

TechnoFeature™

Practice management and technology articles written by experts.

The Other Document Assembly Systems—Part 2

By Seth G. Rowland, Esq.

INTRODUCTION

Below I explore [DealBuilder](#), [QShift](#), [Exari](#), and [Perfectus IPManager](#) in detail, focusing on the strengths of each system. I also offer a peek at [D3](#).

Each of these solutions are incredibly sophisticated tools in their own right. Each tool would result in significant people multipliers from its use. Each tool empowers the author of templates to reach numerous end users and give them the tools to rapidly create sophisticated legal documents.

DEALBUILDER'S POWERFUL RELEVANCE ENGINE

One fundamentally different aspect of DealBuilder is the “relevance engine” and associated analytic tools. DealBuilder empowers the author to build sophisticated rule-based systems with simple brackets (“[.....]”) and braces (“{TestatorName}”). The DealBuilder Author tools analyze the markup, check for proper syntax and logic, and build a complete dependency tree that determines the end-user interview.

This relevance engine enables rapid development and deployment of complex legal documents. The relevance determination is not limited to single templates, but extends to the entire dependency tree of inserted and assembled documents that form part of a master template. The relevance engine checks the dependencies and re-assesses the entire tree of pages, groups, and variables each time a variable changes, making the interview entirely dynamic. Moreover, as the templates change, and the relevance of variables changes, the interview automatically reflects the current state of the documents.

If an extraneous variable appears in the interview,

it will be because the variable is used in one of the documents being assembled. There exist myriad analytic tools available to the author/developer to debug interviews, including color-coded templates, multiple alternate markups, annotated interviews in developer mode that display the entire relevance tree for each variable, and the ability to produce documents in redline mode to show included or excluded text.

The “authoring” process includes a powerful logic check for syntax, dependencies, text structure, and more. Tim Allen, President of Business Integrity, has an open offer of a bottle of Dom Perignon to any lawyer who can produce a marked up document that does not have at least one logic error. Sad to say, when I built my own DealBuilder templates for [E-Lawyer.com](#) I was unable to collect on that offer of Dom Perignon. Not to say that the syntax of DealBuilder is difficult; rather the authoring analytical engine works so well.

The most recent version of DealBuilder — version 2.7 — included some significant upgrades that built on the core Relevance Engine that has been central to DealBuilder’s success. The Web-based interview now includes a tree navigation structure that you can toggle on and off. This tree is dynamic, like everything else about DealBuilder, showing only relevant pages and variables and refreshing automatically on each page display.

The earlier version included support for what DealBuilder calls ActiveText, the inclusion of variable data entered earlier in the interview in Prompts, Guidance, and Preamble text for variables, as well as Lookups. With DealBuilder 2.7, you can now include Computations in ActiveText which opens up just about any form of reporting into the interview process.

(Continued on next page)

DealBuilder 2.7 also introduced two features of particular interest in Corporate Counsel implementations: (1) transaction server reporting tools and (2) Lexicon support. DealBuilder has always had a transaction server where answer data was stored in a series of related SQL tables. With 2.7, Business Integrity introduced Web-based reporting tools, an interface to enable you to datamine the answer data and produce reports that you can export to Excel. Similarly, DealBuilder has offered use of Lookup Tables to create dynamic prompts that could vary based on state or country. But with 2.7, there is now multi-language support. This includes localization for multiple languages in the core prompts and syntax of the system. It also includes a feature called Lexicon, which enables you to have multiple-language prompts, with the language of a particular prompt varying on the language setting of a particular variable.

Finally, while DealBuilder does not bundle a workflow engine, it does contain flags and hooks that can be used by such an engine, including a parameter of the document that indicates whether an interview has been completed fully and is thereby LEGALLY SAFE to enter the workflow process. This determination of completeness (and the criteria thereby) is more than simply a determination of whether any unanswered variables exist. It is rather a determination of whether there are any unanswered relevant questions that the author has determined are required to be answered.

QSHIFT — KNOWLEDGE MANAGEMENT REPOSITORY

As a practicing lawyer, I would have loved QShift. I would build my documents in an outliner, so I could take advantage of an outline tree to navigate the different provisions and decide which ones to use in the final documents. QShift is the ultimate outliner. It is a desktop client to a Web-based document model and clause repository, hosted by Ixio.

When you use QShift for a document, you take the document structure and parse it into paragraph-sized elements. The analytic engine automatically identifies the outline structure of the agreement,

including the titles of each paragraph and sub-paragraph. These are parsed in to paragraph-sized (or multi-paragraph-sized) clauses which are then displayed in a document model.

You can then assign each clause a variable title and several properties, including an outline level, and status as optional or required, as well as group them with several other similar clauses in either a select-one choice or a multi-select. QShift enables you to see the document model live as you make your selections as to clauses to include in the particular model you are building.

QShift shines in the metadata included in the clause-bank and optionally included in the outputted document. Each clause, and each document model can have unlimited annotations. Authors can add comments on the appropriateness of use. You can workflow individual clauses until the language is formally approved and released. The End User can also add comments. In Author mode you can easily add comments to the system and notify the original author of the clause. In the outputted document, one of the formatting options is a Model Summary which shows the Title and Text of each clause, its status, and any comments on the clause.

Clause usage is not limited to a single template. A clause paragraph from one model can be borrowed by or shared with another model. Moreover the system is smart enough to recognize a shared paragraph and flag the author that changes to a particular paragraph will affect other models.

QShift proves particularly adroit at handling Word/RTF paragraph styles. Much of this process is automated, including the selection of independent Word stylesheets so that the output of a single template model can be formatted differently depending on the preference of a particular attorney user.

QShift is not strong on variable logic and does not currently support repeats and collections. While its QShift Document Variables wizard is much improved over an earlier viewing (including the ability to import and export XML answer sets and field management), the real strength of QShift lies in the clause

(Continued on next page)

management, melding knowledge management repositories, and document assembly.

EXARI — THE POWER OF XML AND PLATFORM INDEPENDENCE

Exari claims it is the only system that can be hosted on an entirely non-Microsoft based system. Exari can be hosted on a LAMP server (Linux/Apache/MySQL/PHP) and run assemblies on an Apple Macintosh. This is true because Exari adheres to standards for XML encoding of templates, variables, interviews, and logic.

Exari (which is SpeedLegal rebranded) is proud of its XML heritage. The authoring environment is an XML Editor. Word/RTF documents are uploaded into a parser which intelligently breaks the Word document into XML paragraph objects. From there, you can group or split the paragraph objects (as required) into objects that can be separately addressed. Each object in the system can have properties and conditions associated with it.

The sooner you can make the conceptual leap from “text markup” to object properties, the easier you will find Exari to work with. For the author, the ability to address an “object” instead of careful placement of brackets and expressions in a WYSIWYG editor like Word provides a much greater degree of control over the formatting of the resulting document.

Along with the XML comes a powerful, Web-based preview engine. As part of the process, you can see the document in HTML, but with color coding indicating the variables and the optional text. You can hover over these items and see the question that triggered that particular text, and even resume the interview at the question that triggered a particular passage.

Exari offers a unique feature, not present in the other systems ... the notion of bias. While you can manually code bias into templates in other systems, bias in Exari is a property of a field. The bias options include a particular configuration of defaults that have a particular slant. In addition, there is an Interview Style option that offer Express (only mandatory

questions) or Comprehensive (all questions).

Exari ships with a built-in output option to create either a Word/RTF document or PDF document. You can manage documents entirely on the Web-server or through an integrated document management system. The built-in system uses the user’s login to determine what the particular user can see.

Exari also handles lists of object particularly well. It includes a List Property that includes punctuation syntax for list items (a feature also found in Deal-Builder), enabling you to specify standard punctuation for list items in a particular list, the penultimate list item, and the last list item. But Exari takes this one step further with two unique features. The first tells the system to “dissolve” a single list item, which is great when the sublist contains only one numbered item (e.g. “(A) The testator gives the following gifts: (1) a gift of ____”). When combined with the second function to set the lead-in text, this becomes “(A) The testator gives the following gift: a gift of ____”. I cannot tell you how much advanced scripting I have done in HotDocs to accomplish this same result.

PERFECTUS — .NET/XML OBJECT-ORIENTED PROGRAMMING AND INTEGRATION

Perfectus is a technological wonder. Where Deal-Builder excels with its relevance engine that figures out all the scripting of the interview automatically, Perfectus gives a power-programmer the tools to build an EasyInterview that is both comprehensive and flexible. Taking the best of .NET, XML, and object-oriented programming, every aspect of the system is an object with multiple properties that you can set and script. You can reuse and reference objects, whether they be variables, clauses, expressions, outcomes, pages, pictures, or bound strings.

The Perfectus IPManager brings all the objects together into a powerful development tool for designing the pages, variables, and variable layout objects that makeup the Easy Interview. The system also includes a visual representation of the data flow and constraints between each of the page objects of the Easy Interview.

(Continued on next page)

Object orientation also extends to template objects and clauses. These objects and clauses can exist inside the IPManager Package or you can pull from an external source, such as templates hosted in a DMS like Hummingbird. The Query editor applies equally to variables as well as databinding to external sources.

Like DealBuilder and Exari, Perfectus includes configurable CSS (Cascading Style Sheets) that enable the developer to brand and style the interview to fit into a larger Web site. Perfectus includes layout options of such flexibility that they are typically only found in a desktop-document assembly software environment like GhostFill. You can control and configure each object on the page, with the final page representing the sum result of the properties of the constituent objects.

Perfectus takes the next step to include a built-in process manager options in its Package Explorer. The process manager includes control over templates that are part of a package, including rules over those templates, choice of conversion to RTF, PDF, HTML and XML, and choice of distributors including posting to a DMS system or e-mail. The Hummingbird integration features that ability to pass variables into profiles as well incorporate tags to associate a collection of outputted documents with a single EasyInterview and each other, including the ability to resume an interrupted interview or to restart as a fresh new interview.

The Easy Interview is particularly easy on the eye. It includes variable colored typeface as well as type size. The page editor enables you to drop HTML text, including text with variables and conditional outcomes right on the page. Prompts are displayed in bold black, help text in small grey font, and example text is in small light blue font. The Easy Interview page editor even includes support for a Picture Object in the interview.

D3 – MACRO SUITE AND CLAUSE MANAGEMENT

MicroSystems, makers of D3 declined to respond to my request for a demo and access to their system

for this article. From a brief demo I was able to have at New York LegalTech, I found D3 lacking in many of the tools of a full-power document assembly system. D3 focused more on incremental improvements in efficiencies, rather than on the People Multipliers evident in DealBuilder, Perfectus, and Exari. While D3 may be easy to use, its lack of power tools does not make it viable as a true document assembly engine that can be used with “Untrusted End Users.” It works best as a “better drafting tool” for lawyers and paralegals, offering major improvements over standard Word-processing and macro-suites.

Like QShift, D3 focuses on paragraphs and clauses. Its use of XML encoding is interesting, including the ability to bind data from various sources to the XML tagged fields. However, it entirely lacks any ability to generate dialogs. Questions are presented seriatim, one at a time, solely in the context of the document. As a knowledge management tool, it also lacks many of the advanced features found in QShift for collaboration on clauses, tracking of commentary, and versioning.

CONCLUSION

I expect that I may have lost some of you in the preceding discussion. It is hard to distill the essence of each of these systems into a few hundred words and still do justice to them. These systems have hundreds of unique and interesting features, and I cannot say you would be wrong in choosing any of these systems.

The people multipliers are there if the system is properly implemented. None of these systems are “purchase and forget.” They all involve some degree of work. The more you put in, the more you will get out.

Roy Lasris, developer of [Pathagoras](#) believes that template creation should be a painless, almost thoughtless process. QShift, DealBuilder, Exari, and Perfectus include rapid authoring tools that can be used by attorneys and paralegals with minimum training to simplify and speed the path to automated templates. But they also give you the tools to take

(Continued on next page)

the next step to build systems that produce final, PDF-ready documents, rather than “better first drafts.” The People Multiplier comes only when the quality of the outputted document is truly ready to use.

In assessing the work required to build a Perfectus IPManager package, a QShift document model, an Exari XML Template, or a DealBuilder template, please bear this in mind: How much would you spend to bring on a new attorney for a year? And how much time (valuable attorney time) would you extend to train that attorney in your processes and methods until he or she becomes a useful member of your firm? Now factor this time into the equation in choosing whether and which online document assembly engine to purchase as your development platform.

Copyright 2006 Seth G. Rowland, Esq. All rights reserved.

ABOUT THE AUTHOR

Recovered attorney Seth Rowland was named TechnoLawyer Consultant of the Year in 2002 for his contributions to TechnoLawyer on the subject of document assembly and law practice automation, and more importantly, his service to law firms around the country. He is a nationally known technologist whose company, [Basha Systems](#), has helped many law firms and content providers build document assembly applications for both internal use and for resale. Please feel free to visit his [blog](#) for the latest on document assembly or the [video tours page](#) to see what such a system can look like. Basha Systems currently offers document assembly consulting services in DealBuilder, Perfectus, Ghost-Fill, and HotDocs.

Contact Seth:

E: sg@bashasys.com

T: 914-827-9173

About TechnoFeature

Published on Tuesdays, *TechnoFeature* is a weekly newsletter containing in-depth articles written by leading legal technology and practice management experts, many of whom have become “household names” in the legal profession. Most of these articles are TechnoLawyer exclusives, but we also scour regional legal publications for superb articles that you probably missed the first time around.