

The Document as Application

The Convergence of Document Publishing and Application Development

A JUSTSYSTEMS WHITEPAPER

Two previously divided worlds are converging: document publishing and application development. Publishing processes have been focused on static documents—print, PDF, HTML, etc. As the web has become a richer and more interactive commerce and branding platform, new content formats—so called, rich media—have emerged.

However, in most traditional business process domains, document publishing has remained focused on static artifacts. Policies, procedures, technical manuals, product design documents, and other documents key to business operations are, today, published as static documents.

Of course, the process of implementing change has been streamlined and accelerated through versioning and other lifecycle services and XML-based componentization of the content itself. Yet the published documents are still just snapshots in time. As time passes, a static snapshot no longer reflects current realities, fostering mistakes as well as unnecessary costs and liabilities. In many process areas—particularly where accuracy is absolutely vital—this simply is not good enough.

How, then, do organizations access live, interactive information? With traditional business applications such as enterprise resource planning, supply chain management, product lifecycle management and business intelligence. On the other hand, documents have been the domain of static information. But that is beginning to change—the worlds of document publishing and application development are converging and the document, itself, is becoming the application—with live data and an interactive user experience.

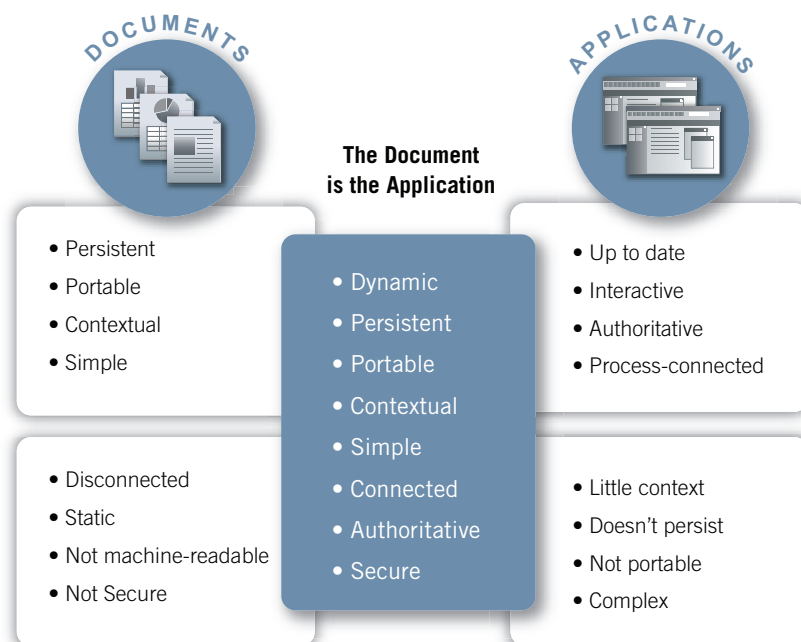


Figure 1. Two previously divided worlds are converging: document publishing and application development can now provide live data and an interactive user experience.

Why Do Documents Matter?

So, why aren't business applications enough? Why must documents be live and interactive? Unlike portal-style business applications, documents are portable, persistent and self-contained as bounded artifacts. They represent fully contextual views into information, which is organized with a deliberate intent and purpose. They are self-describing and they exist to tell stories, relating disparate pieces of information into actionable knowledge. The view into a business application is ephemeral, an ever-changing, disconnected view into whatever a user wants to see at a point in time.

Moreover, many business processes are document-centric—they require the persistence and context of a document. For information workers, documents are how knowledge is transferred. Documents are how information is communicated when it must stand alone. Try making any sense of the columns and rows in a database. It lacks context and persistence. Designed for machines instead of people, a database provides information, not knowledge.

Business applications provide some degree of context for data, but they are not persistent. The view they present is fleeting and episodic. This makes portal-style business applications a poor substitute for many processes.

For example, a technical manual for maintaining the hydraulic system of a commercial airliner; or the standard operating procedures for powering down a nuclear power plant; or the technical specification for building an application on a network of embedded processors. Such information requires the context and the persistence of a document, but these documents are subject to near-constant change. The cost of putting inaccurate or out of date information in the hands of the end consumer is untenably high; it can lead to rework or redesign costs, launch delays, noncompliance with regulations—or worse.

Given their limitations, why count on static documents to communicate vital information? In part, because the most dynamic information is walled off from the documents themselves.

From Data and Documents to Data in Documents

Traditionally, the domains of documents and data have been isolated from one another. Data is stored in relational database, mainframe systems and data warehouses. Documents are kept in content management systems, shared file servers and local drives.

Structured data is typically focused on the “what” of a business—financial information, inventory, etc. Documents tend to focus on the “why” and the “how”—manuals, policies, reports, analysis, etc. Data is structured and empirical. Documents are unstructured and contextual.

The reality is that business is done at the intersection of “what,” “why” and “how”—where fact meets context. Many organizations now recognize this artificial separation and are seeking ways to unify these two worlds.

Middleware technologies are emerging to help organizations federate access to data and documents—side by side—as part of a unified application. That is an important step forward, but what if the data is actually part of the document content? What if the data needs to be viewed within the document itself?

For example, a technical field service manual for complex capital equipment; a recipe for a drug or other chemical compound; or a pilot’s electronic flight bag. These are examples of documents that include data—data stored in relational databases and other sources outside of the document domain.

Viewing the data and the documents side by side “on the glass” is better than nothing, but the logical user experience is in the document itself. The data and document convergence must transcend data and documents and support data in documents.

Dynamic Documents and Document-Based Applications

XML-based structured authoring and publishing have emerged to allow organizations to more rapidly propagate change to unstructured documents. But data that originates in structured databases has no native connection to documents themselves. Data that is copied and pasted into a document becomes obsolete almost immediately, which instantly devalues the document and has the potential to introduce significant cost and risk. Structured data in documents needs a direct and persistent link back to the native source, so the documents are as accurate and authoritative as an organization’s systems of record.

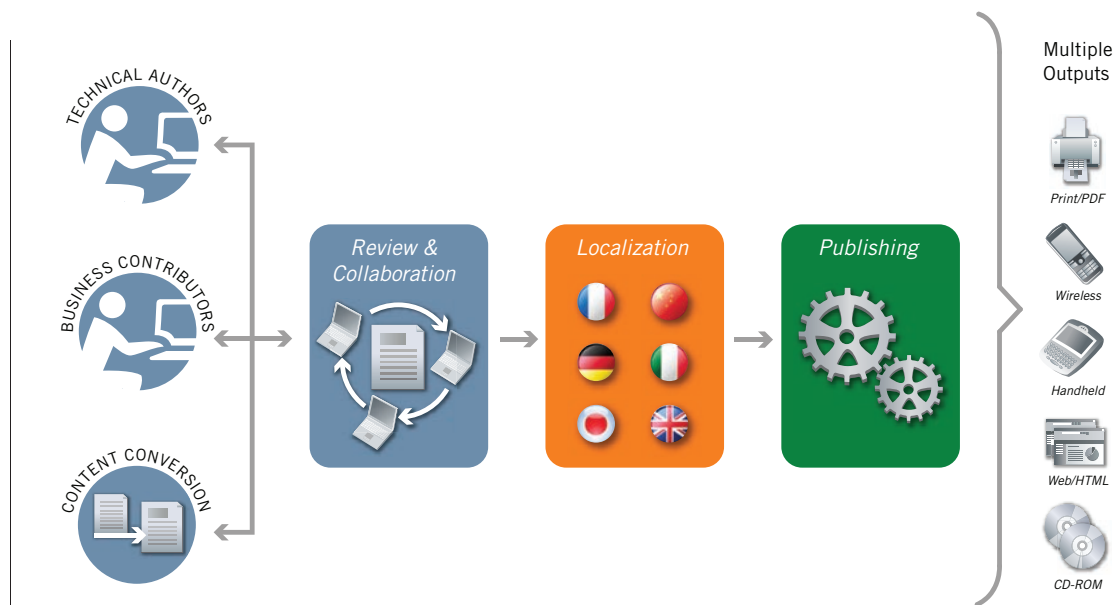


Figure 2. XML-based structured authoring and publishing allow organizations to rapidly propagate change to unstructured documents. Structured data in documents needs a direct and persistent link back to the native source, so the documents are as accurate and authoritative as an organization's systems of record.

How do people interact with documents? They make inline edits. They make annotations and comments. They initiate a workflow process or act on a workflow step. They e-mail attachments. They review, approve, update. They change the lifecycle state and declare records. All of this interactivity is limited to the document itself.

What if they could interact with a document the way they interact with a business application? Invoke a query. Execute a transaction or perform a calculation against a backend data source. Update a field within a system of record or run a report or create a chart based on live enterprise information—all within the document, without ever losing context or persistence? Organizations have never had this sort of interactivity within documents—but that is changing as the worlds of documents and business applications collide.

Sometimes, the document will live on as a dynamic entity—a dynamic document—constantly changing and constantly allowing change. Field manuals and technical datasheets are good examples of this. At other times, this interactivity will be limited to the creation process—bring together data and content with the explicit goal of creating a static artifact.

For example, the creation of a proposal will certainly benefit from dynamic document creation—integration of pricing data, using check boxes to automatically add or remove standard sections, and reusing standard descriptive language and terms and conditions. On the other hand, publishing the proposal marks an event in time and the document must become static.

Clearly, the lines are beginning to blur between document publishing and application development. Many processes require the persistence and context of a document and the dynamism and interactivity of a business application. They require structured data to come together with unstructured content—not only on the glass—but as part of a unified document.

For many document-centric processes, the document becomes the new application context. Organizations must no longer make the tradeoff between the live data and interactivity of a business application and the persistence and context of a document. The document is the application.

The Document as Application in Action

From financial services to manufacturing, life sciences, and beyond, the document as application—or dynamic document—promises tremendous advantages in information sharing and collaboration, document process transformation, and elsewhere. The scenarios below reflect a sample of the possible use cases.

Constant Information Flux

In some cases, making static documents current is simply impossible when they depend on a complex array of structured and unstructured source information that is constantly changing. As soon as one data element changes, the whole document needs to be republished. The dynamic document is the only delivery format that accommodates a constant state of information flux.

An airline pilot's electronic flight bag or interactive electronic technical manuals (IETM), are perfect examples of such rapid, constant change. The pilot and the maintenance crew depend upon the most up to date information. The cost is too high to deliver anything but the most authoritative information.

The challenge here is two-fold. First, ensuring that the documents are absolutely up to date as they depend upon hundreds or even thousands of data elements that may be changing. Second, ensuring that only the most relevant information is provided, so people in complex roles under pressure to turn the aircraft at the gate do not have to comb through pages and pages of irrelevant information.

Dynamic documents ensure that the information presented is literally always up to date—not since the last publishing, but since the last refresh or local print. Dynamic documents also store the application logic in the document itself, so it is highly aware of its environment—imagine a document invoking a web service query about local weather conditions and dynamically rendering the policy and procedure for icy conditions, windy conditions, northerly winds, etc.

Extreme Information Collaboration

Dynamic documents also deliver value in more collaborative processes, such as a manufacturer's sales and operations business processes. Here, sales, marketing and operations departments jointly grapple with issues that cross departmental boundaries, e.g., forecasting demand, coordinating production, and making trade-offs in profitability by dynamically matching demand and production.

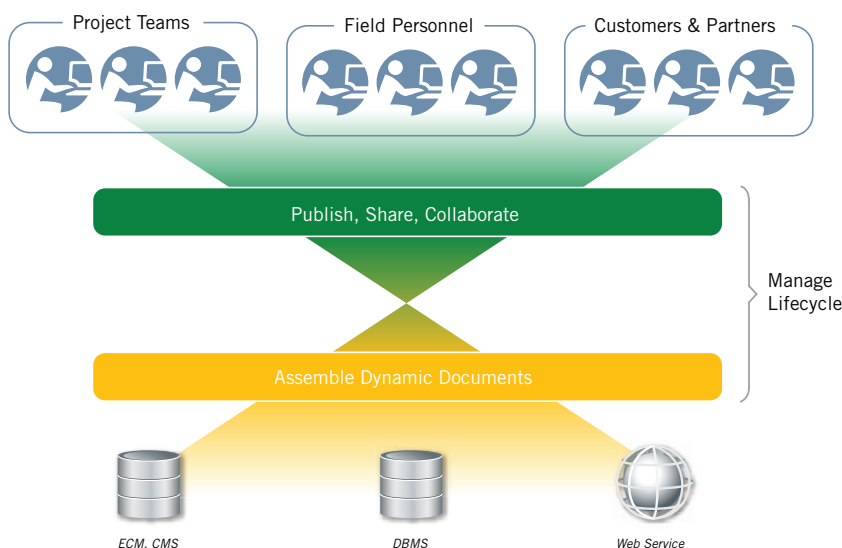


Figure 3. Dynamic documents are the basis for collaboration ensuring that information is always accurate and up-to-date whether working collaboratively internally between departments or externally with partners.

Today, this process relies on multiple static spreadsheets with nearly constant confusion about the latest versions. With a dynamic document, each participant gets the latest version with up-to-date XML links to pricing, procurement and other variables stored in ERP, CRM, SCM and other enterprise systems of record. No more manual reconciliation and validation and no more anxiety about whether the information is right.

Another collaborative issue dynamic documents address is seamless information sharing in trading partner relationships. Imagine a high-tech manufacturer with hundreds of suppliers and with a contract manufacturer that actually assembles the manufacturer's product—a common business model in high tech.

The manufacturer has supply chain management software, but not all of its suppliers do. Instead of requiring every supplier to install costly supply chain software, the manufacturer utilizes dynamic documents, so they can collaborate around demand forecasts, inventory data and production schedules. A supplier receives an e-mail from the manufacturer with the dynamic document attached. The supplier uses the dynamic document to access the latest information, spots a potential shortfall in the manufacturer's inventory and quickly addresses it before it becomes problem.

Likewise, the same high-tech firm could seamlessly share forecast data with its contract manufacturer in a dynamic document. In the same document, the contract manufacturer could share its capacity data and production schedules. Manufacturer and contract manufacturer could collaboratively align capacity with demand to maximize revenue and minimize cost and risk.

Bridging Information Silos

In many business processes include manual gaps between isolated silos of automation. These manual gaps are typically the document-centric phases of a process. Dynamic documents help to streamline business processes by bridging the gap between automated transactional workflows and more human-centric document-based workflows.

Organizations can automatically capture transactional information and pull it into dynamic documents for human review and analysis. Likewise, information within documents can be automatically processed by back-office systems or packaged and delivered as an XML stream without manual re-keying or scanning of information.

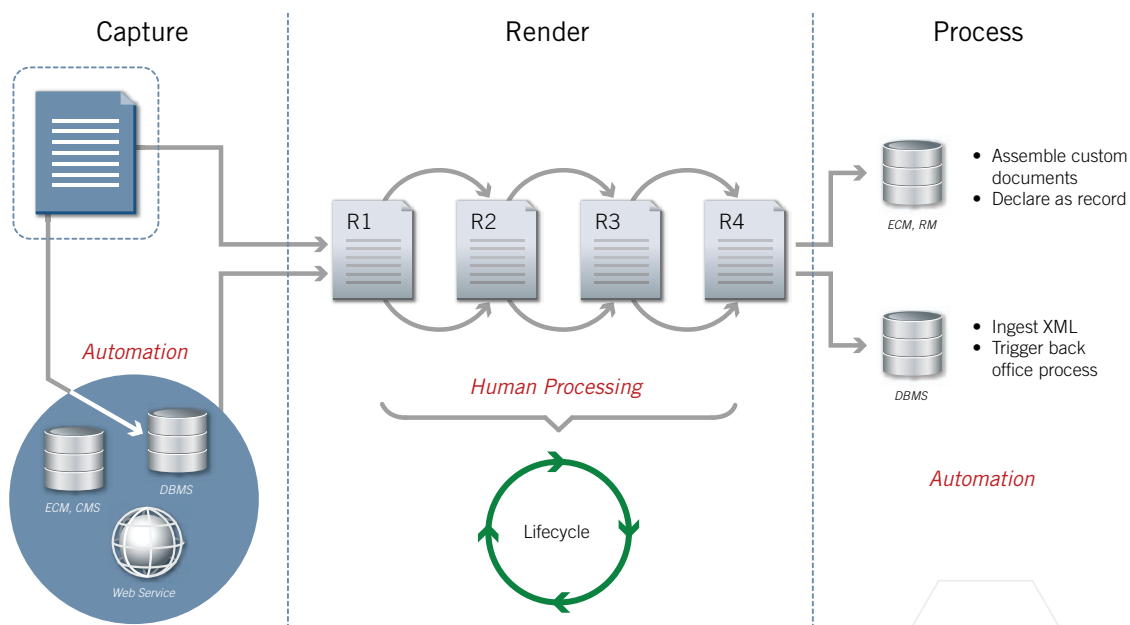


Figure 4. Streamlining business processes by connecting transactional business processes and document-centric business process flows accelerates process cycle times, increased process throughput and eliminates processing errors.

This document process transformation can be illustrated by a request for a bank loan. The initial step in this business process is often a form filled out by the applicant. Suppose the form becomes the basis of a dynamic document. Some information in the dynamic document is handled as a transaction within an enterprise software system. Other data from the dynamic document becomes part of a series of dynamic document renditions for human review and analysis across various steps in a workflow process.

Later in the process, the loan is approved and the information in the dynamic document is extracted to trigger and populate an automated transaction again, handled by back-office systems—for example, the account opening procedure. In turn, the dynamic document eventually needs to be rendered as a new set of static documents, such as an acknowledgement letter, payment coupons, etc.

Many processes require the persistence and context of a document and the dynamism and interactivity of a business application. They require structured data to come together with unstructured content—not only on the glass—but as part of a unified document. For many document-centric processes, the document becomes the new application context. Organizations no longer need to make the tradeoff between the live data and interactivity of a business application and the persistence and context of a document.

The document is the application.

JUSTSYSTEMS.

<http://www.justsystems.com>

North America:

+1.866.793.1542

contactsales-na@justsystems.com

Europe:

+44.0.1462.889.082

contactsales-emea@justsystems.com

International: +1.604.697.8705

About JustSystems

JustSystems is a leading global software provider with three decades of successful innovation in office productivity, information management, and consumer and enterprise software. With over 2,500 customers worldwide, the company is continuing a global expansion strategy based on its xfy enterprise software, XMetaL content lifecycle solutions, and its pioneering work in enabling XBRL financial reporting technologies. JustSystems is one of the 2008 KMWorld 100 Companies that Matter in Knowledge Management, a 2008 EContent 100 member, and was recognized on the 2008 KMWorld Trend-Setting Product list for XMetaL. Major strategic partnerships include IBM, Oracle and EMC.

© Copyright JustSystems

All rights reserved. JustSystems, the JustSystems logo, and xfy are trademarks or registered trademarks of JustSystems Corporation in the United States and other countries. XMetaL is a trademark of JustSystems Canada, Inc in the United States and other countries. Other company, product and service names may be trademarks or service marks of others.