Introduction

A document is a free flowing, persistent, self-contained, and unstructured static block of text written by an author to meet a particular objective. The objective could be a report, contract, letter, proposal, or some other content to meet a particular business need. Documents frequently need to include data and content that are contained in different corporate information systems. Most document authors will use corporate applications and other assets to manually seek out the required data, compile it, and then incorporate it into the master document.

Up until now, the two worlds of “document composition” and “application development” have remained disconnected. However, this is going to change. This paper discusses the need for, and practical concepts behind, “transforming a document into an application.”

Figure 1. Transforming a Document into an Application
The Challenge (Problem)

The challenge is to gain better control of the document composition process to improve responsiveness, completeness, correctness, personalization, efficiency, and integration with other available data sources.

Effective document composition should allow for real-time collaboration between content authors and reviewers. This need is more immediate in document-centric industries like such as law, insurance, health care, finance, and government, where expertise is often distributed throughout an organization. These organizations generate a multitude of documents including letters, legal contracts, reports, invoices, proposals, and more. The majority of these documents are manually generated and assembled by full-time staff one at a time.

Since the advent of the internet, creative people all over the world have built many smart and sophisticated applications—including CRM, supply chain management, and data management software tools. Indeed, people have conquered some of the most vexing structured data, workflow, and presentation challenges. However, while important progress has been made in taming unstructured data, word processing remains, for the most part, human-centric and static. While we may be able to compile a document from various data sources, its composition becomes disconnected when several individuals are responsible for updating and maintaining it. We have built smart forms that are connected to various heterogeneous sources, but when it comes to a document, it remains disconnected after it is created.

So, how can this change? Can a document stay connected with enterprise data sources forever and be more dynamic to address the ever-changing business needs of organizations? The answer is yes. Before I address this, however, it is important to acknowledge why we can build better document composition solutions at this point in time.

Why Now?

Several software companies have addressed this challenge, but most of their solutions are desktop-based. Full-time connectivity to central enterprise data resources is not practical, unless a persistent connection is maintained back to corporate resources. Most vendors choose the client-server approach because very little work was done to define the open standards required by Web-oriented technologies. However, this has changed. Thanks to emerging open standards, such as Open Document Format (ODF) and Office Open XML (OOXML), we now have the capabilities to build Web-based, robust, and rich document assembly solutions.

To draw a parallel, this challenge is similar to trying to watch a movie in High Definition (HD) before the HD standard (e.g., “Blue Ray”) was defined. Up until that point, not many movies were released in HD format because producers were not aware of the compatibility standards for most DVD players (there were competing standards not long ago). Note that the concept of HD has been around for a while, but until a common and universally-accepted standard was defined, it was hard for universally-compatible DVD players to be built and equally hard for movie publishers to release HD movies that could be played on any HD DVD player, regardless of the brand.

The Solution

The solution simply lies in “transforming a document into an application.” A document does not have to be composed on a desktop. Internet and browsers have evolved to allow for an equally satisfying document authoring/editing experience on the Web (e.g., Rich Internet Applications, Silverlight, Flex and HTML 5.) It is safe to assume that the rich experience of desktop-based word processing is closer than ever, readily available on browsers and the Web. Google Docs and Office Web Applications are just few examples of this capability.

What does it take for a document to be transformed into an application? Project Performance Corporation has built a solution for one of our key customers using Open XML, SharePoint, and Asp.Net. It is
important to note that there are quite a few other tools and technologies available in the market today to build such solutions. Regardless of the specific tools, the underlying foundation behind any document composition solution should address seven core characteristics and concepts:

1. **The document should be aware of its environment or container.** The container can simply be a folder or other metadata. For the purposes of this discussion, let us introduce the word “project” to signify a group of unique metadata attributes. These attributes can define some of the unique content in the document. For example, if we are building a legal contract as our document, then the project metadata attributes could be something like party name, contact information, address, state location, any applicable laws, etc. The idea is that any document placed in a parent “project” container can automatically update its metadata fields and have that data applied to the document content. This helps ensure correctness, completeness, and efficiency.

2. **The document should be able to behave as a user interface (i.e., “document as form”).** Besides project metadata awareness, a document should allow for connecting to enterprise resources with a form-like interface. A form is typically a rigidly controlled vocabulary construct with fixed labels in which the form controls the allowance of user input. In a word processing environment, we probably need both—we need to allow users to interact with central enterprise data sources with a controlled vocabulary, but still have the ability to change or overwrite the text.

3. **The document should behave as a data structure or template.** As an example, we may be able to bind Outlook contacts to a document through the “document as form” interface. This would require setting up a corporate paragraph or clause library for users to choose from or simply allowing each document to be saved as a template for future use. Templates are probably the single most important starting point in our attempt to transform a document into an application. We cannot achieve structured application behavior in a document without a template. Templates are the blueprints of what is to come, and is the starting point for a document to serve as an application. It is in this template that all the intelligence about the data source bindings, document as a form concept, and placeholders for any other corporate intelligence reside. For those of us familiar with object-oriented programming, a template is analogous to the concept “class” and each document is analogous to the “instance” of that class. Thus, the document properties are pre-defined in the template and it is only the values that change for each individual document/draft. Authoring a template should be well thought through. The actual template and its constructs may change for each organization, but the underlying process should remain the same. Much of the intelligence (i.e., data source connection and other corporate intelligence placeholders) is embedded into the template.

4. **The document should embody a rich online editing experience.** A rich “What you see is what you get” (WYSIWYG) online editor should be carefully chosen. It may need to be customized and extended to reflect the template placeholders and subsequently the “document as an application” interface. One should consider support for real time track changes and other word processing capabilities when choosing the WYSIWYG editor. Fortunately, there are many tools on the market that provide this functionality.

5. **A document management system should be in integrated in the solution.** This is needed to manage the documents that are created by various users. There are quite a few products in this space, including Documentum and SharePoint to name two (although they differ in their functionality and scalability.)

6. **The solution should allow for real time collaboration with tracked changes.** Given that a document (or any working draft) is created on the Web through a browser and stored on a centralized server, the solution should provide real-time online collaboration in and on a document. In the past, most of us have used emails to exchange documents when we needed to capture individual edits and comments. We may have even kept the MS Word track changes option on and circulated the document to various people; however, all cases required us as authors to reconcile the various inputs received. While a simple document management system may maintain version control, it is still a manual process to reconcile each version and its revisions. Given that documents can now be authored on the Web, collaborating with other
individuals should naturally be embedded in the document authoring process. Also, I believe “simultaneous editing” is critical as it allows two or more users to work on the same document at the same time. Google Docs is a good example of this capability.

7. **The solution should seamlessly support an online/offline editing experience.** An author should be able to start editing a document online, make use of all the application aspects of the document, connect with heterogeneous sources as defined by its template, collaborate online, and more. At any point in this process, the author should be able to download the document for offline use. In the offline mode, the document may be edited just like any other MS Word document and when done, uploaded back online to be “connected” again to corporate information systems. This capability may be further extended to provide two-way “Data Synchronization.” This means that if the author enters a new word in the document at the appropriate placeholder, and if that entry is missing in the underlying connected corporate source system, the solution can be designed to insert/update the source system. In short, a document can be used as a data-entry point to back-end information systems.

## Conclusion

The two previously-divided worlds of "document composition" and "application development" are converging. As the Web becomes richer and more interactive, and open standards like OOXXML and ODF are being adopted by more and more word processors, it is natural to build solutions that can integrate the context and self-describing nature of a document with the interactivity and dynamism of an application. In other words, the storytelling and actionable knowledge in a document can now also be up-to-date, real-time, and interactive—while reflecting evolving business needs. Transforming a document into an application provides the best of both worlds. Project Performance Corporation has demonstrations and videos available to show how this can be done using existing technologies.

## About Project Performance Corporation

Project Performance Corporation (PPC), part of the AEA group, is a management consulting firm offering world leading expertise in environmental and IT and management solutions for top government, non-profit, and private sector decision makers worldwide. At PPC, we solve a wide variety of problems for clients by helping them improve the way their organizations function. From leading-edge solutions for optimizing the use of information and institutional knowledge, to highly effective solutions to address energy usage and climate change, we specialize in delivering effective, reliable answers to the most complex challenges. At the heart of it all are our people — innovative thinkers well versed in business processes and drivers, and an unequalled focus on delivering quality products and services.

## About the Author

Mr. Gupta is the Director of Enterprise Services for Project Performance Corporation with over 13 years of demonstrated thought leadership in building cutting edge solutions. A multi-faceted professional, with a deep grasp of technical as well as business consulting, Mr. Gupta is responsible for providing leadership and direction to corporate information technology consulting business. Among the multiple roles he has played are practice leader, global implementation manager, delivery manager, project manager, strategy consultant, and chief technical architect. Mr. Gupta has been instrumental in building much of the information technology capabilities at PPC. His expertise includes business development, corporate development, practice management, IT road-mapping and strategy, product development, software engineering, delivery management, systems integration, SOA/WOA consulting, and enterprise architecture.

Mr. Gupta leads several key technology initiatives including **online, collaborative Web-based document composition.** He is a frequent speaker and has released many demos and videos, including the one at Microsoft’s Channel 9.

**For more information contact:**

Rakesh Gupta, PMP, Director of Enterprise Services
White Paper
An Automated Solution for Document Assembly

Project Performance Corporation
1760 Old Meadow Rd., McLean, Virginia 22102
p: 703.748.7118
e: rgupta@ppc.com