A Single View: Integrating Structured and Unstructured Data/Information within the Enterprise

January 2002
# Table of Contents

Introduction ........................................................................................................ 3
Business Direction, Challenges and Goals ......................................................... 4
Some Scenarios for Integration ........................................................................... 5
The Technical Challenges .................................................................................. 8
Strategies and Levels of Integration .................................................................. 10
Taxonomies, Schemas, and Metadata Mapping: The Specifics ......................... 13
The Unique Role of XML .................................................................................. 16
Integration Steps to Follow ............................................................................... 17
Conclusion ......................................................................................................... 19
About Unitas ...................................................................................................... 20
Contacting Unitas ............................................................................................. 20
Introduction

Corporations of all sizes and in all industries (including public agencies) are notorious for the amount of information they produce on a daily and yearly basis. According to a recent study, the volume of corporate data doubles each year and the public Web grows by over seven million pages a day. Because of this inexhaustible supply of information – about customers, internal projects, employees, sales and marketing, customer service, product development, operations, supply chain and financial performance, etc. – IT is tasked to make sure there is a coherent enterprise architecture and enterprise-wide information management strategy.

In a typical organization, only 20% of the information that exists is well formed – living in relational databases or legacy mainframe transactional systems. This information, which we refer to as **structured information** – often represents both the current lifeblood of the company’s production business applications (accounting, sales, operations, logistics, etc.) and a historical account of what has happened. Despite the fact that companies today spend millions (even billions in the case of some of the larger companies) to manage this information, it is often fragmented and is not integrated. The trend towards Data Warehousing was an attempt to add coherency to this information and make it available in a meaningful format to the right people – through reports.

In the same typical organization, 80% of the information that exists is often **unstructured information**, which means that it lives in documents rather than being organized in a database. These documents include email documents, memos, reports, proposals, spreadsheets, research, competitive intelligence and specifications, etc. and are often spread out throughout the corporation. Currently, the trend is towards unstructured content management, which is done in conjunction with an Enterprise Portal, as an on-going attempt to add coherency to this large sea of unstructured information, making it possible to search and even browse across the different documents and document types.

In addition to all of this information – which is produced inside the organization – there is a significant amount of information that comes from outside the corporation – news stories, reports from analysts, competitive information, industry information, regulatory information, third-party information, customer, and supplier and partner information – which is extremely relevant to a company’s on-going business and is often unstructured. This information is often strategic, because it provides the input from the marketplace that can impact the entire direction and overall success of a company.

This white paper deals with the challenge of bringing together unstructured content (both internal and external) with the structured information that is being produced by corporations.
Business Direction, Challenges and Goals

Why would a company want to integrate its structured and unstructured information? After all, companies seem to have gotten by for years without doing so.

The overall answer is that end users in all parts of the organization are facing both an **information glut** and a **knowledge shortage**. This is true for all of the naturally evolving communities both within the enterprise and in the extended/virtual enterprise - Employee Community (e.g., HR, Engineering), Customer Community (e.g., Sales, Customer Service, Customers), Supplier Community, and the Partner Community.

The **information glut** is because of the sheer volume of structured and unstructured information that is available. There is too much information, even within a department or team, for any one person to keep on top of it all. To deal with the information glut, users need to be able to filter and personalize the information that is relevant to them. For example, a salesperson might care only about information related to his or her target prospects and customers. This might include internal information, proposals, financial information, sales history, credit information, product information, buying patterns, as well as external information – about competitors in his accounts, etc.

The **knowledge shortage** is because although the sheer volume of information is large, it is often not presented in a meaningful way and it is spread out so it is impossible to see the information in context. For example, it is not easy to find customer service information related to one customer, while browsing another customer who is in a similar industry. Nor is it easy for one sales person to see some of the internal memos that were used in closing the original customer contract for another customer in a similar industry.

The end user is forced to search and assemble this information from multiple sources, in multiple formats. This task is often overwhelming – and so much of the knowledge that is relevant to a given situation (a sales call, a service call, or an employee review) is not present – resulting in the knowledge shortage.

Furthermore, both of these situations lead to a shortage of information sharing and collaboration across the corporation. Often, one division will have information that is relevant to another division’s problems, but because there is not a framework for sharing this information, it isn’t presented at the right time to the right people. The fact that the information is not integrated in a meaningful way to end users, means that collaboration and workflow, particularly across teams, departments or divisions, isn’t nearly as effective as it could be.

Lost productivity caused by not having access to and sharing information cost the enterprise millions of dollars per year. This lost productivity occurs in all of the communities, resulting in a substantial impact on product cycle time, logistics, delivery, customer service, and effective management decisions.
With this in mind, some of the business goals of integrating structured and unstructured information are to:

- Improved access, integration and management of information and improved the quality of decision making
- Improved productivity and workflow, reduced cost and achieve greater output with fewer resources
- Build tighter relationships, reduce cycle time and improve customer service and support
- Outsource and leverage non-core competencies to suppliers and partners, and collaborate with them on an on-going basis.

Some Scenarios for Integration

At Unitas, we often come across scenarios that require the merging of structured and unstructured content into a coherent, single view. These scenarios occur in almost all industries and examples range from insurance (customer-centric claim and risk assessment information) to retail (franchisee documents and financial information) to manufacturing (defect statistics and engineering specifications). The scenarios also occur in almost all functional departments, ranging from HR (employee performance reviews and employee logistical/financial information) to sales (customer-specific documents and sales information) to marketing (marketing program statistics and market intelligence) to finance (financial history of clients linked to comments, contracts, and proposals).

While the number of scenarios that require merging structured and unstructured content together is unlimited, Figure 1 and Figure 2 below show two scenarios that we often use as “representative examples”.

The common element in both of these scenarios is that the end user needs to maintain context not just across different databases, but also across different types of information. Thus, a marketing document or internal memo related to a particular customer, can be shown in the context of that customer’s order history and current prospects for new business, even though all of this information resides in different systems. More to the point, some of the information may not reside in any system in the traditional sense – it may just be one of thousands of documents sitting on a file server.
Figure 1 - Consolidating Customer Data and Information

Figure 1 above shows how consolidated structured and unstructured content might be presented in the customer-centric portal application. The different panes of the portal are referred to as Portlets (or gadgets, web parts, etc.). In this example, the end user might be a sales or service manager, or might be an individual sales rep. The end user can choose a customer in the “Customer” Portlet and see the following information on a single screen related to that customer:

- The Sales Opportunities Portlet (coming from a sales system like Siebel or coming from an order entry / SAP or Oracle ERP system) – This might also include links to unstructured proposals used for selling the previous deal.
- Sales Documents Portlet – This might include proposals, letters, presentations, etc. currently being used. This Portlet might respond dynamically to the sales opportunity that is selected in either the Customer or Sales Opportunities.
- Payment History Portlet (coming from an accounts receivables system) – This might include currently open receivables and also include links to letters sent/received, and comments recorded in relation to receivables.
- Customer Service Portlet (from a help desk system) - This might show currently open service requests from a company, contact, or customer. This Portlet might also show documents that are summaries of previous projects/customer trip reports, preventative maintenance schedules and parts, and project documents (status reports, etc.) from previous projects. These documents may be captured in a service system or may be stored on file servers and need to be coordinated.
- News Stories Portlet – This might point to current and historical news items (press releases, articles) that are relevant to this customer.
• Case Studies Portlet – This might point to case studies that the company has tracked that are in the same industry as this company.

Figure 2 below shows how structured and unstructured content might be consolidated in the asset management industry.

**Figure 2 - Using an EIP for Delivering Business Intelligence, in Context**

In this example, the end user is an asset/fund manager or an analyst. The portal pulls information from multiple sources (both structured and unstructured) and assembles them in a single view of a portfolio and the securities within the portfolio:

• The Portfolio Summary Portlet shows the consolidated value of multiple portfolios. Each of these portfolios contains individual securities, which are often kept track of by a portfolio management system.

• The Portfolio Detail Portlet shows the securities in the currently selected portfolio – this might show both the symbol and the full name of the security. This also would come from the portfolio management system.

• The Key Performance Indicator Portlet shows how the portfolio is performing in relation to other portfolios, or one or more indexes. This might include the integration of third party, structured data into the system.
• The Portfolio News Portlet shows new press releases and news stories, which are coming in a variety of unstructured formats, sorted into the appropriate portfolio. This is important for being able to summarize what is happening with all of the securities in a portfolio.
• The Search Portlet allows searching and browsing of Internal and External Research. This research is often unstructured, though there are new XML formats for representing investment research (such as RIXML).
• The Calendar Portlet shows upcoming and historical earnings calls and major events for the securities in the current portfolio. This also involves bringing in both structured and unstructured information.
• The Discussions Portlet can scan internal discussion databases (and other internal content) that have not been assembled into “official research” but mentions the securities that are in a portfolio.
• The My Alerts Portlet shows when complex conditions (that the portfolio manager specifies) are met. These conditions could include price movements (external structured data), news/press releases (external unstructured content), research reports (external semi-structured content), internal discussions or research (internal unstructured content), or portfolio level events.

These two examples show the power of combining structured and unstructured content into a “single view”, as if they were part of one system. The combination brings the information together into a coherent and complete picture, which allows the end user (i.e., a fund manager, a sales rep, a CFO, VP Sales or a CEO) to make effective decisions.

Enterprise Portals, because of their “Portlet” interface and ability to crawl multiple formats, are particularly suited to help integrate information. Effectively managed, the EIP can isolate the user both from the systems that the content/data comes from, as well as the type of the content. The distinction between structured and unstructured content goes away – as the end users get “all the information” they need, when they need it.

The Technical Challenges

There are clear benefits to bridging the gap between structured and unstructured data within the enterprise and presenting it to end users in the appropriate context. However, it is often easier said than done.

There are a number of technical reasons why it is very difficult to achieve this integration. There are also a number of less technical and more cultural reasons, both historical and priority-related, that have stymied this integration. In many cases, there are separate, on-going efforts to integrate the structured data with other structured data, and to integrate the unstructured content with other unstructured content.

Going forward, we must take a different approach to managing enterprise information. This approach requires a cohesive enterprise information strategy, which includes unstructured content. In addition to dealing with the volume of information, most IT organizations have not integrated the unstructured content into a cohesive enterprise
architecture. The unstructured content is often “tacked on” to an architecture as an afterthought, rather than being seen as a cornerstone of the company’s “Information” that IT is chartered to manage. Other examples of challenges related to integrating structured and unstructured information include:

**Integrating structured data is on-going challenge**
Many companies have not yet fully integrated their structured databases. This has created a very large market for Enterprise Application Integration (EAI) tools and utilities. This often involves integrating schemas from multiple business applications together, which can be a huge undertaking. A true EAI strategy would include both structured and unstructured content, but existing EAI tools and applications do not take this view.

**Unstructured content can come from outside the enterprise, as it often does**
Another complication with unstructured content is that much of the content needed by end users is actually produced outside the corporation – coming from analysts, third party news agencies, or public and private web sites. While it is theoretically possible (though extremely difficult) to control the format of unstructured content produced within the enterprise – it is almost impossible for an IT department to control the “internal format” of “external content”.

**Only a small amount of unstructured content is already integrated and classified**
In many organizations, the only integration of structured and unstructured content occurs in “point solutions” within specific departments. When this happens, it is usually as an add-on to a business application that relies on the end user’s categorization scheme and generally does not consider the content of the document. In the example of a CRM system, the proposals may be attached to a contact, division, company, or activity record. However, only a very small percentage of content may have been attached this way.

Another complication is that unstructured content may be organized into a Taxonomy, but this has usually been done as part of a point solution by a department. These “departmental taxonomies” must be integrated with an overall enterprise taxonomy, in the same way that different schemas exist for different business applications, which are eventually united in an enterprise data model.

**Different logical structure: Schemas vs. Taxonomy**
A technical complication in integrating structured and unstructured content is the difference between schemas in the structured world and taxonomies in the unstructured world. A Taxonomy (see section below on Taxonomies vs. Schemas) can be thought of as a way to categorize unstructured content and is often the only indexing that a company has of its unstructured content (beyond full text indexing, which has its own limitations and advantages). In order to integrate structured data with documents in a taxonomy, you have to define a mapping between the taxonomy and the schema of a structured database. However, taxonomies and schemas are inherently different, which means that some thought has to be put into how they can be mapped.
The search for meaning in documents and the lack of metadata

A major complication with unstructured content, as we have pointed out in our previous white papers, is that the unstructured content does not have easily identified metadata. The metadata about a document is usually how the document should be classified. For example, a proposal that has been created for a client like IBM, might have a metadata field called “Client” which has a value of “IBM”. This metadata can then be used to organize the document into the appropriate place.

Strategies and Levels of Integration

Despite these technical challenges, it is possible to integrate structured and unstructured content together. The best way to integrate structured and unstructured content is to have a general strategy for how your organization will accomplish this, and then augment that general strategy with specific techniques for specific implementations.

There are many ways to integrate different types of content. One way to classify these strategies is in terms of a three-tier architecture: Front End Integration, Middle Tier Integration, and Back End Integration. Below is an overview of the three approaches.

Please note that these are not necessarily mutually exclusive approaches – an effective integration strategy will have elements from all three tiers of integration. The approach that an organization uses is heavily dependent on the location and state of the unstructured content, because the rules for structured content are fairly well defined.

In fact, the most important element in deciding how to integrate is to look at the nature and location of the unstructured content. This is because the technical challenges associated with indexing and querying unstructured content are more complicated than querying and indexing structured data sources. Below is an overview of each “tier” of integration:

The Back End: Data and Content Warehouses

The first (and least used) approach is to put the integration as far back into the data tier as possible. This might be analogous to the concept of a Data Warehouse, but extended to unstructured content. In a data warehouse, the relevant content from different source applications is normalized into a cohesive schema and then migrated into the new database – the data warehouse.

This would mean taking the existing unstructured content and migrating it into an appropriate database-like format (which would most likely be an XML format). This repository of unstructured content can be thought of as a “content warehouse”. Smaller repositories of content can be thought of as “content marts”.

In most organizations, certain content may already have been migrated into “content marts”. Web based content management systems (for example, Interwoven, nCompass, Vignette, Stellant, etc.) generally store the content in a database rather than as a series of
This database generally has metadata available. The same is true for Lotus Notes databases, which usually have fields defined for metadata that are populated.

This tier (the back end) is not so much about putting content warehouses and data warehouses into a single repository – rather it is about migrating the unstructured content into a format that can be easily indexed and queried.

While this is an acceptable approach for some of the unstructured content, the sheer volume of unstructured content and the frequency with which any single piece of content is accessed, makes it unlikely that an enterprise wide content warehouse will be built. Most organizations will need to think about this “back end” content mart approach as a piece of their overall strategy, rather than the overall strategy.

**The Middle Tier: EAI and Content Indexing**

This is the tier where metadata plays its most important role, because it is used as a way to apply a structure across different physical and logical content sources.

At this tier, we can use the analogy of EAI – Enterprise Application Integration, and extend it to the world of unstructured content. In an EAI scenario, each back end structured database has its own schema and business logic. A middle tier is used to query and update the different back end systems. EAI relies heavily on “mapping of fields” from one database to the next, but does not require an explicit migration of data (as data warehousing does).

The analogy on the unstructured side is to have a middle tier that is capable of communicating with all of the different unstructured content sources. This middle tier must map the “content sources” to each other, and to the presentation layer. The content sources are left alone – but the index allows for virtual mapping of one document to another. So, for example, a record in a Lotus Notes database, which contains a summary of terms, is mapped to a proposal in Microsoft Word because they both relate to the same client (e.g., IBM). Once this mapping is accomplished, then linking both of these documents to the IBM records on the structured side becomes easier to do.

To accomplish this, the middle tier integration scenario consists of a series of services:

- **An Indexing Engine** - This engine allows full text indexing of unstructured content. This full text index is primarily used for end user searches, similar to searches of the web. One of the key advantages of corporate indexing engines is that they index more than just web pages, but also Microsoft word documents, groupware databases, PDF files, etc.
- **A Metadata Extraction Engine** - This engine works hand in hand with an indexing engine (and is often the same piece of software). This metadata extraction engine attempts to find metadata within the text of unstructured content. This metadata can be thought of as “fields” associated with unstructured documents. This is easier with content repositories (either XML or other databases such as Lotus Notes) than it is with fully unstructured documents.
• A Classification/Taxonomy Engine - This engine uses both the full text index and the metadata to attempt to classify the different documents together into a coherent taxonomy structure. The easiest way to think of this is that certain documents fit into certain folders. This might also use “concept-searching” technology from companies like Autonomy to classify documents based upon “concepts”, it might use full text searches, or more commonly it may use a metadata-based classification.

• A Querying Engine - A querying engine allows either end users or other programs to submit queries against the indexing, metadata, and classifications for unstructured content. This querying is vitally important to integrate structured and unstructured data.

• Lexical Mapping Engine - An important, but often overlooked middle-tier component is the equivalent of an ID-matching mechanism. This allows an entity from one system (the company IBM with ID 1234) to be mapped to different systems. This becomes particularly important with unstructured content, which may have the format of IBM as International Business Machines, or may not have as metadata the information about which division of IBM is being referenced. It is best to have this lexical mapping “learn over time”, which allows end users to “refine” the integration between structured and unstructured content over time.

• Security Infrastructure - While security is thought of as more of an infrastructure issue rather than a specific issue of integrated different types of data, it is critically important in all Portals, which seek to aggregate content from different back end sources. This includes authentication (login ids, passwords, single sign-on) as well as authorization (which pieces of content a user has access to).

The services provided at this tier call upon the back end integration (the “content marts”) that we spoke of above. Similarly, the front-end tier relies on these services in order to bring the right integration to the end user.

The Front End: Enterprise Information Portals and Business Intelligence
At this tier, the integration generally manages the context of the integration (which entity is being queried – for example which customer, employee, prospect). This context is translated into an appropriate set of queries (both structured and unstructured), which are then submitted to the middle tier query engines. When the results are returned, it is responsible for the display.

• A Conversion / Presentation Engine - Sometimes, the original format of unstructured content is not appropriate for display and a conversion engine is provided. The most important conversion is to XML, which can be transformed again into any other presentation format; however, to create meaningful XML from unstructured content is no trivial task.

• Context Management Engine - Context management is often done at this tier by the business application that needs the integration. Within an Enterprise Portal, which has multiple Portlets communicating with different back end systems, the context is managed by individual Portlets. This might also mean simply monitoring which node of the Taxonomy an end user is browsing and submitting queries based on that context.
• Aggregation Engine - This tier is also responsible for taking the results of several queries and consolidating them to the end user. This is particularly important if there are different indexing systems in place at the middle tier.

Taxonomies, Schemas, and Metadata Mapping: The Specifics

The integration between structured and unstructured content relies in the ability to map context from one to the other. In order to get into some detail, let us use a very simple example of a taxonomy and a schema and use this to demonstrate how the mapping would be done. Let’s assume that we have sales and marketing documents and related unstructured content. Let’s also assume that we have a sales system that is built on a relational database.

What is a Taxonomy, really?
Most companies think of a taxonomy as a series of folders and subfolders. This is the simple definition. In our example this might be:

• Sales
  o Proposals
  o Presentations
• Marketing
  o Case Studies
  o Press Releases

However, a Taxonomy is more than simply a set of folders - it is really a classification scheme for unstructured content. A classification scheme consists both of the buckets (or folders, subfolders) to put documents in, as well as the means to classify the documents.

This brings up the possibility of alternate taxonomies. The exact same underlying documents might be classified in another way, which is what makes a taxonomy interesting.

• Product A
  o Product A Case Studies
  o Product A Proposals
  o Product A Press Releases
• Product B
  o Product B Case Studies
  o Etc.

Yet another way to classify the documents may be by industry or by sales rep. The key to building a flexible taxonomy is actually not in the folders, but rather, the means for classifying documents into the taxonomy. This means is what allows us to have different taxonomies for the same underlying data.
What are the means for classifying documents?
In an ideal world, each document would be tagged with metadata (data about the document), which are often called document properties. It is these properties that allow us to decide how to classify the documents into the appropriate folders.

The way that most taxonomy engines (and Enterprise portals) work on classification is that they create a “virtual card” for each of the documents. This virtual card contains information about the source document (location, size, etc.) and contains its document properties. In our example, a virtual card might have five document properties: Client, Document Type, Industry, Product, and Sales rep. The Client field might have a value of “IBM”; the Document Type field might have a value of “Proposal”, etc.

Some systems would add a field for Folder, which tells the system which folder in the taxonomy the document belongs in. However, because taxonomies can change (and since multiple taxonomies exist), this should be thought of not as an inherent property of the document but a result of the classification scheme.

Once the properties are in place then the folder called Proposals simply has a criteria that says: “All documents whose Document Type=Proposal” are included. By focusing in on the inherent properties of the documents, the end user can browse ANY of the sample taxonomies that we presented and still get to the underlying documents in a logical and orderly way.

Why is a Taxonomy different than a schema?
A taxonomy is different than a schema mainly because it is a set of folders, which contain one or more cards (with each card representing documents). The cards are themselves like individual records of a relational database because they contain inherent properties of the entities. However, they are not like the records of a relational database in that they are only a representation of the actual data; not the data itself.

A relational schema consists of entities (represented as tables) and relationships between the entities (represented using primary and foreign keys). Each of the records in a relational table has a set of columns, which is relatively fixed. Taxonomies are relatively flexible and better represented in XML format (see section below on the unique nature of XML).

How can you map one to the other?
Because taxonomies are so inherently flexible, there are many questions related to the best ways to map a taxonomy to a schema. In a relational database schema, the tables are relatively fixed – they do not change often. A taxonomy, on the other hand, can have new subfolders added or deleted, as we have seen earlier. Therefore it’s not necessarily relevant to consider the folders of a taxonomy to map to tables in a schema.
The best way to map a taxonomy to a schema is to look at the document properties and use those as the basis for the mapping. The value of a property will map to the value of a column in a relational table. Therefore, the context is managed dynamically by looking at the properties of the document (or of the folder) and using those (in conjunction with a lexical mapping scheme) to query the other systems.

If the user selects a record for a particular contact from IBM, from a sales/opportunity tracking system, the front end “context management” system would submit a query to a back end lexical mapper, which would translate IBM into one or more synonyms. The context manager then submits the query to the query manager, which brings back one or more “result sets” of documents that match the currently selected entity.

Please note that you do not have to have metadata in order to query the unstructured content. You could do a full text index and a full text search to bring back all of the documents that match the keywords. However, this would require having a lot of different variations and could produce many more documents than the end user wants to see.

But how do you get the metadata?
So, the next question is, how do you get the metadata out of unstructured content so that you can map it to a SQL schema and do a query? This is, of course, one of the biggest challenges related to unstructured content. The most foolproof and also most costly method is manually tagging each document – the explicit method. This method actually works for new unstructured content (if you are able to put a solid content management strategy in place), but does not work so well for existing content.

It is part of the taxonomy designer’s job to look for both explicit and implicit metadata. Both of these rely on the fact that users have left “clues” to the metadata associated with the document.

For example, even though users may not have put IBM into a field in a document – it is certainly possible that document resides in a file folder structure already – it may be in a folder called IBM that might be in a parent folder called “proposals”. Or the file might be called IBMprop123.doc. Using the existing classification is one way to get implicit metadata. Many crawling engines help you to define rules for extracting this metadata, along with a manual process to correct the suggestions.

Some companies provide tools that categorize documents automatically based upon their content. Some use Bayesian algorithms for concept clustering, while others use more rule-oriented approaches. These generally start with an example document and then find other documents that are similar to this document. This also relies to a certain extent on human interaction – to train the system. Tools are just becoming available to extract XML based metadata out of unstructured content using both implicit and explicit methods.
The Unique Role of XML

XML has begun to be accepted as a universal format for the interchange of data between applications, systems, companies, and industries. Because of this broad acceptance and its inherent flexibility, XML can be a very important part of any integration strategy of unstructured and structured content. It can play multiple roles, including:

**XML can be used to represent metadata, including taxonomies and rules.**
First of all, XML is an excellent way to represent a taxonomy. Most programs that build taxonomies rely on proprietary structures to store the taxonomy folder hierarchy. Furthermore, the rules for sorting documents into the taxonomies are usually hidden away in the bowels of the product. XML Documents, which have a tree structure, are well suited to representing taxonomies, which usually include a tree structure.

XML documents can also be invaluable for the query engine to understand how to query different parts of the taxonomy and how to maintain context wherever in the taxonomy the user is.

**XML can be used to represent unstructured content and its document properties.**
XML is actually well designed for unstructured and semi-structured content. The flexibility of the tree structure, and the nature of textual documents make it an ideal way to represent both the properties and the content of a “document”. An HTML document, for example, is easily transformed into the more neutral XML format.

XML Documents have DTDs (Document Type Definition) or XSD (XML Schema Definitions), which define the formats of XML documents. By representing a back end document (which may be a Microsoft Word document or a Lotus Notes document, for example) as an XML document, you can rely on the DTD/XSD to understand the structure of that document. The DTD is a way to represent the properties that a document might have, in addition to the body.

**XML can be used to return results from different sources and aggregate them.**
Because XML documents are easy to transform into other XML documents, XML is both a great intermediary format for result sets, as well as a final format for result sets. Two query systems that return slightly different result sets can use XSLT’s (Xtensible Stylesheet Language Transformation) to put the results into formats that look similar. These XML streams can then be sent to the presentation layer, which can use yet another XSLT to transform the results into HTML so that they can be viewed by a web browsers.

**XML is used to implement integration services as Web Services.**
Another key aspect of XML is that it is the basis for most Web Services. Web Services is a way for one program to make a query from another program – that other program could be on a different machine on the same network or out on the Internet somewhere. The fact is that its XML allows the data to be translated and easily interpreted by the other side. An ideal scenario for integration would be to implement each of the middle tier and
back end services as Web Services that return XML formats. This allows the organization to stay flexible and introduce new elements into the integration strategy.

Integration Steps to Follow

Although we have reviewed issues and strategies for integrating structured and unstructured content, there is still no universally accepted method for doing this. The following list is an overview of the steps that you might take in approaching a project that involves integration of these two very different types of content. Please note that each project will vary somewhat from the normal roadmap. Ideally, this process would be followed on both a per-project basis as well as on an enterprise basis.

- **Business Requirements and Functional Analysis** - Defining front-end integration points and overall integration requirements. Perhaps the best first step in any complicated data project is to understand what some of the requirements are for integration. The requirements might be very specific for a particular end user community, or might be to come up with a more general strategy for unifying classes of data. Both the business and functional requirements need to be fully understood.

- **Technical Analysis of the “as is” Environment**
  - Analyze your existing unstructured content. This is a very important step, and sometimes may have been done already. Many organizations have implemented point taxonomies but haven’t thought about them. This step includes analyzing the existing taxonomies as well as the existing unstructured content.
  - Analyze the underlying schemas of the structured data. This is similar to the analysis that might be done in building a data warehouse, data mart, or enterprise data model.
  - Conduct a high level analysis of the security models and mapping issues between the different types of content.

- **Integration Design**
  - Agree on the overall technical and integration strategy. This includes understanding roughly where the integration will be done – and how much work is appropriate at each of the tiers – front end, middle tier, and back end.
  - Design the technical mapping of the schema to an ideal taxonomy. This can be done by identifying new “entities” that represent the unstructured content that can be included in enterprise data models. They are “virtual” entities because they exist outside of the relational structure. This will define the “ideal taxonomy”. Then, using the methods we presented earlier, map the taxonomy (through properties in documents and folders) to the schema.
  - Develop the roadmap for migrating unstructured content to new taxonomy. The new “ideal” taxonomy for mapping may not be currently available. This step is to develop a plan – both for the current project, and
on an ongoing basis – to migrate existing content and an on-going content management strategy.

- Design technical middle tier/back end services. This is the design of the application services – query engine, lexical mapping engine, security services, etc. Ideally, each of these will be done as “Web services” which return XML and can be invoked using protocols such as SOAP.
- Define a front-end aggregation strategy. This includes taking the results (XML) from one or more middle tier services and integrating them with the front end. This also includes the context management strategy that the portal will use to maintain context between different types of content.

**Implement Content Migration**

- Refine the taxonomy and content management strategy to be consistent with the integration requirements. This includes re-indexing, migrating metadata into new formats, cleaning up existing data, and most importantly, defining a content management strategy that ensures that the correct metadata will be filled in as new content is developed. For more on developing a Content Management strategy, please see the Unitas White Paper, *Managing Unstructured Content in EIPs*.
- Implement a Lexical Mapping content database that will be used for mapping the name of entity in one system to the other systems.

**Develop Application Services and a Front end Aggregation Mechanism** - This is the actual “software development” part of the project. This include developing various middle tier services, including the context manager, the querying engine, refinement engine, as well as the metadata that will allow the middle tier to index and query the different content sources. It also includes the software for aggregating XML from different sources.

**Test and Train the Mapping for Integration, Starting With the Beta Cycle** - No integration will be perfect. A key part of the beta phase should be to “train the software”, particularly the lexical mapping engine to ensure that the correct documents are linked in the end user interface. The beta stage here includes both software testing and content testing.

This is a high level overview of how the integration might be done. In a real world scenario, there are of course, many intermediate steps and phases that we have not included here.
Conclusion
Organizations can benefit from bringing information from diverse systems, applications, and content repositories together onto a single Enterprise Information Portal screen. An ideal EIP will be able to maintain context across different types of data – finding for example news stories and research about a particular stock, and presenting those in context with the financial information related to the stock. This ability to bridge context between not only different systems but also different types of information is a key part of the promise of a “single, unified desktop” that EIPs provide.

One of the most challenging obstacles to this single view is the nature of unstructured content and how it can be mapped to data in existing structured databases. At first glance, the nature of a taxonomy and the nature of a schema are different. But by delving into the underlying definition of a taxonomy, the proper mapping can be done. Because of the diverse nature of both structured and unstructured content sources and the applications that use them, there are many different ways to implement an integration. An organization looking to integrate these different types of content – whether for a particular project or as part of an enterprise information management strategy – should pay particular attention to partitioning the work across different tiers and using XML because of its inherent flexibility in representing different types of content.
About Unitas
Unitas Corporation, based in Waltham, Massachusetts, is a leading consulting and services firm in the rapidly emerging enterprise portal market. Unitas fuses business strategies and information needs with structured and unstructured information, content sources, business intelligence, portal technology and integration to back-end disparate transactional systems. We define, develop, implement and integrate information solutions for the employee, customer, partner and supplier communities. We also help Fortune 1000 clients to leverage their existing investment in IT, reduce cost, increase productivity and to improve the quality of information and customer relationships and service.

Contacting Unitas
For more information about other portal white papers or “portal buzz”, please contact:
Unitas Corporation
404 Wyman Street, Suite 320
Waltham, MA 02451
Phone: 781.522.9300
Fax: 781.522.9350
www.unitas.com

Survey
Did you find the information in this white paper useful? Please take a few moments to give us feedback on this white paper and future white papers that would be of interest to you by filling out our on-line survey under the ‘white papers’ section of the Unitas web site at www.unitas.com