composed data fields. Each sports its own peculiar array of lines, boxes, and fine print instructions—printed in red, green, or black—and each color presents its own set of imaging problems. Add in the OCR complications of degraded font recognition created by claims filled out by faded ribbons from old dot matrix impact printers chronically used by doctors, and processing medical claims becomes the most challenging task in forms automation.
- **Explanation of Benefits (EOB) forms**—Derived from a medical claim, an EOB statement is no ordinary form. Essentially, it is one big table with multiple columns of data that frequently spans many pages. The number of data items within each column is indefinite, and vertical positions on a page

are unpredictable. There may also be intrusions of summary data items sandwiched between consecutive patient accounts. Moreover, within a single column, several different data items may or may not be staggered vertically. The overabundance of variable-width columns and tightly-packed data fields, often spread over 20 pages, makes EOBs particularly difficult to recognize by conventional recognition systems.

- **Transportation documents**—These include shipping documents such as bills of lading and customs declarations. Like invoices, these forms are processed by parties that did not create them, so their layouts appear arbitrary to the processor. Moreover, the use of carbon copies and the beating that these documents take en route to their destinations compound the recognition problems involved.
- **Tax forms**—The tax form is one of the most difficult forms to recognize by machine, for many reasons, including the complexity and density of the form, its myriad parts, its multiple-page nature, and the variance in possible layouts, especially since many of the state IRS departments accept forms from accounting firms and tax preparation software packages that do not match the government-issued form. The sizes of the data fields on an ordinary tax form do not encourage the taxpayer to separate characters legibly from one another. Moreover, since the data fields can be filled out in either machine-print or hand-print, the ICR engine must be able to switch from one recognition mode to another ad hoc. Statistics are unavailable on how many invoices are processed by American businesses each year. Since invoices are the documents that demand payment—and hence are the form-based lifeblood of America’s capitalistic economy—it is intuitively obvious that they are the most prevalent and mission-critical of all unstructured form types. At last count, there were over 75 million pages of medical claim forms processed daily, at a cost that exceeds \$125 million per day—and there is at least one EOB form for every medical claim. Transportation documents proliferate in obvious abundance, but the degraded conditions of many of them, because they are attached to the outside of packages and are often carbon copies of originals, create additional problems that make them the province of specialist integrators.

Tax forms processing projects, because they involve state and federal government documents, are usually done on a special bid basis, and require massive amounts of tailored validation routines and code-writing by developers and systems inte-

grators. Over 30 states already have systems in place, and the federal government has a number of programs dedicated to automating tax processing and auditing procedures. Consequently, like transportation document processing, tax forms processing is an application for which a software solution is not in very high demand by corporate America—especially when compared to invoice processing, claims processing, and EOB processing.

## Chapter Two: Properties and Procedures

It would be incorrect to treat all unstructured forms as one document class with a set of variable but definable characteristics. Clearly, an EOB differs markedly in appearance from an invoice or purchase order. Each form type must be processed according to its own set of business rules. Each type of unstructured form has idiosyncratic characteristics that present a unique set of problems to the I.T. professional, which, taken together, require a customized approach to process them successfully. In fact, each application has become so customized that the current practice among form processing vendors is to produce a separate software product for each major unstructured forms processing application.

Hence, in today's marketplace, the typical publisher of forms processing software makes available to its customers one software product for invoice processing, another product for medical claims, and still another for EOBs. Alternatively, a vendor can offer a generic "any form" processing engine that accepts plug-in modules that are tailored on an application-specific basis to process invoices, claims, or EOBs, accordingly. Because these are the three highest-volume applications, each one lays claim to its own product. As time goes by, no doubt, vendors will publish a software package that features a transportation forms processing wizard, but currently there is none.

Factors to consider when processing unstructured documents include data location variability, data field ambiguity, data field density, form removal, multiple pages, knowledge of the business process, and foreknowledge about forms.

### • **Data field location variability**

As previously discussed, due to the variety of graphical layouts and data formats, the major difficulty in processing unstructured documents involves simply locating the array of suspect fields that must be captured in order to yield the specific data required to process a given form type. There are a variety of ways to locate suspect data fields on forms processed "on the fly," which will be described later in this paper. For example, in order to locate a sum at the end of a column, brute force OCR can be used to recognize every field in a certain region of a form, and then the total can be discovered by looking for a "total" or "sum" field. Alternatively, the morphology of the form itself can be examined in order to detect the last entry in a columnar "blob" that is then passed on to an OCR engine to be recognized.

### • **Data field ambiguity**

Due to the imperfections of the real world, objects on forms, such as invoices, can appear as many things, open to a variety of differing interpretations. For example, not every amount on an invoice form is the amount that the customer must pay. There may be the original billing amount, the net billing amount "if paid within thirty days," an amount that is a subtotal of the final total, and the amount before sales taxes, just to name a few. The same goes for dates. There is the "pay by" date, the "date sent," and the "date received" on the invoice. The latter field could be stamped on the invoice in ink and, if so, might have to be filled in by hand. Consequently, the system must employ intelligent algorithms to figure out automatically what each individual amount and each date means. The same is true for other unstructured documents, such as purchase orders and medical claims.

### • **Data field density**

Data field density can be a problem, especially if the form is cluttered with a lot of "noise" that partially fills in white spaces between the fields and makes them difficult to extract by OCR engines. In particular, healthcare forms, such as HCFA's, UBs, EOBs, and dental claims, are extremely packed with data fields of all types: text, hand-print, check boxes, bar codes, and signature fields. On an HCFA form, there can be over 100 fields on a page, squeezed into 31 numbered regions. State-of-the-art form removal, special data parsing rules and scripts, as well as accurate OCR and ICR, are essential for extracting information from tightly packed data fields.

### • **Form removal**

Invoices and purchase orders typically contain fewer graphical objects, lines, or boxes on them than do medical claim forms. Separating the "passive" healthcare form itself from the "active" data supplied by the medical personnel filling out the form is an extremely complex task, requiring sophisticated forms removal technology. This is primarily because the typical healthcare form is composed of densely-packed data fields, each with its own peculiar array of lines, boxes, and fine print that instructs users on what information to enter in those data fields. White space on a healthcare form such as an HCFA is a scarce commodity, which makes the process of extracting data fields from their prison of passive form elements a far more exacting task than with other types of forms, such as invoices and transportation documents. In

order to facilitate the degree of precision that is required to separate an HCFA form from its medical data elements, the form removal process must be tightly integrated with the form ID and registration process.

- **Multiple pages**

Extracting data from multiple page documents pose significant location and classification problems, due to the variable number of item lines that can be contained on a given page. One invoice, for instance, might be a single-page document, while another one could span many pages. Simply finding the grand total on an invoice or claim form can be a challenge that requires complex and sophisticated algorithms that formulate hypotheses about data content, iteratively test and modify them based upon results, then rerun the new hypotheses through the entire procedure in a continuous feedback loop. Because of the parallel processing capabilities and enormous power available from contemporary computing technology, these routines can hone and refine recognition accuracy tremendously.

- **Knowledge of the business process**

To ensure accuracy, a thorough knowledge of the business process of the user, as well as the invoice forms themselves, is essential. For example, with invoices, the vendor number typically is not on the invoice itself, but is assigned internally and kept in a separate database. After OCR, the vendor is identified automatically by matching the vendor name to a number in its own database. Within the business process, security considerations also enter the picture: business rules could require that high-dollar invoices be flagged and routed for manual review before they are approved for payment. In the case of medical claims, they must go through an adjudication process and be checked for fraud and illegal “bundling” procedures before reimbursement can be authorized and paid.

- **Foreknowledge about forms**

Because a static template cannot be used to recognize a functionally defined class of forms on the fly, prior knowledge about forms takes on extreme importance. In fact, prior

user knowledge about forms is essential to ensuring document classification and character recognition accuracy. Such knowledge could include, for example, the relationship between data and keywords, tabular structures and isolated fields, and information about data formats.

The more advanced knowledge about the forms that can be employed, the better. An invoice, for example, contains an invoice number and purchase order number, “bill to” and “ship to” address information, and a total, as well as a variable number of item data lines with product codes and/or descriptions, quantity, unit cost, etc. “Stuffing” the forms processing engine with this prior knowledge by integrating it with an existing purchase order database enables the engine to verify vendor information and compare purchase order amounts to invoiced amounts. This process narrows the range of the search and accelerates validation of recognized invoice data, which speeds up the data capture process significantly.

Once trained, an invoice processing system can be set up to learn about user documents as it goes along. In fact, some systems can be systematically retrained to take into account the elimination of old vendor’s invoices as well as the addition of new invoices from new vendors as these changes modify the processing set. Each system learns more each time it processes an invoice. It recognizes invoices from the same vendor and remembers the location of the required data, which improves accuracy and speeds up processing time with each successive invoice. Other software systems use neural network technologies to automatically learn invoices on the fly and store the results as a series of quasi-dynamic templates.

Processing unstructured forms involves taking multiple approaches that create alternatives that are compliant with classifying consistently similar data elements. It takes a powerful and sophisticated combination of document classification, image parsing, and character recognition algorithms to sort out the results correctly.

## **Chapter Three:** **Enabling Technologies**

Processing unstructured documents is nothing new to the field of forms automation. Techniques for processing “forms on the fly” were pioneered in mid-1990 by Nestor, Symbus (renamed Captiva), Mitek, Daimler-Benz (changed to OCE), MTI, and the now-departed TeraForm. Most of these companies typically used neural networks to analyze and generalize the image morphology of a particular class of unstructured forms such as faxes, invoices or medical claims. Advanced feature extraction techniques helped to facilitate page decomposition, and complex search routines located specific data fields on variable and problematic forms. The software could even find form data fields even if they were discovered separated from their expected locations by as much as a half-inch. However, due to the complexity of the algorithms and the expensive, pre-Pentium hardware requirements, attaining acceptable processing speeds at reasonable accuracy levels was costly; hence, during the 1990s, the software failed to achieve mainstream user adoption.

Today, however, given the incredible computing horsepower and memory that resides in the average desktop PC, a variety of techniques, including ICR brute force, can be run in parallel to achieve remarkable results. Blob analysis, edge detection, multi-line character segmentation, and long-line detection can be used to find form objects, columns, and data fields based solely on their topology. Sometimes the geometrical and spatial relationships between the text data elements, such as rows or subheadings (rather than graphical objects), are used to locate the places where data most likely will be found. In fact, the process need not involve character recognition at all; the text can be treated as a pattern of blobs.

Conceptually, there are two basic approaches to processing unstructured forms, located at opposite ends of the methodological spectrum. The document image understanding (DIU) approach treats the form primarily as a pattern of specific images, whereas the character recognition-driven approach treats the form primarily as a concatenation of text data elements. Neither technique is used solely by itself; contemporary vendors each use a hybrid version that uses techniques from both schools of thought.

### **The document image understanding (DIU) approach**

In the neural network-driven approach, the software engine analyzes, learns, and then generalizes the image morphology of

a particular class of unstructured forms, such as invoices. Neural network technology is used to train on a large training set of images consisting of hundreds, even thousands, of different samples of the same form type, in conjunction with feature extraction technology that distills and extracts a wide variety of topological elements—text lines, columns, graphical objects, and pictures—from the training samples. When linked to a conventional forms processing system, the DIU software engine facilitates page decomposition, which dissects and then automatically locates specific data fields on variable and problematic forms. The DIU engine also locates variable numbers of fields across variable numbers of columns, rows, or subheadings.

In the dynamic template-driven approach, the user defines a dynamic template or a map that describes the data zones contained on the form type in question. The DIU engine uses techniques such as blob analysis, edge detection, multi-line character segmentation and long-line detection to locate form objects and data fields. Defined characteristics could include the width of one data field relative to the width of another, the number of characters in a field, the size of an expected logo relative to the data columns on the page, and other factors. Fields can be detected at the pixel level with extreme precision, which allows for extremely fine separation of wanted and unwanted image elements, especially when locating tightly-packed data fields, such as EOBs. Once the data fields are located, they can then be sent to an ICR/OCR engine to be recognized and then validated.

Utilizing an intelligent scripting language approach, the end-user can employ a DIU engine to write a set of image parsing rules in a scripting language that guides the DIU engine in its search for data fields. The script is essentially run as a Visual Basic program that instructs the recognition engine on how to locate the required data. The script actually reads the image, using predefined information about intrinsic relationships—for example, the location of a “total” field relative to columnar lists of item values, geometrical location of data elements, such as corporate logos and address blocks, augmented by text prompting.

Another form of DIU involves focusing more on the spatial relationships between the text data elements rather than graphics in order to locate the places where invoice data is most likely to be found. The process of locating clues based upon text need not involve character recognition: the text can be perceived simply as a pattern of blobs. However, once the text is found, an ICR or OCR engine is used to recognize and validate the found data.

Processing does not stop once the data has been accurately recognized. Complex algorithms are used to sort out the recognized data and make sure that it fits the criteria for a given form type. Totals must be balanced, dates must be checked, and application-specific business rules must be followed in order to determine that the right sums and payment dates are reported.

### ***The character recognition-driven approach***

The character recognition-driven approach relies upon a powerful, OCR-based, literal search technique to locate and extract data from pre-designated regions on a form, and then processes that data as part of a user-defined routine. Instead of a special scripting language, users can set up routines using a special dialog box to edit, graphically map, and then view the procedures. Searches can be set up according to predefined data definition parameters, often exhibited as a flowchart, then the compiled results can be displayed in a number of graphic formats. Rules for accepting the data as valid can be established contingent on the confidence values of the OCR results.

The user can replace a defined character with another character, remove questionable flags, hide rows, or flag rows based on whether or not a character is found or not found. The user can also hide or flag rows, remove tables, and/or execute another procedure contingent on the results of the first procedure.

Conditions for launching additional procedures are based upon whether or not the criteria set for a primary or secondary search are met. This can involve applying specific business rules to the information found on invoices, (e.g., the length of a purchase order number must be greater than nine numeric characters, must begin with the characters "P1," and that payment terms must be net 30 days, etc.).

Almost all vendors support their form processing engines by

using a dynamic template library to speed up form identification and data element location. Using this method, the user designs a series of generalized form templates to help recognize the invoices received by classes of customers, grouped both by company and invoice format. These differ from the conventional, static, form definition templates in that they do not define data element regions by exact pixel location; rather, data location can be defined by general regions of the form, and by rules to follow once certain graphical structures or text elements are discovered at those designated regions of the form.

When a document is scanned, it is first checked to see if it matches a similar form type in the predefined dynamic template library. If it does, then the template can be applied immediately to find the data without a further search. If it does not, then the regular invoice analysis process is launched; but once the invoice is processed, the variations between that invoice and the nearest-matching template are noted and stored either as a new or variant template by the system for later use.

As mentioned earlier, none of these forms processing techniques is radically new. Vendors have been using them since the early nineties. Every vendor uses its own blend of techniques to optimize document processing accuracy. Many employ a multi-stage approach: they use one technique to recognize the form, and then another to locate the fields and their values before passing them to an ICR engine to recognize. Some vendors employ a neural network-based, DIU approach for doing form identification, and then apply a powerful, OCR-driven, literal search technique to find data elements where their neural network method leaves off. Others utilize still a third technique to validate found data values. There are also vendors that employ different methods simultaneously throughout the various data capture phases, in a voting scheme.

## Chapter Four:

### Completing the Processing Cycle

#### • Adding workflow to the process

Every unstructured forms processing application has a workflow component, necessary to successfully complete that application. For example, before payment for an item can be authorized in an invoice processing operation, it must be verified against a purchase order and/or other documents that describe the item, its price, and that the item has been received and accepted by the company that placed the order. As previously referred to, with respect to medical claims and EOBs, those forms must go through an adjudication process and be checked for fraud and illegal “bundling” procedures before payments can be authorized. Moreover, in the case of processing Medicare and Medicaid reimbursement forms, they must meet certain deadlines and turnaround standards, and might additionally be required to interface with a state EDI network to receive speedy reimbursement.

Setting up the correct workflow for an application demands a thorough knowledge of its underlying business processes, an understanding that is deep enough to enable the user to define explicitly all steps of the workflow and the exception procedures. Along these lines, there are two approaches for adding workflow to document processing procedures: (1) the vendor can supply its own workflow engine fully integrated with the forms processing software; or (2) the vendor can supply an application programming interface (API) for integrating its software with a number of existing workflow engines. Each workflow and business process for each unstructured forms processing application requires its own work process analysis.

In this regard, we raise a word of caution. Whether it involves training a neural network, setting up a series of dynamic form templates, predefining ICR thresholds, or integrating a forms processing system with a workflow engine to create a “total solution,” none of these tasks should be taken lightly. Despite the ease of a given user interface, no unstructured forms processing or workflow solution works smoothly “out of the box.” Setting up an efficiently running system for processing unstructured forms is at best a complicated and rather trying procedure that requires the assistance of a trained professional—either a consultant in process analysis with an ICR background, or a systems integrator that knows forms processing.

#### • Measuring accuracy

Interviews with industry vendors report that, when it comes to automatically processing invoices and purchase orders, per-document recognition accuracy can be achieved reliably and consistently at rates greater than 60%. This means that, out of every 1,000 invoices processed, today’s automated invoice processing software can be counted upon to classify, validate, and insert at least 600 of the documents into a company’s business process at a zero error rate with no human intervention required. The remaining invoices are rejected by the processing engine and passed to human operators to review and correct, because recognition results failed to meet user-defined parameters based upon preset ICR confidence levels.

Vendors report similar results with EOB forms and medical claims. The result is labor savings that can translate into at least a 60% reduction in overall data entry costs. For now, however, these numbers are purely anecdotal, for no independent, third-party, industry benchmark tests have been conducted to establish industry-wide performance standards.

The fact of the matter is that the specific mix of documents that a company receives in its incoming mailbox creates a document management profile that is a product of the type of business in which a company is involved and the companies with which it does business. Moreover, since America is in the early stages of user adoption regarding automated forms processing software, it is too early to know what generalities will hold true across the board with respect to estimating the kind of success that a given company can expect from automating, say, its invoice payments operation.

This brings up an important point: virtually every forms processing installation requires some customization, regardless of whether the forms are structured or unstructured. As previously mentioned, there is no such thing as a strictly “out of the box” solution—that is, if maximum accuracy is desired. Each forms processing installation has a unique profile—based upon variations in skew, print shop preferences, color of ink, paper reflectivity, and font degradation and other physical and design idiosyncrasies—that requires analysis and experimentation in order to optimize recognition accuracy. In other words, the complexity and variety of these factors will determine the extent to which the application must be tailored. Therefore, the best practice for a business user to follow when evaluating the potential effectiveness of an invoice

processing, claims processing, or EOB processing software solution is to request that a vendor run a pilot project using the client's own business forms. That way, the translation of laboratory theory into practical reality can be empirically tested and observed at low risk to the potential customer. It is important for the user to ensure that the pilot is, in reality, upwardly scaleable. The pilot must involve a random assortment of several thousand forms in order to enable the user to tune the forms processing system to a document profile that is upwardly scalable and truly representative of the installation. A pilot that does not accurately portray the variety of documents that a user processes daily is useless.

A vendor that wants the business will comply with a request for a pilot because they know that this is the price that they must pay in order to ensure successful early adoption.

However, a potential buyer should be prepared to define for the vendor the terms of success that, if achieved, will trigger their decision to purchase.

- **Performance Improvement Drivers**

Since 1996, the forms processing industry has been governed or driven as much by Moore's Law and advances in computer

technology as it has been by innovations in ICR/OCR technology or in new and sophisticated form classification algorithms. Because under Moore's Law, computing power exponentially grows by doubling every eighteen months, the enormous power available in today's PCs makes it possible for brute force to be introduced as effectively as elegance into the equation. Vastly increased computing power also enables forms processing software engines to run several different or orthogonal algorithms in parallel with each other, and even to feed the results into voting algorithms. As computing power increases, unstructured document processing can only continue to improve.

In the past, applications rooted in ICR-based, forms processing and image data capture technologies failed to meet great expectations of user adoption—largely because of the applications' narrow scope of use and perceived accuracy limitations. Now, armed with market-broadening capabilities of processing unstructured documents, mainstream application adoption is a far more likely possibility. Perhaps, at long last, a preponderance of end users will finally realize the promise of these remarkable technologies.

## *Major Forms Processing Applications*

<i>Market Segment</i>	<i>Form</i>	<i>Structured*</i>	<i>Unstructured*</i>
<b>Healthcare</b>	Medical Claims	X	X
	Explanation of Benefits (EOB)		X
	UBs (Series 80 & 90)	X	X
	Pharmaceutical Tests	X	
<b>Banking &amp; Financial Services</b>	Mortgage Applications	X	
	Loan Applications	X	
	Mutual Fund Shareholder Proxies	X	
	Checks	X	X
<b>Transportation &amp; Utilities</b>	Bills of Lading		X
	Custom Declarations		X
	Utility Bills (Remittance)	X	
<b>Retail</b>	Order Entry	X	
	Warranty Cards	X	
	Invoices		X
<b>Insurance</b>	Policy Applications	X	
	Policy Claims	X	X
<b>Surveys and Questionnaires</b>	Sweepstakes	X	
	Market surveys	X	
	Educational Tests	X	
<b>Government</b>	Internal Revenue	X	X
	Expense Requests	X	
	Court Documents		X
	Licenses and Permits	X	
	Violation Notices	X	

*An "X" in either of these columns indicates if that document application is predominantly structured or predominantly unstructured. In a few instances, there are significant proportions of each.*

*Arthur Gingrande, Partner, IMERGE Consulting*

Arthur Gingrande is a nationally acclaimed expert and pioneer in image-based intelligent character recognition (ICR), electronic forms, and forms automation. In 1988, he founded a neural-network development firm called Symbus Technology, now known as Captiva (San Diego), which has grown to become America's largest ICR-based, forms automation software firm. He is also the former director of marketing and business development for Nestor (Providence, RI), one of America's most prominent neural network software development companies.

Since 1991, over 200 of his articles have been published in various trade periodicals such as *KM World*, *AIIM E-DOC Magazine*, *Business Solutions*, *Integrated Solutions*, *Imaging Business*, *inform*, *Imaging and Document Solutions*, *VAR Business*, and *Imaging World*. The topics of these articles have included workflow, document management, electronic imaging, COLD/ERM, ICR/OCR, e-forms, e-business, CRM, knowledge management, wireless communications, processing unstructured form data, and content management. He has also written numerous white papers on those topics, including research reports for Dataquest and BIS CAP, now known as GIGA Information Group. Recent white paper clients include IBM, Tower Technology, SER Macrosoft, AIIM International, TAWPI, CharacTell, and Ceresoft. He has also written four patents in the areas of intelligent check recognition, ATM-based direct payment systems, and smart cards.

Mr. Gingrande is a founding publisher and former director of document and image management at ISIT.com, a website dedicated to integrated solutions in information technology. He is also editor and publisher of *Contemplor*, a newsletter

dedicated to document management and forms automation technology. He has participated in scores of industry trade shows and business conferences as a coordinator, guest speaker, panelist, and industry commentator.

Mr. Gingrande is the author of *Forms Automation—from ICR to Electronic Forms to the Internet*, published by AIIM International, a book about the role of forms automation in document management and electronic commerce. He wrote *Cost Justifying an ICR Solution*, published by The Association for Work Process Improvement (TAWPI). He also wrote *Technology Convergence, Document Management, and E-Commerce*, published by AIIM International, which shows the impact that the convergence of digital technologies—including the Internet, Web TV, wireless communication devices, EDI, and smart cards—will have on the application environment of the future. He also authored the AIIM white paper entitled *Enterprise Application Integration: Connecting the New Application Frontier with the Old*. Other publications include two white papers written for IBM on the topics of customer relationship management (CRM) and Web catalog management to introduce IBM's new WebSphere products in those areas.

As a partner of IMERGE Consulting in Arlington, MA, Mr. Gingrande has written the marketing or business plans for six of the leading software development firms in image capture and automated forms processing, which collectively make up 80% of the market. For IMERGE, he also consults to end-users in the areas of needs analysis and implementation oversight of automated document processing systems. Readers may contact him at 781-258-8181 or by email at [arthur@imergeconsult.com](mailto:arthur@imergeconsult.com).

## **AIIM International**

1100 Wayne Avenue, Suite 1100  
Silver Spring, MD 20910 USA  
800-477-2446 / 301-587-8202  
aiim@aiim.org  
www.aiim.org



For over 60 years, AIIM has been the leading international organization focused on helping users understand the challenges associated with managing documents, content, and business processes. Today, AIIM is the leading international authority on Enterprise Content Management (ECM). ECM is the technologies used to capture, manage, store, preserve, and deliver content and documents related to organizational processes. ECM tools and technologies provide solutions to help users with the four C's of business: CONTINUITY, COLLABORATION, regulatory COMPLIANCE, and reduced COSTS.

### ***AIIM provides:***

- **Market Education**—AIIM provides unbiased information through *AIIM E-DOC Magazine* and *mID* (Managing Information and Documents), the leading industry magazines in, respectively, North America and the UK; its 20-city Content Management Solutions Seminar in the U.S. and Canada; the IM Expo event held throughout the UK, and InfoIreland.
- **Professional Development**—This industry education roadmap provides a variety of opportunities. IM University is a multi-faceted program offered in Europe. The Web-based Fundamentals of ECM Certificate Program familiarizes users with the core concepts and technologies related to ECM. The AIIM Webinars round out user education on key issues.
- **Peer Networking**—Through chapters, networking groups, programs, partnerships, and the Web, AIIM creates opportunities that allow, users, suppliers, consultants, and the channel to engage and connect with one another.
- **Industry Advocacy**—AIIM, as an ANSI (American National Standards Institute)-accredited standards development organization, acts as the voice of the ECM industry in key standards organizations, with the media, and with government decision-makers.

## *Processing Unstructured Documents: Challenges and Solutions*

The majority of business documents (e.g., invoices, purchase orders, resumes, work orders, etc.) are arbitrarily structured and cannot be processed using conventional “ICR-friendly” document processing approaches.

Unstructured data accounts for nearly 80% of all corporate data on record. The overwhelming volume of information involved makes processing unstructured data for storage and consumption a major priority for most corporations.

Over the past few years, advances in automated forms processing technology have dramatically improved recognition accuracy in processing arbitrarily structured business forms.

This paper defines the “unstructured” document; outlines properties and procedures; discusses enabling technologies; and completing the processing cycle.



The Enterprise Content Management Association

### **AIIM International Headquarters**

1100 Wayne Avenue, Suite 1100

Silver Spring, MD 20910 USA

301.587.8202 / 800.477.2446

**[www.aiim.org](http://www.aiim.org)**