Web-based document assembly in the civil legal assistance community

An overview of concepts, issues, tools, and methods

November 20, 2001

Prepared by
Capstone Practice Systems

with the support of
The Legal Services Corporation
Legal Services of Indiana
and
ProBono.net

An electronic version of this paper will be available at
http://www.lstech.org/WorkGroups/DA/webdocwhitepaper.pdf

Contents

Executive Summary

This report summarizes how document assembly technology, broadly understood, can be used in the delivery of legal services by nonprofit and pro bono providers, with particular attention to Web-based modes. It explains basic concepts, describes typical processes, and lists and analyzes some alternative vendors, products, and approaches. It provides some guidance on how programs should plan, fund, and manage projects. It includes a list of articles, organizations, and links to further information.

We have tried to be (1) comprehensive yet not overwhelming, (2) intelligible to the non-technical reader yet meaty enough for technically advanced readers, and (3) fair to the various vendors and alternative approaches without pulling punches or failing to give meaningful guidance.

This document should have broad exposure across the country, and may help catalyze a long overdue take-up of productivity- and access-enhancing technology in the nonprofit legal services world. Readers will include program managers and technologists, funders, vendors, consultants, law teachers, and staff attorneys. It should be a reference of first resort for anyone interested in doing document assembly work in legal services.

We have cast a very wide net in identifying major ideas and issues relevant to document assembly, but have not been able to do them all justice here. In some cases we have just scratched the surface. We apologize for the inevitable errors and omissions. It’s also important to note that the tools, techniques, and players in this field are constantly changing, and that aspects of this report will become outdated soon after it is released.

Introduction

This paper is part of a national process of analysis and planning regarding Web-based document assembly technology in the nonprofit legal services world. It is designed to help advance knowledge both in the field and among vendors who may wish their products or services to be adopted in that community. The authors have interviewed people who have been involved in legal services document automation projects and gathered a broad range of issues and viewpoints.

The purpose of this document is to provide an up-to-date compendium of information and ideas for legal services advocates, managers, and technologists. It reports on programs that are doing and have done projects of this kind. It is intended to be a practical guide, not a theoretical exposition. It is designed to be useful in many different contexts and under many different agendas.

By "nonprofit legal services" we mean the whole range of organizations that provide information and assistance on civil matters without charge. That includes governmentally and privately funded legal aid and legal services organizations, bar association volunteer lawyer projects, law firm *pro bono* operations, courts, law school clinical programs, and the like. There are approximately 8,000 lawyers and paralegals in programs directly funded by the Legal Services Corporation alone. Many of the considerations regarding document assembly for the nonprofit world are of course identical to those in the private sector, but differing missions and resources affect how those considerations play out.

Technologies for the automated production of legal documents have been in use for well over twenty years, and increasingly sophisticated applications can be found on law office desktops across the profession. Document assembly tools offer great improvements in both productivity and quality in the delivery of legal services. But for various cultural, political, and economic reasons, actual use remains limited to discrete islands of enthusiasts. The nonprofit legal services world in particular has made little use of this technology.

In the last several years these applications have begun to be deployed over the Web, promising dramatically greater scope and easier distribution. (The advantages and challenges of Web-based approaches are the focus of this report.) Again, take-up has mostly been by experimenters and early adopters. But the opportunities have been recognized by many programs, and funds are now becoming available to pursue them on a large scale. This is an important time for coordinating efforts and leveraging common resources.

"Document assembly" is used here in a broad sense, covering

- both the automated production of documents and the "intelligent interviewing" that typically precedes it;
- both word processing documents and official or "graphical" forms;

- applications used by lawyers and other advocates, as well as those used by *pro se* litigants and other individuals doing legal work on their own behalf;
- both fully online configurations and mixed topologies, e.g. involving desktop applications and local area networks.

Our focus, then, is on the general *process* by which networked computer technology can be used to assist people in the preparation of law-related documents. This can be accomplished in a wide variety of ways, involving a bewildering array of technical and managerial choices.

WHAT is document assembly?

Basic concepts

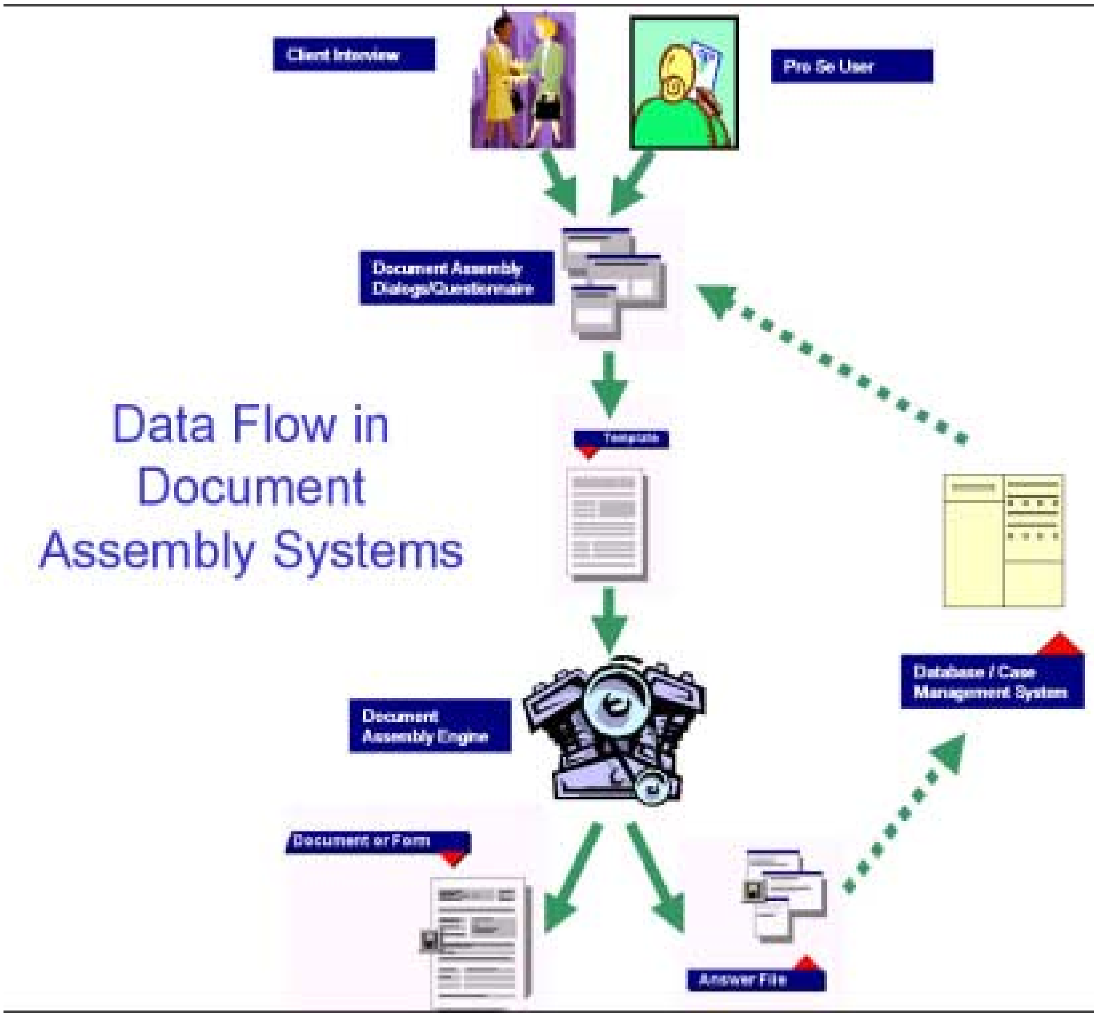
Document assembly. Computer-aided drafting. Document modeling. Document automation. There are many names for software tools that help people quickly generate certain types of well-structured documents. Contracts and wills are good examples. A lawyer, paralegal, secretary, or do-it-yourselfer responds to a series of dialogs and prompts, often from within a familiar word processing program, and the system assembles a draft document. Or the user picks forms, clauses, and other document components as needed from libraries of alternatives.

Sometimes, an organization develops a custom system with one of the document assembly "engines" mentioned below, using its own forms and experience. This can require a fair amount of up-front time and tedious work (thinking through and programming many possible alternatives) but can result in excellent leveraging of practical legal knowledge.

Other times, the document assembly system is one that is obtained from a legal publisher or document assembly vendor, designed to produce specific types of documents valid in certain jurisdictions. Some well-known off-the-shelf systems include Immigrant Pro from Immigrant Software (http://www.immigrantsoftware.com), WillMaker and LeaseWriter from Nolo Press (http://www.nolo.com), and JC Forms from Capsoft (http://www.capsoft.com), which generates approved California court forms. TurboTax from Intuit (http://www.quicken.com/taxes/) is probably the most popular program of all time with form automation features. Sometimes legal services organizations develop their own systems and make them available to fellow organizations. One example of such sharing is the Greater Boston Legal Services family law system, which is being used by a variety of organizations in the Boston area.

Either way, the basic goal is to capture some of the regularities underlying the documents—what sections, paragraphs, sentences, and words go where under what circumstances. The document assembly engine provides a kind of power steering for lawyers and others to make choices and specify details like names, numbers, dates, and phrases. Instead of cutting and pasting, you can pick desired options or alternatives from lists; instead of searching and replacing phrases like "Plaintiff name" with your client's name, you can respond to questions and let the computer do the clerical work.

While terminology varies among programs, there is usually a "template" that represents a model of particular kind of document, with "variables" placed in locations that change from case to case. When the template is run, the user answers questions corresponding to the variables (posed in a series of interview-style dialogs), the answers are stored in some kind of "answer file," and the desired document is generated. Typically a given answer file stores all the data relevant to a single client or client-matter, and that answer file can be used to generate more than one document or form (e.g. a complaint for divorce, a financial statement, and various motions in a family law system). Answers can be changed later (e.g. a correction to the name of the client) and the document(s) can be re-generated. The generated document is usually in some textual format (e.g. Word, WordPerfect, RTF) and can be freely edited after assembly. The following figure is a simplified view of the typical components and processes involved, including an optional database connection.


Client Interview
Pro Se User
Document Assembly Dialogs/Questionnaire
Data Flow in Document Assembly Systems
Template
Database / Case Management System
Document Assembly Engine
Document or Form
Answer File

In addition to basic point-and-shoot clause selection and fill-in-the-blanks variable replacement, these systems can store drafting rules and other kinds of practitioner knowledge that can be used to guide the hand of novices and experts alike. For example, a divorce system can be designed to ask the user about the client's state of residence, marital status, financial situation, and number of children and, based on the answers and follow-up questions, insert appropriate clauses into the complaint for divorce and associated motions. Document assembly technology has been applied to everything from simple thank-you letters to elaborate expert systems that advise on the laws of many jurisdictions and generate document sets that can reach into hundreds of pages.

While still negligible as a percentage of overall law office automation professionals, in absolute numbers there are quite a few people gainfully employed in legal document automation. Over a dozen vendors of software tools are active in this business, and a similar number of legal publishers offer packages with document assembly features. There are probably at least fifty independent consultants specializing in this area in North America, and many more people similarly engaged within law firms, law departments, government agencies, and publishers. Thousands of systems have been developed and used across the professional landscape.

As a result, a lot of practical experience in the challenges of modeling legal documents—and the comparative advantages of different approaches—is accumulating.

Web-enabled document assembly

The World Wide Web opens up several new opportunities for organizing and delivering document assembly applications. Any or all of the major components – engine, templates, answers, documents, help material – can be served from or stored on a Web server, providing location independence, multi-user access, ease of use, and other benefits characteristic of the Web. For advocates, a big advantage of Web-based implementations is the centralization and instant updating of template collections. For *pro se* users, they allow access to robust document automation without requiring special purpose local software to be purchased, installed, configured, and maintained. Often just a browser is required, together with an Internet connection and a printer – tools that are available in most public libraries. For IT professionals (and budget conscious managers) a single centralized server and staff can economically provide document assembly capabilities to hundreds or thousands of users.

Web-based document assembly environments can not only replicate much of the functionality we've seen on the desktop, but add interesting new features, such as hyperlinks in the assembled documents that take you back to associated questions, where you can enter or change answers and reassemble the document. And, helpful links to ancillary Websites can easily be placed in the browser-based dialogs.

There are at least two different, albeit complementary, forms of "Web-based" document assembly. In the first case, sometimes referred to as a "fat client" scenario, the templates are accessed and downloaded from a Web site, but run in a regular local computer application, like Rapidocs Classic or HotDocs. In the second case, a "thin client" computer is all that is needed, and the assembly process either happens entirely on the Web server, or via a small application that serves as a "plug-in" to the local browser, such as an ActiveX control, used by Rapidocs, or a Java applet, used with Grantha. Several vendors support both methods.

Possible disadvantages of Web-based approaches include disrupted access due to server downtime, some loss of functionality (due to the limits of the browser interface), the necessity of an internet connection, and some greater difficulty in linking to databases, case management systems, and other third party applications that may run locally in legal services offices.

Important distinctions

Some recurring distinctions are important to make up front:

1. *Word processing documents vs. graphical forms.* Document assembly generally encompasses both freely editable word processing documents and fixed-format, "graphical" forms, where the background is static and information typically can only be placed in pre-designated fields. The terminology for these two kinds of documents varies, and can be a source of confusion. But most contexts we care about require both kinds of documents.

2. *Questioning and advice vs. document generation.* One of the characteristics of most document assembly applications is that users provide information and make drafting decisions through questionnaire-like screen dialogs that are *outside* of a target document. The document being generated may be visible during the interview, and the user may be able to access it to revise answers or edit passages, but usually there is a discrete interface in which questions can be asked and advice given. Many document assembly tools can in fact be used to produce information gathering modules, advisory systems, and intelligent checklists that needn't result in any traditional document at all.

3. *Advocates vs. self-help users.* Document assembly technology can be and is being used both by advocates providing services for clients and by individuals doing work for themselves. Software selection criteria and project planning can differ dramatically for the two target communities. And even among advocates, there can be important differences between the needs of staff advocates and pro bono or other volunteers. This report tries to cover most issues in a common fashion, pointing out differences when appropriate. There are also hybrid *pro se*/advocate scenarios, in which the client answers an online questionnaire on his or her own, either in the law office or elsewhere, and the answer file goes to the attorney for further review, revisions, and actual document drafting. The dot-com AmeriCounsel made heavy use of this method, a form of "unbundled" service.

4. *Users vs. developers.* Document assembly software typically involves distinct tools and interfaces for "end users" and developers. There are many features and issues that can be critical for people charged with developing applications that are irrelevant to the ultimate users, and vice versa. Some software choices offer excellent end user interfaces but clumsy development tools, and vice versa. (Beyond users and developers, system managers often present a distinct, third, perspective of their own. See below for some notes from this perspective.)

5. *Accessing templates on the Web for local processing versus assembling them as part of an online session.* See previous section.

These distinctions often are combined. For instance, a given online document assembly initiative might involve interactive questioning and advice that is entirely browser-based, but document generation that happens on the desktop. Or word processing documents that are assembled on the server, but graphical forms that are built locally. Or one approach that is followed for self-help users and another for advocates. In characterizing any such implementation, you really need to ask *what* is happening *where*, *when*, and *how* for *whom*?

Related subjects

Electronic filing

Electronic filing of documents in connection with courts and government agencies is a large and growing field of its own, with some similarities to document assembly. Obviously the output of a document assembly session could be the subject of an electronic filing session, and some electronic filing solutions involve interactive question-gathering and submission of resulting data sets without any intervening traditional document – paper *or* electronic – at all (e.g., through XML.) After all, it's the *information* that is important. We recommend that the legal services community keep careful track of developments in this area. We would not want to build a document assembly system that is great at producing documents but that was incompatible with or obsoleted by an e-filing system in a particular state or practice area.

Artificial intelligence

People sometimes ask how document assembly programs differ from expert systems or other forms of artificial intelligence. The quick answer is that the categories overlap. Most contemporary document assembly functions are accomplished with conventional programming techniques and standard database structures. Most artificial intelligence applications, on the other hand, rely on a more sophisticated "inference engine" and an explicit "knowledge base" with the goal of giving advice and solving logical problems. But generating documents based upon applications of rules to facts is one of the things both document assembly and expert system can be made to do. In fact, all non-trivial document automation applications can fairly be said to encode significant knowledge, and even artificial intelligence researchers seem comfortable with calling some of them expert systems, without quibbling too much about the sophistication of programming techniques used. The MicroMax government-benefits-eligibility system provided by Community Catalyst in Boston, for instance, is an expert system built in a document assembly tool (CAPS).

Document assembly functions also are often joined with "intelligent checklists," case management, and other tools in more comprehensive "practice systems."

Knowledge management

Knowledge management (KM) has received a lot of attention in the legal world in the last several years, and many organizations have launched initiatives and created positions dedicated to this activity. A related phenomenon is the rise of Web-based information "portals" that provide a single point of access to a related body of information, often with some degree of personalization. These include LawPort (http://www.svtechnology.com), Hummingbird's Enterprise Information Portal (http://www.hummingbird.com), the Hyperwave Information Portal (http://www.hyperwave.com), and Microsoft's SharePoint

Portal Server (http://www.microsoft.com). The legal publishers have also tried to get into the act with the LexisOne and WestWorks sites. ProBono.net (http://www.probono.net) is a leading player in the legal services world.

Document assembly applications are a recognized form of KM in law practice organizations, often drawing upon and complementing work product repositories, collaboration tools, data mining, and other techniques. In fact, they can be a critical means of preserving the know-how of legal services professionals who may later leave the organization.

Online legal services

Online legal services, involving various forms of “e-lawyering,” encompass the whole range of activities whereby legal information and advice are made available through communication networks like the World Wide Web. Document assembly technology, and the interactive questioning that accompanies it, is just one aspect of an online legal services environment. Other aspects include hyperlinked reference material, interactive checklists, lawyer/client matching, matter management, and online dispute resolution.

Online services – whether self-help or advocate assisted – raise many fascinating policy and ethical issues that will not be addressed here, such as privacy, confidentiality, unauthorized practice of law, improper advertising, and malpractice and other forms of liability. Interactive forms that provide explanations and include logic that guides users through basic forms of legal decision making are vulnerable to challenge on some of these grounds.

Interactive multimedia

Particularly in the educational technology world, interactive video and other multi-media applications have received a lot of attention. For *pro se* oriented applications in particular, these technologies can usefully be incorporated into document assembly environments. TurboTax and the I-CAN project described below are powerful examples.

WHY deploy document assembly?

While this report is focused on the tools and techniques of document assembly, it is important to spend a little time reviewing the substantial benefits (and associated costs) that need to be taken into account in pursuing one of these projects. You need to know what your goals and resources are to properly choose appropriate software and methods. There are many, sometimes subtle, components both in the return on investments (ROI) and the total cost of ownership (TCO) of a document automation project.

Benefits for advocates

Document assembly systems can provide the following benefits, from the advocate's perspective:

- **Quality assurance.**
  - **Correctness**. Assure that client information is correct on all forms; assure that forms are filled out in the correct manner (e.g. that income is listed in weekly, monthly or annual amounts as appropriate on a court form).
  - **Completeness**. Assure, for example, that all appropriate requests for relief are made (or at least considered by the advocate).
  - **Consistency**. Assure that the latest language, for example, on interrogatory questions, is used by all advocates.
  - **Standardization**. Encourage legal services organizations to reconcile language differences among models used by various advocates and standardize on best practices.
- **Productivity/efficiency**. Dramatically reduce the time required to draft complex documents such as financial statements in divorce matters; enable paralegals and students to create first drafts of documents previously done by attorneys; enable secretaries to create first drafts that were previously done by paralegals; enable legal services programs to serve more clients with the same or fewer resources.
- **Responsiveness**. Dramatically reduce total elapsed time between client interviews and court filings, or the settlement of a case and its documentation.
- **Process improvement**. Allow instant sharing (across time, staff, and offices) of client data captured and stored in answer files; facilitate access and re-use of client information already stored in a case management system. Better understand and re-engineer these kinds of processes through the very work of automating them.
- **Training and continuing education**. Guide less experienced advocates through the correct questions and options via dialog screens; provide optional help screens, with explanatory text, to teach legal and advocacy skills.
- **Consolidation of expertise**. Capture the substantive knowledge of more experienced and specialized attorneys (e.g. the right questions, options, language, strategies) so that:
  - Knowledge is shared within an office and across legal services offices.
  - Knowledge is preserved in case of staff turnover.
- **Job satisfaction and enrichment**. Liberate advocates to focus on more challenging and satisfying tasks (e.g. the stuff attorneys went to law school for).

- **Access to legal services**. Help discharge professional responsibility to improve access to legal services through better technologies both for self-help and assisted scenarios..

Benefits for *pro se* litigants and other self-helpers

From the perspective of someone pursuing legal work on their own behalf, a document assembly application can provide:

- Access to information and assistance when other sources are unavailable or ineffective.
- Meaningful guidance at a meaningful point in the process.
- Ability to work at their own schedule and pace.
- Generation of forms that comply with the format and content requirements of a court or agency.

Effectively implemented, Web-based document assembly technology has the potential to produce *transformative* increases in access across the spectrum of non-advocate users. The return on investment of self-help-oriented initiatives can be quite spectacular, even compared to enhanced tools for advocates.

Costs

The costs involved in implementing a document assembly system are reviewed in greater detail below, but include the following factors:

- Legal professional time (development)
- Technical staff time
- Management time
- User time (training, learning)
- Consultants
- Software licenses
- Hardware
- Services (e.g. telecommunications)

Obstacles and opportunities

Document assembly technology is mature and ready. It offers dramatic benefits in efficiency, quality, and job satisfaction. It can usually work in your existing computing environments. It can be a powerful supplement to case management, electronic filing, and *pro se* initiatives.

So why doesn’t the civil legal assistance community make much use of document assembly? How do we get from pilots and demonstrations to mainstream and routine implementations? This is not only a matter of good technology and effective project management, but of courage, leadership, and cultural sensitivity. Our problem has not been lack of ideas or tools. There are deep cultural and managerial challenges that need to be overcome. Here are just a few of the factors often identified:

- Unawareness of available tools and how to tap their benefits
- Lack of in-house resources and expertise for automating templates
- Personal and organizational inertia
- Innovation overload
- Conceptual difficulty
- "It's nobody's job"
- Professional arrogance. Many lawyers feel that their specialized knowledge is not amenable to any useful forms of automation.
- "Not invented here." It is often important to standardize on document models before automating. Standardizing can be good (it forces advocates to think about why they have different models and chose the best one or two) but it can be a huge obstacle too (attorneys need precious time to focus on this and of course, everyone thinks his/her language is better).
- Fear of lost creativity. People worry that form systems can produce routinized, unimaginative, compromised practice. While this is a legitimate concern, it should also be pointed out that by automating the routine and mechanical aspects of practice, people will have more time and energy to attend to the humanistic and creative dimensions.
- Too busy bailing … (to repair the hole in the boat)

On the other hand, our challenge is not just to discover and apply established technologies and well-understood methods – there are fascinating questions about how to use this technology to best effect in support of the equal justice agenda.

HOW do you run one of these projects?

Some opening issues

A legal services organization considering a Web-based document assembly project needs to consider issues like the following:

1. Document assembly technology for *whom*? - LSC grantee staff attorneys? or also paralegals, secretaries, students? staff in non-LSC funded legal aid offices? bar association initiatives? pro bono? law school clinics?

2. Document assembly for advocates only, *pro se* litigants and the general public, or both?

3. Pure Web-based, or also desktop applications?

4. Basic document generation, or also associated functions like document management, workflow, knowledge management, and case management integration?

5. How far would you like to go in acquiring or building substantive templates, as opposed to the just the “plumbing” (e.g. a Web server and some enabling software) and some basic forms?

6. What general approach will you take to staffing the development, maintenance, and support functions?

7. To what extent will the applications be distributed among desktops and local servers, and to what extent centralized in pooled resources like a shared server farm?

See Appendix I for a useful set of questions to answer in defining your project.

What’s involved? The major ingredients

To provide the benefits of automated document drafting to a community of users, you need the following basic ingredients:

1. A delivery environment (hardware, networks, and general software)
2. An underlying “engine” (e.g., Adobe Acrobat, GhostFill, Grantha, HotDocs, Rapidocs, SmartWords, or custom equivalent)
3. Some “content” (intelligent templates)
4. Educated end users
5. Arrangements for building and maintaining document automation aspects of the delivery environment
6. Arrangements for maintenance of content
7. Arrangements for support of users

To the extent that you need or want some of #3 (content) to come from within your own community, you need

8. Educated template developers
9. Arrangements for support of developers

And to achieve #4 and #8, you need

10. Training of users and developers

What'll it cost? Budgeting

Above are the main cost factors for any scenario. To flesh them out:

1 – The general **delivery environment** will generally pose major new costs only in situations where existing desktops and servers are not reasonably up-to-date for general office suite and Web site applications. Most Web-based document assembly delivery solutions are not particularly demanding in computing resources, and can usually live alongside other applications. Building a fresh Web site involves major costs and considerations outside the scope of this paper. One approach for organizations without an existing Web facility is to outsource application hosting and maintenance to a vendor or third party, in which case the only "delivery environment" you need to directly worry about would be a Web browser and local word processing software. Outsourced HotDocs template serving is available, e.g., from DocsEngine in New York for a few hundred dollars a month for a megabyte or so of content. Rapidocs is offering a complete hosting solution which includes over 150 completed state specific interactive documents for a set-up fee which depends on the complexity of the Web site, and a monthly hosting fee of about $500.00 a month. Servers can also be shared across organizations within a state, and even across states. Whatever organization takes the server responsibility on is going to have to have good systems administration support - not just application support.

Document assembly functionality does not generally entail connectivity speeds or arrangements beyond what should be in place for an underlying portal or intranet.

Extra costs and concerns in building a delivery environment for *pro se* users may include kiosks or workstations that are optimized for audio-video content, provide touch screen technology, etc.

#2 – The **engine** involves software licensing (and annual maintenance) costs per desktop and/or per server. Pricing models for Web-based document assembly are still in flux. All of the main specialized programs list for under $200 per seat in reasonable quantities, and the server licenses list between $5,000 and $25,000 per server. With the aggressive discounts some vendors seem willing to offer (and others are presumably willing to match), programs can safely plan on "engine" costs per end user (whether desktop-based, Web-based, or a mixture) being well under $100. Implementations involving large numbers of public users should have very low marginal costs per user.

#3 – Content is pretty open-ended, because it encompasses both material already commercially available, and material custom-built for legal services. Lexis has a lot of forms already available in HotDocs format (immigration, bankruptcy, family law for some states, California Judicial Council forms, etc.), with prices averaging in the few hundred dollars range per package. They probably could be persuaded to donate or heavily discount some of this material. Rapidocs has a large collection of forms it has already vetted out with lawyers in most states, which it would bring to the table without much additional cost as part of an overall relationship. GhostFill would be willing to do likewise with its own growing collection of automated forms.

So, for material that has already been created for other markets, legal services offices can probably acquire it at low to negligible cost. But of course most such material is oriented toward practice areas not much seen in legal services, like incorporation, patents, estate planning, commercial lending, etc.

And some ground can be gained by identifying, standardizing, distributing, and customizing form sets that have already been developed in the various pockets of legal services document assembly pioneers. (see **Who** section, below.) But beware that content developed using one engine is generally not transferable to another platform without considerable re-programming.

Sharing of certain content development efforts at the national level (e.g. immigration, bankruptcy, social security) and certain content at state level (divorce, housing) offers a cost-effective approach, but requires high levels of coordination.

Even for content that is free or low cost, there will typically need to be customization for specific legal services contexts. For example, changes in prompts, options, defaults or language of generated documents can involve a significant amount of gap-filling and modification.

Building new, custom document assembly content for legal services practice areas will require significant staff time and programming costs. The economics here will depend largely on (1) how ambitious you want to be, (2) how well you can leverage common resources, and (3) the ease of use of the authoring system. Simple letters and pleadings can be built by a lawyer, paralegal, secretary, or law student in a few hours after a few hours of self-guided tutorial work with one of these programs. A full-featured but still basic set of templates, say, for divorce practice in one particular state, could require $10,000 to $20,000 in expert template developer service fees and a hundred hours of attorney staff time. Substantive experts are needed to identify documents, standardize language (both within and across programs, not an easy task), mark-up documents, develop help screens and prompts, answer programmer questions, and extensively test various iterations of the system, including the final product. (Once created, however, such a template set could be deployed and adjusted for use in a second office in the same state for around $1000.) These costs increase if documents change significantly during the development process.

An additional cost in this category is the time to format documents appropriately with auto numbering, styles, tables of contents, cross references, pagination, etc. Supporting several different versions of several different word processors (WordPerfect, Word) can be a daunting task, and require word processing expertise beyond that of some document assembly specialists.

Templates intended for *pro se* use typically involve less programming complexity, but require additional work in customized prompts, instructions, and multi-language support.

#4 (& 10) – End user training can be addressed by budgeting for some basic user documentation and an hour or two of small group training. $50 per user is a good ballpark for this. Properly designed, these programs tend to be easy to learn and use. Web-based training modules can make sense for some users. Complex applications written for advocates will likely require more training than simple applications aimed at *pro se* users.

#5 – Building and maintaining the document assembly aspects of the delivery environment involves some combination of in-house staff and consulting time. Figure around 40 hours per LAN or Web site to set up (at a average hourly cost of $60, that's $2400), and *considerably* more if you want to do document management or database integration. A similar number should probably be allocated each year, to cover maintenance, operations, and upgrades. This is of course additive to underlying Web site maintenance and management.

#6 – Content maintenance. Figure at least 20% per year of initial content development cost to purchase or provide support to keep templates substantively up to date and relatively bug free.

#7 – End user support costs can vary greatly depending on (1) presence and quality of internal help desks, (2) quantity, scope, and complexity of templates, etc. Effective deployment of document assembly, though, can often ride on good, responsive support. For technical questions, the major vendors all have national support hotlines, and seem willing to provide discounted access. (List prices can be expensive - Capsoft charges e.g., $2.99/minute, which generally only makes sense for technical questions by developers.) Most programs will need to provide user support relating to their own templates themselves or through shared hotlines. This can involve an incremental cost of $5,000 to $10,000 per year for an average size program.

Note that there are at least three kinds of end users in the scenarios we've been discussing: program staff members, pro bono or volunteer advocates, and *pro se* users. For public users, admirable commercial models like Land's End may worth studying. Web site visitors can enter their phone number in a form and receive a quick call from a helpful specialist.

#8 (& 10) – Developer training. There are lots of independent document assembly consultants who do training, and most vendors offer training events at selected regional locations and/or customer sites. Two days of basic training for document assembly template developers usually involves around $600/person. But figure on at least a few days per developer of self training before and during a template automation project. And there may be travel costs. It generally is not realistic to expect existing staff to learn and do substantial document assembly development work without an extended period of relief from other responsibilities.

#9 – Developer support can be critical and expensive if a strategy of substantial in-house development is followed. A novice template developer can be very thirsty for coaching or mentoring services. Experienced developers on the outside generally charge nonprofit rates of at least $60/hr. If you allocate each new legal services template developer 10 hours of consulting help the first year (potentially just a drop in the bucket), figure on another $600 each.

In general – Beware that technology projects often take much longer and cost more than initially estimated. Some project veterans recommend that you estimate as accurately as you can, and then double. Also be aware that some of these costs are not recoverable if the platform changes or the vendor goes out of business.

Cross-program organizational issues

Effective deployment of Web-based document assembly raises tough policy questions for the legal services community as a whole. How do we best balance the need for a healthy diversity of approaches with the desire to avoid wasteful reinvention and duplication of effort? Where will state and national support and coordination be most effective? How can successful projects best be replicated and sustained?

Two extreme scenarios might be painted as follows:

- “Rally around the flag” – large groups of programs settle on some common technology standard and divide up the work. Requires a high level of coordination, and some all-eggs-in-one-basket vulnerability, but produces a high degree of leverage and reuse.

- “Let a hundred flowers bloom” – many different platforms and strategies are pursued across different programs. Involves a lot of duplicated effort and reinvention, but avoids political paralysis and may yield several paths to success.

To the extent that there is substantial agreement on at least some aspects of the work to be done and the tools to be used, one can imagine various ways to distribute responsibilities. For instance,

- A shared regional or national resource might provide
    - Web server(s), security, answer file storage and management, template storage and management, related Website development and technical maintenance and installation of the above
    - Engine vendor relations (licensing, upgrades, bug fixes)
    - Tech support and training for template developers
    - User support for technical questions only (toll-free numbers)
    - Content acquisition from and negotiation with publishers
    - Content development of nationwide forms of specific use to legal services such as immigration, social security, federal court forms, etc., along with related help resources
    - Technical support for case management, document management, answer file synchronization and other integration efforts in local offices
    - A “template factory” for those local offices that don’t want to build their own templates (possibly billed back to the local offices)
    - Documentation (technical, training, content for national templates)
    - Identification/certification of available consultants for content development and specialized integration
    - Technical support for video, touch screen, security, and related *pro se* issues
    - High level consulting/guidance to state and local organizations on which practice areas would benefit the most
- A state-wide organization might provide
    - Content development of state-wide forms and related help resources
    - User support for state-wide content
    - Documentation for state-wide content
    - An additional “template factory”
    - User training for state-wide and national content and/or train the trainer
    - Coordination with state-wide e-filing and other related state court initiatives
- Individual legal services offices might provide
    - Content development of local forms
    - Contribution to state-wide materials (with different programs taking the lead in different practice areas)
    - Basic user training and support
    - Integration with local case and document management systems, with help from the national or state level

Developmental and substantive standards – Toward platform independence

Any long-running legal document automation initiative needs to anticipate periodic changes not only in specific user and developer tools, but in the entire technology context. Investing too shortsightedly in the specific syntax and features of one particular vendor’s authoring paradigm can require expensive and disruptive conversions when business or technology circumstances require a change. It is a good practice to build and maintain models of legal documents, and the implicit know-how behind them, in a platform-independent fashion (for example, using the Unified Modeling Language or eXtensible Markup Language) prior to coding them within the framework of a specific tool. This can add upfront costs, but pay large dividends down the road.

Even within a single vendor's offerings, conversion can be tedious and difficult (e.g. CAPS to HotDocs).

To the extent legal services groups undertake to design their interactive documents independently of a particular current platform destination, they will more easily be able to migrate when the need or opportunity presents itself. They may also find themselves needing or wanting to support more than one platform *simultaneously*: e.g., one that exceptionally addresses the needs of *pro se* users, and another that is more convivial to the needs of lawyers.

These ideals are more easily described than accomplished, and no current player is visibly pursuing them. One set of approaches to this challenge has been articulated in the Open Practice Tools (OPT) initiative. For a summary of some ideas behind that project, see http://www.capstonepractice.com/OntoOpen.html.

Adherence to other established standards such as XML, SQL, ODBC, ODMA will be increasingly important in many implementations.

HOW do you select and implement the needed technology?

Basic platform and architectural choices

There are two major categories of technical decisions to be made: (1) what software platform(s), existing or new, will you base your solution on?; and (2) what kind of "architecture" will you follow in allocating processes among various server and client machines?

On the first question, choices include:

1. going with an existing, specialized document assembly tool
2. building a custom solution using more generic software
3. combining the above two strategies

The three basic architectural options are:

1. Traditional desktop / local or wide area network ("fat client")
2. Serving applications from Web site, but running them with local software (ditto)
3. Pure server-based ("thin-client")

Specialized engines

Computer-aided document drafting can be accomplished through any number of software tools. Macro and merge features built into today's word processors are of course often used. Similar features are also available in some database programs, spreadsheets, groupware applications (like Lotus Notes), and general purpose programming tools like Visual Basic. But quite a few specialized programs have emerged for building legal document assembly applications—variously dubbed "engines," "platforms," "authoring environments," and the like. And there are distinct products and vendors for other vertical markets like accounting, banking, health-care, and insurance, with little apparent cross-pollination, despite great functional similarity.

It is remarkable how many law-oriented document assembly products there have been over the past fifteen years—especially given the relatively low level of serious usage. For each of the dozen or so products now available, there is at least one that has faded from view. Every few years someone does a comprehensive survey of software alternatives. Jim Eidelman wrote one in *The Lawyer's PC* around 1989; Joe Kashi did another in *Law Office Computing* in the early 90s. In 1997 Alan Soudakoff and Marc Lauritsen got copies of all the commercial engines, and wrote a detailed "shopper's guide" that organized them into a comparative framework of common features. Their full survey appeared in the October/November 1997 issue of *Law Office Computing.*

A follow-up piece, called Power Tools for Document Preparation, appeared in *AmLaw Tech*, Spring 1998, and included thumbnail sketches of a broader set of contemporary products. (A version of it is at http://www.capstonepractice.com/amlaw6.pdf). Many of the products reviewed, interestingly, are no longer actively marketed or supported.

Here are sketches of some of today's prime candidates that currently offer or will soon offer Web-based solutions. Different kinds of details are given for each, to illustrate the complex tangle of considerations involved in choosing products in this field. These and other solution providers are being asked to respond more definitively to the questionnaire in Appendix II, and their responses will made available to the legal services community. A fuller list of engines can be found later in this document.

HotDocs

Capsoft Development's HotDocs presently has an overwhelming edge in market share. Capsoft (http://www.capsoft.com) is part of the Lexis family (in turn part of Reed Elsevier), which appears to be making a substantial commitment to HotDocs after several years of lukewarm corporate attention. It has an information-rich Web site, a history of national user conferences, a very active Internet discussion service, and regional user groups. Dozens of independent consultants provide system development and training services for Capsoft's products. Major publishers have used CAPS and HotDocs as platforms for off-the-shelf document assembly systems, and many case management and other commercial software packages affirmatively support integration with HotDocs.

Capsoft has built communities of users, publishers, consultants, and software partners that dwarf any comparable phenomena in other document assembly contexts. It has survived the vicissitudes of the legal document assembly marketplace for over 12 years, and one of the founders still leads its software development activities.

HotDocs has been through five major versions as a Windows-based product, and is available both in "regular" and Pro versions. (HotDocs 6 is due out soon.) It runs with nearly every version of Word and WordPerfect, and includes a very robust graphical form generator. Web support came in two stages: (1) support for auto-assemble (HDA) files that can be posted on a Web site and then run seamlessly on any Internet-connected PC that has HotDocs installed, with support for uploading and downloading of answers; and (2) support for pure browser-based assembly (HotDocs Online), using server technology that automatically builds dynamic Web pages from conventional templates.

HotDocs Online can be previewed at http://hdo.capsoft.com. It is also used as part of LexisOne (http://www.lexisone.com), and available in an Application Service Provider (ASP) mode through http://www.docsengine.com.

Because of its market share, full feature set, and vendor stability, HotDocs will likely be a natural platform candidate for many programs. It is already used by a variety of legal services offices. Some negative considerations include:

1. A history of using proprietary approaches (answer files, component files, library files), which makes customization and integration with other applications more difficult.

2. HotDocs Online is Windows NT-based. If a program chooses to use a more secure operating system (AIX, BSD, UNIX, LINUX), aspects of HotDocs Online can still be used, but it requires much more customization.
3. HotDocs Online doesn't have built in support for graphical forms, although effective third-party solutions are available.

Rapidocs

Rapidocs originated in the United Kingdom, where it was developed by Epoch Software. Epoch has been particularly innovative in the Web-based document assembly and online legal services field. Rapidocs is sold in the United States by MyLawyer.com, Inc., run by Richard Granat, a legal services and legal technology veteran. It is used as the document engine in the MyLawyer.com site and in Web sites run by legal insurers ARAG (www.LawExpress.com) and by Royal Sun Alliance (http://www.hrlawinfo.com). Rapidocs is also being used as part of the Maryland Legal Assistance Network (MLAN). There they make published Rapidocs forms available for free to the legal services community, together with a package of publishing tools so that MLAN providers can create others.

Rapidocs US has already automated more than 300 forms, including a core document collection (wills, powers of attorney, living wills, marital separation agreements, etc.), for every state, which has been reviewed by local attorneys. Their forms come with detailed help instructions and are designed for public consumption in quantity (which is different from what lawyers need). They are willing to license this collection of already automated forms for use by legal service programs at very favorable terms. They have spent large sums to do these forms and believe they have a fast development system. They are next going to start automating court forms in fillable .pdf format.

Rapidocs Classic allows processing on the desktop as a Windows application, independent of an Internet connection.. **RapidocsX** delivers Rapidocs through the browser as an ActiveX object and integrates with other Rapidocs components and workflow tools. **RapidocsJ** is a pure Java-based version that is now under development.

Rapidocs is built around a single file architecture to simplify the process of exchanging documents. Each file is compressed and secured with a 128-bit key to protect its contents, although there are interfaces to allow access to meta-data to allow effective document management. Each Rapidocs file contains the questionnaire itself, extensive explanatory text and help, document automation logic, every permutation of formatted document text, answers already entered by a user or a server process, details of every change made by every user, and annotations. Rapidocs packs all of this into a compact file that can be transferred via the Internet in seconds.

Rapidocs follows a multi-tier architecture for reasons of performance, distribution, and scalability. The document and user manager applications run as server processes, and are supported by standard SQL databases.

Some possible downsides of Rapidocs include:

1. Rapidocs requires its own word processor to be used, although this has some advantages (a user doesn't have to have Word or WordPerfect installed) and export to common formats (Word, RTF, and HTML) is supported.
2. Templates (before and after assembly) contain all logic, variations, and answers in a single file, which can be awkward for complex and multi-part documents, although those are rarely found in consumer contexts. The single file architecture also has some distinct advantages, such as making it easy to create workflow-approval systems.
3. There are concerns about the continuing viability of Rapidocs given its small presence in North American and Epoch's recent financial difficulties in the UK.

GhostFill

GhostFill (http://www.ghostfill.com/), part of South Africa-based Korbitec (http://www.korbitec.com/), is a software development company with 20 years experience in the field of document automation, and with over 50 full-time developers working from its head office in Cape Town. GhostFill is well established in South Africa, having sold several thousand seats. A major new version - GhostFill2001 - was released in March 2001, aimed specifically at the international market. In 2001 GhostFill also established operations in North America, based in Toronto and headed by document assembly veteran Doug Simpson.

GhostFill offers a modern, object-oriented programming environment, and is especially strong on application and data integration. It provides a Windows-Explorer-like organization of templates and documents and a sophisticated "Fillpoint editor" for template building.

Another major release (version 4) is expected in December 2001. It will add improvements to the user interface, including dialog scripting, plus a PDF filler and a revamped core that supports a forthcoming server-based assembly engine that it plans to release in Q1 of 2002. This engine will be available both on IIS (NT Server, Windows 2000 Server, Windows NT, 2000, XP, etc) and Linux configurations.

The version 4 desktop and the Server Edition will have support for graphical forms assembly to PDF format. End users will require only Adobe Acrobat Reader, whereas forms designers will require a full version of Adobe Acrobat. Users will be able to capture information via dialogs or from databases (such as case management systems) and feed that information to PDF forms.

Development of templates for the Server Edition will be done in the existing desktop version of GhostFill (3.05 or 4) and templates so developed will run on the server. GhostFill Server will support an HTML rendition of dialogs that are developed in the desktop version of GhostFill. The Server Edition will support full document logic and nesting as in the desktop environment. It will be optimized to support high-speed, multi-threaded assembly. It will also support a variety of mechanisms for data capture, including data capture in dialogs, XML data, and data links to database servers via ODBC (e.g. MS SQL Server/Oracle/MS Access).

GhostFill has already been employed in some Web-based environments, including several applications in South Africa where GhostFill 2.5 is the document generation engine, and two large-scale legal applications in North America where data capture occurs over the Web and assembled documents are delivered to end users.

To mention some possible negatives: GhostFill does not directly support WordPerfect, which many legal services offices still use, although it has excellent support for RTF files, which can be opened and edited in WordPerfect. Its combination of developer and user versions in a single product raises some concerns both about price and end user confusion. And it has only a tiny installed base so far in the United States.

SmartWords

Ken Frank's Technology Group in Baltimore has long had one of the most advanced and complete visions for high-end document assembly software. Like Rapidocs, SmartWords provides its own word processing environment, which allows it to deliver exceptional interactivity and re-assembly of edited documents. (Re-assembly of edited documents is not presently supported in products that rely on external word processors, such as HotDocs or GhostFill, although this can be accomplished through special techniques and add-ins.) The Technology Group also has a very robust online assembly and e-commerce-ready environment. See http://www.lawontheweb.com.

Over 1500 law firms use one or both of the Technology Group's current commercial applications – Wealth Transfer Planning (a comprehensive estate planning and drafting system) and DistribuGuide (a retirement benefits application.) Both of those applications exist as desktop and Web versions.

Some possible downsides for SmartWords include the complexity of its more advanced development tools, the company's primary interest in commercial publishing rather than custom applications, and recent financial difficulties.

Grantha

One new entrant to the online legal document assembly space is S Square Technologies of Salt Lake City, Utah (http://www.ssquaretech.com). Their Java-written, Web-based document assembly tool, called Grantha, was developed in the context of an award-winning project for the Utah Courts (http://courtlink.utcourts.gov). They are also working with Utah Legal Services. Their pricing model is based on applications -- for around $25,000 you get the right to develop and deploy up to five applications (each containing multiple templates), with no user or runtime charges.

S Square released Grantha at the annual court technology conference this year in Baltimore. It is a pure Web-based application generator, with XML support and JDBC connectivity to several back-end databases. Using Grantha you can create an Internet application that will take users through a question-answer dialog session, capture the user responses, and then generate a customized document that is downloaded to the user's computer for printing.

S Square claims that its applications can be developed without any programming other than writing stored procedures for calculations. Other benefits it points to include:

- The dialog session can be set up for English, Spanish, and up to ten different languages.
- Being a pure Internet based application, all of the Grantha components can be run through the browser. Both IE and Netscape are supported.
- It generates Microsoft Word documents, which can be viewed using the freely available Microsoft Word Viewer, on any platform.
- As this is a thin client implementation, there is no need for additional software (other than Word viewer) on the client machine.
- Applications can be rapidly developed, deployed, and maintained.
- As there is no programming requirement, typically a business process expert can do the implementation. There would of course need some training on Grantha application development, which has a Web based user interface.
- Grantha can be used for applications that will facilitate easy document generation (for the public) and also for applications that will just solicit information from users (e.g. jury information) or to provide some checklist information or advice on the Internet.
- S Square is working on making Grantha available as an ASP model and will also support PDF and WordPerfect in their next version.
- The information provided by users during the question/answer session can be saved and updated to any backend database.
- S Square Technologies has an engineering team of over 40 engineers who can provide help with implementation of projects using Grantha or with making custom enhancements / backend integration work for these applications.

Possible negatives are the present lack of direct support for WordPerfect and PDF, the necessity of a constant Internet connection both for users and developers, and the absence of substantial experience in the legal field.

Some other platforms and approaches

Adobe Acrobat (PDF)

A number of online document assembly sites use Adobe Acrobat (http://www.adobe.com) technology to fill graphical forms. As mentioned before, the term "graphical forms" generally refers to documents with extensive graphics (usually in the nature of boxes and frames), often requiring the placement of text in precise locations. An example in the legal context would be a court form with fill-in boxes.

Out of the box, in desktop mode, Adobe Acrobat offers extensive functionality for on-screen filling of forms. Adobe also distributes a free "Reader," which allows the user to view and fill documents, including forms, created by the full version of Acrobat, but not to save the filled forms, or the associated answers. Acrobat files use a proprietary format, referred to as “PDF,” these letters forming the file name extension. Because the Reader is free, the PDF format is a popular choice among government agencies and court systems when it comes to distributing forms in electronic versions. Many electronic filing systems require it.

Adobe publishes an extensive API (Application Programming Interface), which allows developers to incorporate Adobe functionality in their programs. This is generally the approach used in online implementations. The standard Adobe license does not permit deployment of fillable forms that can be saved using Reader. Some implementations rely on a special license from Adobe that permits users to save the filled forms. Others simply allow the completed forms to be printed.

Adobe Acrobat is not designed to produce editable text documents (such as those one might create in a word processor) from templates. Using Acrobat for this purpose is possible, however, and one commercial product, Cardiff's (http://www.cardiff.com) Audience One, can be used to do so.

Advantages of an Adobe Acrobat approach include:

- Acrobat provides reliable display and printing across a wide range of platforms.
- The Acrobat reader program is free, widely available, and often already present on a client computer. No additional local software is required.
- Adobe has a corporate giving program that can make the authoring software available free.
- Acrobat forms can be pure facsimiles of court and governmental forms that can’t accidentally be changed.
- Help notes and nonprinting instructions can be made available.
- Version 5.0 supports collaborative editing and annotation of forms from within a Web page.
- The forms are easy to fill-out and can do basic calculations.
- Security can be imposed to prevent forms from being changed, copied from, or even printed.

Disadvantages of an Acrobat approach include:

- There is no built-in support for a separate “interview” that can guide a user through a complex form.
- Users can’t save filled forms or associated answers with the free Reader.
- Documents that require post-assembly editing in a word processing environment cannot easily be supported.
- Acrobat forms lack some features of more sophisticated programs like HotDocs Automator, whose forms can include interactive dialogs, conditional fields, conditional and repeated pages, more sophisticated calculations and error checking, built-in support for answer storage between sessions and use across

> forms, and advanced overflow handling (shrink to fit, automatic appendix creation.)

Note that PDF technology may fit well in hybrid implementations – e.g., users could be taken through a Cold Fusion information-gathering session, and the answers used to populate a PDF form via a utility like ActivePDF or a custom Perl script. Several of the engine vendors mentioned above have or will soon have support for PDF.

Jnana

Jnana, from Jnana Technologies Corporation – http://www.jnana.com – is a pure Web-based "inferencing engine" that can be used to design and deliver interactive sessions that guide a user through facts and considerations to reach a legal (or other) decision, based on a set of pre-programmed rules. It is being used in several projects at the Public Interest Clearinghouse in California. Jnana does not have a document assembly module per se, although it can create texts as part of a session, and now has a built-in link to HotDocs Online and could be similarly integrated with other platforms.

Lotus Notes

Iowa attorney David Hirsch, whose firm provides free assembly of wills from its Web site (see below), provided the following summary of their approach:

> Lotus Notes is a database oriented program designed for collaborative interaction. It is transparent to the Web. One can set up Web enabled document assembly through a Lotus Notes database using standard Lotus Notes mechanics of forms, fields and views. This approach is extremely powerful; it can take advantage of many features that the most powerful document assembly programs still lack. It also lacks some features that simple document assembly programs have. Three things make the Lotus Notes approach to assembling documents unique: 1) its database orientation (that is a powerful concept), 2) its collaborative strength; and 3) its natural Web plumbing.
>
> The cost of ownership ranges from free to expensive, depending on your skill level and setup. Minimum requirements are one Domino Web Server, and one Lotus Notes Client (with Designer). If you are already using Lotus Notes, there is no additional cost.
>
> Domino Designer is now $546. Domino Server is $1,941. Domino Client is from $32 to $80. You only need one of each. One should have Designer and Client on the desktops of everyone who will be involved in programming. If you know nothing about Lotus Notes and want to do this yourself, it will take you years to figure it out, even though the concept is simple. Notes does not realize what it has, and there is no manual on how to use Notes to assemble documents. Hiring someone to do your installation is possible. The cost of that ranges from reasonable to high, depending on the price and skill of the programmers hired. It is not difficult to find skilled Lotus Notes programmers. While one can use "lower level" programming with Lotus Notes, such as Java and Lotus Script,

> document assembly can be designed entirely with Lotus Notes' relatively easy high-level formula language and “visual programming principles” such as property boxes for easy selection of behavior and appearance.

Hirsch reports that all of the Iowa district courts use Lotus Notes to assemble jury instructions, and that his office uses it to create interrogatories over the Web.

Microsoft Office XP

Microsoft Office XP has recently been released, and a huge developer community is aggressively developing solutions using its new technologies, which have useful document automation applications.

Smart Tags

Smart Tags, introduced in Microsoft Word 2002 (and other products in the Office XP suite), permit particular text to be associated with a menu of actions. The text may already exist in the document or be recognized as the user types it. The appearance of the text (by default a red dotted underline) indicates the presence of a Smart Tag. Placing the mouse over the text triggers the appearance of a distinctive icon next to the text, which, in turn, can be clicked to reveal the menu of actions.

The text recognized can consist of specific words. One could create a Smart Tag that recognizes the word “client”, for example, and pop ups a list of existing client names from a database. A Smart Tag can also recognize a generic category of text, such as a name, address, etc. Very simple Smart Tags can be created without programming, relying on an XML file to hold the definition. Any reasonably sophisticated Smart Tag requires programming.

As they now function, it does not appear that Smart Tags will replace traditional document assembly engines. If Smart Tags were used in a document template, the user would need to find the relevant text and invoke the Smart Tag actions. The existing document assembly engines in effect handle this automatically. Smart Tags could supplement or extend document assembly programs, however, by providing a powerful way to make the text in a document dynamic even after the assembly process is complete.

Word 2002 ships with some Smart Tags, and third party developers can be expected to offer others. Microsoft’s Web site for third party Smart Tag offerings had some impressive items: data insertion and integration tools for simple document assembly, and post-drafting update and verification tools for legal citations. The New-Zealand-based ActiveDocs product (http://www.keylogix.com) makes use of Smart Tags technology. Smart Tags could enable users to free-form draft, copy from other documents, and pull together smart clauses and variable-rich boilerplate using sophisticated document assembly logic, all at the same time. Post-assembly data revision, collaboration, and business-rule workflow and routing could be initiated and controlled while still in the document.

You can implement Smart Tags by creating a Smart Tag plug-in in Microsoft Visual Basic and associating it with a document. The plug-in determines the text strings to evaluate and the actions that are offered to a user. A plug-in is made up of a recognizer component and an action provider component. The recognizer component specifies the list of text strings that the document will recognize and annotate as smart. The action provider component specifies the menu of commands that are associated with strings. For information about using Microsoft Office Smart Tags, see the Smart Tags software development kit, available on the Microsoft Office XP Developer installation CD.

Custom software

Of course, one final strategy is to proceed without any specialized document assembly engine at all. Many law-related Web sites have used generic Web technology (active server pages, Cold Fusion, etc.) to deliver interactive question-and-answer sessions and build customized documents.

While all document assembly projects involve a substantial amount of custom work, most knowledgeable document automation developers recommend not "rolling your own" *tools* if you can avoid it. Especially for advocate-oriented implementations, the vast number of specialized document automation features found in commercial tools are very hard to replicate. See the following table for a summary of some pros and cons of commercial and custom tools.

Advantages of custom development over commercial tools
Flexibility in features and interface May provide functions not available commercially Can be shared freely and collectively elaborated in an "open source" spirit
Advantages of commercial tools over custom development
Usually much less expensive for comparable functionality, since costs are spread over a wide groups of users Less likely to be "orphaned" Access to a community of fellow users for support (technical and moral) Integration with third-party software may already be supported Up and running much more quickly

Selection criteria

Here is a high-level, five-part way of looking at the major considerations one should take into account in judging a technical solution for a Web-based document assembly project. A more detailed framework for product comparison is included in Appendix II. The first two categories appropriately deal with the critical issue of **usability**. Note that some of these criteria only make sense for certain kinds of projects.

1. *User friendliness* – Ease of learning and use for the end user. Minimal local software requirements. Ability to run with any or no word processor. Speed of download and operation (of program components and templates). Ability to function offline. Ability to see document during the assembly process. Ability to observe during the assembly process how specific choices affect text in assembled documents. Ability to support re-generation when there are post assembly edits. Ease of providing and accessing help.

2. *Developer friendliness* – Ease of learning and use for template authors. Functional completeness of development environment. Modularity: ability to re-use pieces of templates and associated programming within systems or across systems. Efficiency of development and maintenance. Top quality technical support for developers. Ability to document logic for lawyers and non-technologists. Flexibility, evolvability. Open interface. Support for graphical forms.

3. *Web readiness* – Deliverability through browser. Browser independence (ability to run at least in recent versions of Internet Explorer, Netscape Communicator, and AOL browsers). Interoperability and scalability of applications. Support for a variety of server operating systems and Web servers.

4. *Price worthiness* – Fair licensing prices and reasonable total cost of ownership.

5. *Vendor stability and partner-friendliness* – Financial and managerial stability. Extent and nature of existing user community. Willingness to partner and implement custom requirements, including support for different languages. Ability to establish and maintain long-term relationships.

Different solutions will be strong (and weak) in different areas, and their appropriateness will of course depend on the specific characteristics of the project a program is undertaking. It's useful to take a "balanced scorecard" approach to product comparison, and look for at least satisfactory ratings across all important dimensions.

Integration with other applications, such as case management

Document assembly operations both require and produce information closely related to the kinds of data typically covered in a case management program. For example, a case management system may track the very same information about the client – such as his/her name, address, spouse's name, and income sources – that is also needed to produce the court filings handled by the document assembly system. There are thus naturally quite a few ways the two types of applications can interact:

(*Using* data from a case management system or other database)

1. A case management program can be configured to automatically invoke a document assembly application, specifying an appropriate template and passing along necessary data. This is sometimes called a "push" approach.

2. A document assembly application can be programmed to access information that is stored in a database ordinarily maintained by an office's case management program, again sparing the user from data re-entry. This is sometimes called a "pull" approach.

(*Adding to or updating* a case management system or other database)

3. The *fact* that a particular document has been generated can be stored a case record.
4. Data gathered from a user during a document assembly session can be sent back as an addition or update to a case management system or other database.

The benefits of such integration are obvious. Integration will help avoid duplicative data entry. It makes sense, for example, to populate documents with data drawn from an existing client database, rather than rekeying the same information in document assembly. And, it minimizes – or, depending on the implementation, eliminates – the chance that the data may be inconsistent.

Although the precise implementations vary, there are basically two approaches to integration. One is the aforementioned "push" technology: The user runs another application or database, initiating document assembly from within the other application, which launches the document assembly process by calling the assembly engine, specifying a desired template, and supplying it with data. The other is a "pull" approach: The document assembly software, alone or in conjunction with an add-on utility, accesses a database and retrieves the information it needs to assemble a document.

The push technology may come off the shelf or be home grown. A number of major case management software publishers offer integration with HotDocs, for instance, in addition to native document generating mechanisms. Programs with this capability include Time Matters, LawBase, and Amicus. These are generic programs aimed primarily at the legal market. They and competing products are designed to be customizable to fit the requirements of a specific practice. We have also seen custom integrations with Kemp's Caseworks' case management program and Lotus Notes in legal services contexts.

Many vendors equip their case management offerings with native document assembly functionality, either alone or in addition to HotDocs integration. The native functionality is generally somewhat rudimentary, lacking many of the features found in the stand-alone document assembly software.

Off the shelf integration is sometimes limited in terms of what data can be sent to the document being assembled. For example, a case management program may permit you to associate multiple "contacts" (individuals and institutions) with a particular matter, but only include one such contact in a document. This may or may not be a problem, depending on whether you have a specific application which requires the ability to include multiple contacts in a document, such as might be required for a list of witnesses in pending litigation. In the typical legal practice situation, including legal services, there is a fairly high likelihood of bumping up against integration limits.

You can also roll your own push technology, developing custom applications that address the document assembly engine. This requires that the developer have access to the inner workings of the document assembly engine, with enough information to write code that manipulates it. This can take the form of an API (application programming interface) or an open architecture, such as a COM server. In either scenario, any number of programming languages or development environments can be used. Not all document assembly offerings, however, offer an easy integration path.

Developing applications to control document assembly engines requires a fairly high degree of expertise, which means hiring a programmer, either on staff or outside consultant, to handle the development.

Pull technology, which may avoid the need for skilled programming talent, is often a more economical alternative. A number of document assembly packages include the capability, either built in or by way of an add-on, to fetch data from a database and populate a document. Some of these do not scale, however, and performance will suffer under heavy loads. A large number of users accessing the database, reading a large number of records from the database simultaneously, or some combination of the two may produce unacceptably slow access times.

In any of these scenarios, one commonly encountered issue is where and how to store data that is used in document assembly sessions but is not part of the information normally managed by the case management system. For example, the details needed to complete a financial statement in a divorce usually go well beyond what is stored in a case tracking or client/matter system. If this kind of information is stored in answer files or database records peculiar to the document assembly system, how are the various “islands” of data connected, and which if any can be relied upon as definitive?

A systems administrator’s view

Operating system and platform choices

While the chosen platform for development and hosting of an application should for the most part be transparent to the end users, it can greatly impact the available development methodologies, as well as other considerations, such as security concerns, scalability issues, and support possibilities.. Some of the high level alternatives for any Web application implementation include:

1. Microsoft based implementations (NT)
2. Sun Solaris (or other Unix based implementations)
3. Linux based implementations
4. Other operating systems and development environments

Each of the above systems can be considered for operating system cost, development environment costs, support costs, security issues, ease or difficulty in finding support and development staff, as well as availability of off the shelf products. As likely individual components of an implementation are considered, how they would fit within an operating system and with what development platforms should also be considered.

Server based vs. client based

While the World Wide Web allows browsers to provide client[1] access to server based content, this approach alone can be limiting in that HTTP-based systems rely on passing information back and forth between the client and the server to maintain a process flow. A constant connection is required, and the concept of a session or "state" can only be achieved through cookies or other work-arounds. Other solutions are possible, including client based technologies, and non-HTPP-server based solutions. While Web browsers that communicate over HTTP are extremely common (they are found on all newly purchased PCs, and are included with Linux, Solaris, and other operating systems), other clients could be utilized if good mechanisms are provided for delivering the client to the desktop for installation, and if good support mechanisms for such a client are in place. Other client based systems rely on off the shelf products with support from the vendor, and content delivery mechanisms for extracting data appropriately and securely from a server, manipulating the information on the user's desktop, and potentially returning updated information to the server. Synchronization of data stores on a user's desktop and on a server can be a complicated issue, and poses security concerns.

Implementation choices

Some of the available solutions for a document assembly project are packages with well defined functionality. Other solutions allow for more custom development and integration. Where a particular product would fall needs to be considered against the desired flexibility and degree of customization. Integration with other desktop applications (covered above) can be greatly enabled or hindered by the operating system, implementation languages, and off the shelf product choices.

Operations and support

Any document assembly system once deployed will require attention to ensure continued good performance as usage grows, continued security from intentional and inadvertent acts, as well as general maintenance for the possibility of systems failures. Built-in redundancy can be critical. Consider the availability of skilled personal to ensure that the above needs are met, and that expectations are met for supporting installed software and the systems that it runs on. Client systems also must be considered. While Web based (HTTP browser) systems may not require technical support systems other than documentation of data entered into the system, proprietary systems with unique clients may require a specific technical support infrastructure, or at least quality documentation and online resources.

[1] "Client" in this section obviously does not refer to a human, but to a machine or software that is receiving service from a computer "server." In most cases, the "client" is a local workstation.

Security

To the extent that client-specific data (e.g. answer files and/or database records) are stored on or accessible from Web servers, security of data is a critical issue. Attorneys have an affirmative duty to maintain the confidentiality of client information in order to guard the client's privacy, protect the attorney client privilege, and prevent the disclosure of case strategies to opposing parties, among other reasons. Thus any document assembly system must ensure that only those individuals who have the right to view or modify entered data have access to it. Encryption of data should be available to prevent access to sensitive information if system security is breached. Authentication mechanisms must intelligently deal with lost passwords, security attacks etc. Mechanisms for continually monitoring and reviewing systems to ensure security should be considered as well. These issues are of concern both when choosing and installing a Web server and in choosing and installing a document assembly solution on the server. In a *pro se* environment accessible from, for example, home computers or public libraries, this is even more critical as such systems are more visible to potential hackers.

Scalability

Document assembly project are often intended to support a broad user base. Strategies for doing so include segmenting the system to support smaller constituencies, and providing a larger system that appropriately deals with these constituencies. Use of data should be maximized where possible so that e.g. common document templates entered within the system are available to all, or to many. All of the above issues including security and flexibility come into play here. Scalability, however, should be ensured, so that performance doesn't suffer in delivering content and systems to all members of the target audience. Platforms and products used to implement the system can greatly impact the ability of a system to scale.

Scalability in larger systems may also be obtained by having multiple servers available to handle client requests. Components for parallel processing, load balancing, fault tolerance, etc. should be evaluated.

WHERE can you get more information?

Articles and books

Introduction and overview

Lauritsen and Soudakoff, Power Tools for Document Preparation. *AmLaw Tech*, Spring 1998. Also at http://www.capstonepractice.com/amlaw6.pdf

Sprowl, Automating the Legal Reasoning Process: A Computer that uses Regulations and Statutes to Draft Legal Documents. 1 *Am. B. Found. Res. J.* 1-81 (1979)

Legal services applications

Lauritsen, Delivering Legal Services with Computer-based Practice Systems. 23 *Clearinghouse Review* 1532 (April 1990)

Product reviews

Soudakoff and Lauritsen, Shopper's Guide to Legal Document Assembly. *Law Office Computing*, October/November 1997 (Most of the article is available at http://www.docauto.com/locart.htm.)

Online services

Calkins and Granat, Client Self Help Strategies: Technology Educated And Assisted Pro Se With And Without Advocate Backup (1998). http://equaljustice.org/visions/TechConf/09-strategies.htm

Granat, From Legal Services to Information Services. Internet Practice Newsletter, May, 1997. Available at http://www.granat.com/legalservice.html.

Hornsby, William. Improving the Delivery of Affordable Legal Services Through the Internet: A Blueprint for the Shift to a Digital Paradigm. http://elawyering.org/what/improving.asp

Lauritsen, Assembling Documents on the Infobahn, *WORD Progress*, Summer 1997, p. 14, http://www.abanet.org/lpm2/newsletters/wp/su97laur.html

Project management

Lauritsen and Soudakoff. Unlocking the Power of Document Assembly. *Law Office Computing*, June/July 1999, p. 70-77

Artificial Intelligence

Branting, K., An Issue-Oriented Approach to Judicial Document Assembly, *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, pp. 228-235, ACM Press (1993)

K. Branting, C. Callaway, B. Mott and J. Lester, Integrating Discourse and Domain Knowledge for Document Drafting, *Proceedings of the Seventh International Conference on Artificial Intelligence & Law*, pp.72-81, ACM Press (1999)

Lauritsen, Knowing Documents. *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*. Amsterdam, June 1993.

Lauritsen, A Dispatch from the Document Automation Trenches. Workshop on Automated Document Drafting. *Seventh International Conference on Artificial Intelligence and Law*. Oslo, June 1999

The bigger picture

Katsh, M. E. (1995). *Law in a Digital World*. Oxford University Press: New York.

Richard Susskind, *Transforming the Law* (2000)

Zorza, Legal Information Access for Poor and Middle Income People, and for the Organizations that Advocate for them (2000), http://www.zorza.net/legalinfo/index.html

Web sites

General

http://www.elawyering.org/
http://www.equaljustice.org/
http://www.lawofficecomputing.com [good access to product reviews – requires password from the table of contents page from a current issue]
http://www.lawschoolconsortium.net/
http://www.technolawyer.com [does a “best of” every year, including document assembly]
http://www.unbundledlaw.org/
http://www.zorza.net/resources/lst-res.html

Directories of online forms

http://www.allaboutforms.com/
http://www.alllaw.com/forms/
http://www.blumberg.com [has Blumberg’s Forms Online; also DL Drafting Libraries for all states except LA]
http://www.divorce-forms.com [Online divorce, automated separation agreements for all 50 states, using Rapidocs.]
http://www.divorcelawinfo.com/
http://www.draftinglib.com
http://www.easy-divorce.com/
http://www.easylegalforms.com
http://www.findforms.com/
http://www.ilrg.com
http://www.ilrg.com/callforforms/

http://resource.lawlinks.com/Content/Templates_of_Legal_Documents/templates_of_legal_documents.htm
http://www.legal.net
http://www.legaldocs.com
http://www.lib.memphis.edu/gpo/forms.htm
http://www.marylandlawonline.com/
http://www.nolo.com/product/forms_home.html
http://www.quickforms.net
http://www.uslegalforms.com/uslflawfirms.htm

Products and vendors

Document assembly development tools for legal contexts

(most not yet significantly Web-enabled)

ActiveDocs	http://www.keylogix.com
Boilerplate	http://www.wordsite.com/Boilerplate.html
CAPS	http://www.capsoft.com
DAS@H	www.das-h.com "Document assembly system @ hyperspeed"; used by http://www.legaladviceline.com
Docuscribe	http://www.docuscribe.com/index.html
eDrafter	www.docdev.com Formerly Legal Knowledge Systems, New York. Now Document Development Corporation (DDC)
FastDraft	http://www.fastdraft.com Valeo Software Has a new Document Automation Server that supports Web-based applications
GhostFill	http://www.ghostfill.com
Grantha	http://www.ssquaretech.com S Square Technologies
HotDocs	http://www.capsoft.com
IQDocs	http://www.iqdocs.com focus on loan documentation
KillerDocs	http://www.killerdocs.com
Lawgic	http://www.lawgic.com not available for external development
Legal Ease	http://www.legal-ease.net
OwlCentral	http://www.owlcentral.com Docdolittle
PowerTxt	http://www.interconweb.com/html/powertxt.html
ProDoc	http://www.prodoc.com
Rapidocs	http://www.rapidocs.com
SmartPrecedent	http://www.speedlegal.com/smartprecedent.html SpeedLegal

SmartWords	http://www.lawontheweb.com
ThinkDocs	http://www.thinkdocs.com DataTech Software (800)556-7526
Visual eForms	www.mmacorp.com Publishes EILA immigration forms package. Also bankruptcy, divorce, and California Judicial Council forms.
WinDraft	http://www.lawtech.com/WINDRAFT

Legal document assembly engines no longer marketed

Agility
Expertext/FormBank
FlexPractice
JumpStart
Legalware
MasterDraft
Millrace
NovaDocs
Scrivener
WorkForm
… and many others

Related products that are not law-oriented:

activePDF – http://www.activepdf.com
Amgraf – OneForm - http://www.amgraf.com/pages/iforms.html
Calligo Enterprise from InSystems Technologies (Toronto) – insurance industry focused - http://www.insystems.com/products/calligo-index.htm
Cincom – http://www.cincom.com
ConText, from USCFX – http://www.uscfx.com
Document Library Services from Document Sciences – http://www.documentsciences.com
DocX In a Box, from DocX http://www.docx.com
Drug Documentation System, from Michael Umen & Co.– http://www.natson.com
JetForm – company now called Accelio - http://www.accelio.com
LiquidOffice from Cardiff – http://www.cardiff.com/
Napersoft – http://www.napersoft.com
NETdelivery Corporation – http://www.netdelivery.com/products/eforms.shtml
WordShare from www.cetara.com

Case management software with document assembly features or links

Amicus	http://www.amicusattorney.com
Clients for Windows	http://www.kempscaseworks.com Kemp's Case Works
LawBase	http://www.lawbase.com
Legal Files	http://www.legalfiles.com This product has its own document assembly engine, but does not integrate with external products out of the box. You can create templates in either Word or WordPerfect and generate documents in the same. You can insert fields from the case management database or insert fields the user is prompted for and that get inserted during assembly. Legal Files is used extensively by Legal Services of North Carolina for letters, pleadings and other documents. (Contact: Gray Wilson) Documents created with Legal Files templates are automatically associated with the appropriate client or office file and are listed in the Documents window, so it acts as a powerful document management system also. The template editor is fully compatible with MS Word (in fact, will become MS Word in an upcoming upgrade), so it's easy to cut-and-paste from existing documents In addition to allowing you to insert the values of database fields in templates, it allows for variables (called "user tokens") as well, with a variety of data types supported. It allows you to conditionally insert clauses, but there is no branching logic like IF..END IFs. Repeats per se are not supported, but multiple values can be inserted from the database. Like similar products, there is a document management/calendaring system which is pretty tightly integrated with the document assembly engine (it logs documents assembled, can notify others within the office, calendar responses required etc.) (There is limited support for bitmaps, none for headers and footers, etc.).
Practice Manager (Real Legal)	http://www.reallegal.com/ Has its own document assembly features; also has HotDocs integration built in. Used by Legal Services of North Texas (contact: Charles Grimm)
Prolaw	http://www.prolaw.com
TimeMatters	http://www.timematters.com

WHO is doing related work?

Illustrative projects in legal services contexts

AARP – Legal Counsel for the Elderly

Summary	D.C. Office on Aging grant for two “virtual legal services offices” with tools that include client-oriented document assembly and public benefit checkups.
Parties/contacts	Wayne Moore
Technology	Rapidocs
Current status	Operational

Bay Area Legal Aid (California)

Summary	Integrated document assembly and case management systems. 2000 TIG grant
Parties/contacts	
Technology	
Current status	

Central New York Legal Services

Summary	Documents for landlord-tenant, SSI, matrimonial, etc. About 340 forms. Used in about 20 programs at one point.
Parties/contacts	Jim Murphy, Cortland
Technology	WordPerfect 5.1 macros and merge codes
Current status	Still in use in Cortland.

Detroit

Summary	Legal Aid & Defender Association (Detroit, Michigan) Civil Law Group, Litigation & Resource Group and Private Attorney Involvement use document assembly via desktop and intranet.
Parties/contacts	Sharon Horner Gant, Esq., developer, Alicia Bembry, administrator
Technology	HotDocs, HotDocs Pro, Microsoft Word 2000
Current status	Development started late 1999, Abuse and Neglect appeals template in use starting Spring 2000, Estate Planning in use starting Spring 2001.

Georgia Legal Services Domestic Violence Protection Project

Summary	Web site for victims of domestic violence. Includes interactive questioning to prepare papers needed to request a protective order. Can click on highlighted words for definitions. Information not stored on server for confidentiality reasons.
Parties/contacts	Richard Zorza, Fund for the City of New York
Technology	
Timeframe	1996

Current status	

Greater Boston Legal Services

Summary	GBLS has had extensive document assembly systems in family, housing, and government benefits contexts since the mid 1980s.
Parties/contacts	Bob Sable, Capstone
Technology	Document Modeler (past), CAPS, HotDocs, Lotus Notes
Time frame	1985 – present
Current status	Ongoing use.

Harvard Legal Aid Bureau

Summary	The Harvard Legal Aid Bureau has built and used document assembly systems for eviction defense, divorce, and other practice areas since the mid-80s. No online dimensions yet.
Parties/contacts	Betty Allebach, Larry Farmer, Marc Lauritsen
Technology	CAPS, HotDocs
Time frame	1985 – present
Current status	Ongoing use.

I-CAN! Project (Orange County, California)

Summary	The I-CAN! (Interactive Community Assistance Network) system is one example of an ambitious online document assembly initiative in the legal services world. Developed with the leadership of the Legal Aid Society of Orange County, the system is designed to be used either in a kiosk setting or on an Internet connected personal computer. The system, in operation, but still being expanded and enhanced, is designed to produce graphical forms for court filing in a variety of civil matters. The nine forms currently available include child support, domestic violence protection, denial of license revocation, small claims, landlord and tenant, and license denial revocations. I-CAN! offers services in English, Spanish and Vietnamese, although not all interviews are available in all three languages. Many of the forms can be filled out online. In some instances, one can simply print out a form and fill it manually. Either planned or in the works are the conversion of these to online fillable forms. In all instances, the actual forms (constructed to court specifications) are in English. The system is in use in various public locations, including libraries, courthouses, shelters, and district attorney's offices in Orange County. Five surrounding county legal services organizations are planning to deploy the system. I-CAN! currently localizes the forms and instructions to the court at which they will be filed. One of the hallmarks of I-CAN! is the use of video to instruct the user

	in how to fill out the form and file it. At present, acceptable video quality is only available with a high-speed connection. It is anticipated that planned technical changes will make it possible for users with a 56K dial-up connection to view the videos. The videos consist of a “talking head” explaining each step of supplying information and filing the documents. For online fillable forms, the user does not actually fill a visual representation of the form. Instead, a series of interview screens, with accompanying video instructions, elicits the required information. At the end of the process, the completed form is rendered and printed out, along with written instructions about filing procedures. Development is handled by six people, including project director A. J. Tavares. The team includes a designer, who meets with the staff attorneys to glean the information needed to provide the content, a quality assurance person, and three developers who write the code. Translation services are outsourced. Development efforts began in July, 2000, and cost $485,000.00 to date. Tavares estimates that approximately two thirds of the development effort involves meeting with the staff attorneys. The content of the I-CAN! system is geared for a fifth grade reading level. Each interview now includes a user survey (for English and Spanish speakers only), and Tavares reports that 90% of English speakers rated the system helpful or very helpful. 78% of Spanish speakers gave it the same range of marks. From March through September, 2001, 2,687 people used I-CAN!. Among the future plans for I-CAN! are the building of a bridge to the electronic filing systems contemplated for California's courts. A demonstration of I-CAN!, along with further information, can be accessed at http://www.legal-aid.com.
Parties/contacts	Robert Cohen, executive director; A. J. Tavares, project director
Technology	The I-CAN! has been built in-house from scratch. The implementation is Microsoft-centric, using ASP (Active Server Pages) code and using Windows Media Player to deliver the video. Although only Internet Explorer (version 5 or later) is supported at present, Netscape users will be able to use the site shortly. Microsoft's SQL server 2000 is the data storage backend. The technology was chosen in part because of the relative abundance (and hence lower cost) of developers trained in Microsoft tools. The court forms are delivered via an Adobe Acrobat plug-in.

Legal Services of New York - online eviction defense app

Summary	Eviction defense system; preparation of tenant's answer.
Parties/contacts	Richard Zorza, Fund for the City of New York; state courts?
Technology	
Timeframe	
Current status	

Maryland Legal Aid Network

Summary	Rapidocs-based site (discussed above), designed to serve the MLSC clients so once a client is authorized by a staff member they get all the documents for free. There is a revenue-sharing component so that non-eligible clients will be referred to the mylawyer/maryland site and if they buy documents the revenue is dedicated to maintenance of the non-profit site, so this an experiment in public/private partnership. MyLawyer has guaranteed to MLAN that they will not have any hosting or maintenance costs going forward, but they paid a reasonable set-up fee up front.
Parties/contacts	Ayn Crawley, Richard Granat; www.mdjustice.net
Technology	Rapidocs
Current status	Operational

Neighborhood Legal Services (Lynn, Massachusetts)

Summary	Divorce template set.
Parties/contacts	John Welsh, Ross Dolloff; Capstone
Technology	HotDocs, integrated with Kemp's Caseworks system
Current status	In use

Peoples Law Library (Maryland)

Summary	The Peoples Law Library of Maryland provides not only on-line information about various family law, probate and landlord-tenant problems, but also an online Child Support Calculator and the necessary forms to complete most family law cases, many of which can be prepared on line.
Parties/contacts	http://www.peoples-law.com/
Technology	
Current status	Operational

Public Interest Clearinghouse

Summary	2 projects: (1) Fair housing intake and (2) employment law audit.
Parties/contacts	Aleem Raja, araja@pic.org. Bay Area Legal Aid; the Volunteer Legal Services Program of the San Francisco Bar Association. Private law firms are also involved in both projects.
Technology	Jnana
Current status	Field testing #1; #2 in design stage.

Pine Tree Legal Assistance (Maine)

Summary	http://www.ptla.org/forms.htm Site provides on-line, court approved, PDF forms, including family law forms, some other court forms, and several legal services specific forms. Interactive forms perform some calculations and provide “rollover” help. Forms are provided in conjunction with extensive on-line *pro se* assistance materials and reference to other *pro se* resources. The forms are used by legal services staff, *pro se* litigants and private law offices. (An earlier site, abandoned in favor of the on-line PDF forms project, walked users through the process of preparing the myriad forms necessary to a divorce action in the State of Maine. Like the Georgia Domestic Violence Project, it was designed to ask only those questions that are appropriate, based by the information already submitted by the user. After asking a series of questions, the program printed out eight pleadings necessary to file in most divorce cases.)
Parties/contacts	Hugh Calkins http://www.ptla.org/
Technology	Using JavaScript and Adobe Acrobat.
Current status	Completed and on-going. In September 2001 more than 20,000 forms were downloaded, more than half of them interactive. Figures probably overstate use because of Acrobat Reader’s refusal to allow completed or partially completed forms to be saved.

Washington State

Summary	*Pro se* Forms for Domestic Violence Victims: Internet forms application that uses a question-and-answer interview format to obtain information from the petitioner and then automatically fill in six forms (the petition, temporary order, final order, information sheets, etc.) needed to file for a protection order. Ready-to-file forms can then be printed on the user's local printer.
Parties/contacts	Developed in partnership between the Washington Administrative Office of the Courts and the Washington State Access to Justice Board. Brian Backus (Washington Administrative Office of the Courts), Brian.Backus@courts.wa.gov.
Technology	Cold Fusion
Time frame	2000 – 2001
Current status	Available to courthouse facilitators, domestic violence advocates, domestic violence support organizations, county clerks, and superior courts in Washington state. “We plan to add more forms in other areas including domestic relations and make them available on our public Website. We would like a more rapid development than we had with Cold Fusion and plan to look at document assembly products.”

Court-sponsored projects

California Judicial Council

Summary	Self-Help Center includes forms and self-help information.
Parties/contacts	www.courtinfo.ca.gov/selfhelp
Technology	
Current status	

Indiana

Summary	Indiana has opened a Website with family law forms and other information. See www.IN.gov./judiciary/selfservice/index.html or www.state.in.us/judiciary. Hendricks County, IN, was to open one of the state's first court-based legal self-help centers at the courthouse on September 6, 2001.
Parties/contacts	Judge David Coleman of Danville, IN, was the team leader at the 1999 national conference in Scottsdale, and is a member of the Indiana Pro Se Advisory Committee, which developed the site.
Technology	
Current status	

Maricopa County, Arizona

Summary	Quick Court kiosks in Maricopa County, Arizona - http://www.maricopa.gov A kiosk based interactive multimedia computer system that uses text, graphics and an on-screen narrator to provide important information and instructions about the process to litigants and produce legal documents for use in court cases. QuickCourt is specially designed to guide individuals through their cases, with instructions in both English and Spanish. For a small fee, the system will produce documents for use in court in divorce and child support cases, probate, small claims and landlord-tenant cases (eviction complaints). For no fee, the system provides general information on the court system, ADR, family crisis and other matters. See more generally http://www.supreme.state.az.us/selfserv/forms.htm.
Parties/contacts	
Technology	
Current status	

Philadelphia

Summary	Philadelphia Municipal Court – http://courts.phila.gov
Parties/contacts	done by Veralaw Technologies
Technology	lots of online PDF forms, unclear how many if any fillable
Current status	

Utah

Summary	Online Court Assistance Program for the public to prepare uncontested divorce and landlord/tenant dispute documents. The application received an honorable mention in the NACM justice's competition and was also ranked as #2 in the top ten Websites in the US by justice served, inc. The application can be accessed at http://courtlink.utcourts.gov
Parties/contacts	Kim Allard, kima@email.utcourts.gov
Technology	Grantha from S Square Technologies Linux, Apache, Informix, JDBC, RTF; moving to XML, e-filing
Time frame	Went live in November 2000
Current status	Well received by public as it is very easy to use. County clerks are now getting documents that are consistent, easy to review. By November 2001, about 10% of divorce document generation will be through this system. They are currently in the process of implementing other new applications.

Washington Supreme Court – has statewide *Pro se* Forms committee

Commercial online legal information and service providers using document assembly technology

http://www.desktoplawyer.com/
http://www.divorcelawinfo.com/
http://www.hrlawinfo.com/
http://www.lawexpress2.com/
http://www.lawontheweb.com/
http://www.legaladviceline.com/
http://www.lexisone.com/
http://www.marylandlawonline.com
http://www.mylawyer.com/
http://www.namechangelaw.com/
http://www.uslaw.com/ [Once claimed to have “the only Web-based document wizard.”]
http://www.visalaw.com/ [Has an online registration form for the “green card lottery”]

Other related projects

Beckman & Hirsch

Free online will preparation using Lotus Notes. They also have pro bono medical power of attorney and living will.
www.iowalaw.com

Large law firms offering Web services

Clifford Chance, Linklaters, Allen & Overy, Davis Polk, Blake Waldron, …. See, e.g., http://www6.law.com/ny/tech/092500t1a.html.

Community Catalyst

Developers of MicroMax. Now has online Food Stamps analysis and application generator. http://www.micromax.org (old site)

Acknowledgements

This information gathered here draws upon the work of dozens of dedicated individuals over the last fifteen years. We gratefully acknowledge the support of the following people who have been especially helpful in the compilation and refinement of this report:

Tad Bohlsen – Indiana Legal Services
Richard Granat – MyLawyer.com
David Hirsch – Beckman & Hirsch
Michael Hertz – ProBono.net
Jay K. Jayaraman – S Square Technologies
John Kemp – Kemp’s Caseworks
Michael Mills – Davis Polk & Wardwell
Joyce Raby – Legal Services Corporation
Seth Rowland – Basha Systems
Doug Simpson – GhostFill Technologies
Ron Staudt – Chicago Kent School of Law
A. J. Tavares – I-CAN!
Richard Zorza

About the authors of this report

Capstone Practice Systems (http://www.capstonepractice.com) provides software development and training services to a wide range of law firms, legal departments, publishers, government agencies, and other organizations. It has worked with many legal services and other nonprofit groups. Capstone has special expertise in CAPS, HotDocs, and GhostFill, but has had hands-on experience with nearly all similar technologies, past and present, and is an independent consulting firm with good relations with other vendors. Its principals are world leaders in the comparative assessment of law practice tools and their online deployment.

Capstone president ***Marc Lauritsen*** practiced and supervised in legal aid offices for seven years, then served as an instructor, director of clinical programs, and a senior research associate at Harvard Law School. Marc directed Project PERICLES there, which focused on computer applications in legal services. He was chair of the American Bar Association's document assembly interest group and moderator of the law office automation forum on Counsel Connect. Last year he served as vice president for practice technology at AmeriCounsel.com, which developed an online environment for low-cost, high-quality legal service delivery through a nationwide network of private attorneys.

Capstone vice president ***Alan Soudakoff*** is an attorney who specializes in developing complex document automation systems for major law firms and legal services organizations and in integrating such systems with case management and document management software. He has been a technology consultant to law firms for the last eight years, including two years with the Law Firm Services Group of Price Waterhouse in Washington, D.C.

Capstone senior consultant ***Bart Earle*** is a programmer and technology consultant with a 23-year history of private legal practice. In addition to maintaining an active caseload and managing operations, he was his firm's technology guru. For the past six years, Bart has done intensive work in document automation, building law practice systems with HotDocs. His recent experience includes constructing systems for delivering legal services on the internet and evaluating new platforms.

Capstone affiliate ***Richard Farrell*** is a programmer and manager who has worked on the cutting edge of information technology for nearly a decade. He was the founding CTO both of Boston.com, New England’s leading regional Website, and AmeriCounsel.com, and managed application development and integration for many years at the Boston Globe. Rich has expertise in Internet/Web development and large-scale architectures that support millions of visitors per month. Specific technologies include Java, JSP, Perl, Cisco, Sun Solaris, Linux, XML, Sybase and Netscape iPlanet Servers.

Appendix I: A Project Definition Framework

No two document assembly projects are the same, even within the same organization. Here are some of the ways projects differ. Where does yours fit?

Users	Who are the intended users of the application? Lawyers, paralegals, secretaries, students, clients, unrepresented individuals? Are they proficient in the area in question, or novices? Are they a few or many? Do they work in proximity, or are they spread among floors, offices, or cities?
Purposes	What are the driving goals of the application? To speed up processing? Improve quality or consistency? Achieve greater capacity? Allow work to be delegated to more efficient staffing levels? Assist in training? What else?
Documents	What documents is the system designed to produce? Short and simple, long and complex, or somewhere in between? Are they typically first-draft-final, or do they require lots of post-assembly editing? Are official, graphical forms involved? Are the documents typically produced individually, or in related sets? Can they be neatly handled with fill-in-the-blanks variables and alternative/additional passages, or do they involve lots of material that doesn’t lend itself to straightforward rules? How often do the documents change or need to be updated?

Scope	What range of transactions is the system intended to support? How deep do you intend to go in modeling the variations from transaction to transaction? Is the system designed only to produce first drafts, or guide users through several stages of revision and negotiation? Should it offer project management and decision support features?
Novelty	Is this the organization's first effort of this kind, or one among several? Are the team members experienced in this kind of thing? (There are vast differences between doing a first project of this kind and later ones. They involve different states of organizational and personal readiness.)
Staffing	Is the project being done entirely with in-house personnel, by an outside consultant, or some blend? Is it conceived as a project by lawyers, aided by others, or as an initiative of an IT team, aided by lawyers?
Technology	What underlying software will you use? How will you allocate needed functions between the server and the desktop? If your implementation involves both advocate- and consumer-oriented applications, will they be driven from a shared knowledge base?

Appendix II: A Technology Evaluation Framework

This evaluation framework is organized as a questionnaire for vendors and other solution providers. There are hundreds of features around which solutions differ, and conflicting vocabularies for describing them. We've tried to include here those considerations most often important for selecting a solution in the nonprofit legal services context.

To respondents: Please submit your response in a Word document containing a verbatim copy of this Appendix, sent to marc@capstonepractice.com by December 10, 2001. Keep in mind that responses will be shared with potential customers and fellow providers, and may be made public. If you offer several alternative solutions – e.g., one that makes most sense for pure Web-based *pro se* clients, and another that makes sense for advocates to run on their local machines – consider separating them into two responses for greater clarity. Please avoid "puffing" – be as objective and specific as possible.

1. Product or solution	
1.1. Name of product or solution	
1.2. Version number and release date	
1.3. Provider name and contact information (address, phone, Website)	
1.4. General description – nature of solution; intended users	
1.5. What kinds of nonprofit legal services applications do you think your solution is most appropriate for, and why?	
2. Pricing	
2.1. Please describe your pricing for different versions and scenarios. 2.1.1. Do you offer volume discounts? 2.1.2. What does annual software maintenance cost (and cover)?	
2.2. What donations or discounts do you offer for nonprofit legal services providers?	
3. Technology requirements	
3.1. Minimum hardware specifications (both for server and desktop) Processor speed RAM recommended Disk space required for full install Other requirements	
3.2. Can your solution be accessed through a browser? What browsers and versions do you support (IE, Netscape, AOL)?	

3.3. Operating system requirements or limits (both server and desktop)	
3.4. Network operating systems requirements or limits	
4. General features	
4.1. Word processors and formats supported	
4.2. How do you deal with fixed-format, graphical forms?	
4.3. Do you support the generation of PDF files? How?	
4.4. Where and how do you support XML? 4.5. Are user and developer tools integrated or separate?	
5. User interface	
5.1. What user help is built in? What can be added to applications?	
5.2. Can users easily move backwards and forwards through the questioning?	
5.3. Do you provide an on-screen question tree that allows users to jump non-sequentially among questions, and that adjusts dynamically to reflect currently relevant questions?	
5.4. In what ways do you support default answers, so that users can move more quickly through routine information?	
5.5. Does the interface allow entry of repeated information (e.g., children, gifts, borrowers, creditors) in a straightforward manner? Can users enter an arbitrary number of items, and easily modify, reorder, and delete such items? Can users enter more than one answer for each item (e.g., debtor name, account number, current balance), including second-level repeated information (e.g., several gifts for each of several beneficiaries in a will)?	
5.6. Can answers be used in the prompts of subsequent questions?	
5.7. Do you allow multiple questions on single dialog?	
5.8. Can you dynamically change, gray, and hide questions and texts on dialogs based on user answers? How?	
5.9. Can a user suspend and resume a session? How does that work?	
5.10. Do you support answer revision by clicking on hot spots in the assembled document?	

5.11. In what other ways does your solution promote ease of use by end users?	
6. Developer interface	
6.1. What control do developers have over interface look & feel?	
6.2. How can your application development environment be used by non-programmers to efficiently create interviews and document models? 6.2.1. Is the creation and revision of computed variables (pronouns, husband/wife, child/children, etc.) do-able in a straightforward way? Describe. 6.2.2. How do developers gather and use repeated information (like names of children)? Is there support for sorting and filtering such lists? nested repeats? 6.2.3. Can developers use repeated information in automatically punctuated lists? on separate lines or in table cells? to drive repeating phrases, paragraphs, or other sections of a document with appropriately iterated answers merged within those sections?	
6.3. Are answers to questions validate-able by data type (text, number, date, etc.) and range to minimize user error?	
6.4. Can answers be validated against prior answers?	
6.5. What scripting does the product support? if....then statements? nested if....then statements? how deep? repeat loops? nested repeat loops? how deep? subroutines? graying/hiding in dialogs?	
6.6. Can developers define logically discrete document components (phrases, paragraphs, sections, etc.) of arbitrary size, from a single character to multiple pages?	
6.7. What features or approaches help developers document logic for lawyers and other non-technologists?	
6.8. Can the application development environment be used by a dispersed team of authors? Can more than one developer be working on an application at a time?	

6.9. In what other ways does your solution promote ease of use and developmental efficiency by application developers?	
7. *Pro se* oriented applications	
7.1. Can your application work with any or no word processor?	
7.2. How do you support users who use different word processors but want to share common templates?	
7.3. Does your solution support video or other multimedia? Touch screen technology?	
7.4. What components need to be downloaded and/or installed? What is the speed of download and operation?	
7.5. Do you support languages other than English? Which and how?	
7.6. What other features do you offer that are particularly appropriate for *pro se* users?	
8. Advocate-oriented applications	
8.1. Do you offer automated document preparation functionality roughly comparable to that found in mainstream legal industry document assembly products?	
8.2. Can developers assign styles (e.g., margins, tabs, typeface, automated paragraph bullets and numbers) to textual components that will activate in the assembled document?	
8.3. Can you handle dynamic numbering, automated cross-references, tables of contents, and other features?	
8.4. Do you support clause assembly (where users pick desired document components from a library of some sort)?	
8.5. Can templates launch macros in the end user’s word processor?	
8.6. What specialized merge formats for numbers and dates (e.g. “three and sixty-eight one-hundredths percent”, “the fourth day of March, Nineteen Hundred and Ninety Six”) do you support? Do any require special coding?	
8.7. Can your applications be used offline?	
8.8. How do you handle spreadsheet-like data input – e.g., assets for husband/wife/joint in wills questionnaire; income/expense items in bankruptcy with varying periodicities. Can you do on-screen running totals?	

8.9. Do you support post-editing reassembly of documents that retains edits yet applies template logic? How?	
8.10. What other features do you offer that are particularly appropriate for lawyers and other advocates?	
9. Application integration	
9.1. What database connectivity do you support?	
9.2. What support do you provide for integration with case management, document management, and other applications?	
9.3. What add-in modules are available?	
9.4. Do any third party applications include built-in support for integration with your solution?	
9.5. What tools or support do you provide for conversion of applications built with your software to or from applications built using other software?	
9.6. Describe any API, SDK, or other features that relate to the openness of your solution	
9.7. What else should we know about the interoperability and scalability of applications built with your technology?	
10. Provider characteristics and user community	
10.1. What can you tell us about your financial and managerial stability? How long have you been in business? How many employees do you have, and what is their average length of service?	
10.2. What can you tell us about your willingness to partner and implement custom requirements? Your ability to establish and maintain long-term relationships?	
10.3. What automated content based on your platform is available commercially? Do you or others offer ready-made templates that run with your software? If so, describe.	
10.4. Who is using your technology, and how?	
10.5. What is the nature and approximate extent of your installed base?	
10.6. What uses of your technology have been made that come closest to the needs of the nonprofit legal services community?	
10.7. What kinds of user support do you provide, and what does it cost?	
10.8. What technical support do you provide for	

developers	
10.9. What training programs do you or others provide?	
10.10. What developmental and other consulting services do you provide, and on what terms?	
10.11. What resources do you make available on your Web site?	
10.12. Are there other sites with relevant material?	
10.13. Are there any user groups for your products?	
10.14. Do you or others run conferences for users or developers?	
10.15. Is there a listserv relating to your solution?	
10.16. Are there any newsletters or other publications that cover your product?	
10.17. Are there any independent specialists with expertise in your technology? Resellers? Certified consultants?	
10.18. Please identify any reviews or case studies that have been published about your solution.	
11. Additional information	
11.1. What forthcoming releases and features can you describe?	
11.2. What special features or advantages of your solution haven't been captured in answers to the above questions?	
11.3. Does your solution involve any particular limitations people should be aware of?	
11.4. What limitations or disadvantages of competitive solutions do you feel we should be aware of?	
11.5. Name, title, and contact information for person completing this questionnaire	