Building Clinical Information Spaces on the World Wide Web

Paul Wehr, STATPROBE, Inc., Ann Arbor, MI

Abstract

In clinical research, a clinical investigation is made up of a research protocol, data, analyses, and a research report. Each component depends on, or refers to, information contained in the other sections, and within itself. The goal of Clinical Research—to demonstrate that a drug or medical device merits approval—is really nothing more than an information space. This paper will show that using SAS® Software, and the capabilities of the Web, one can create a convenient system for navigating this space, and even lay the foundation for a fast, inexpensive, and effective computer-assisted new drug application (CANDA).

Introduction

In March of 1989, Tim Berners-Lee proposed the idea that a large amount of information could be conveniently maintained and accessed by storing it in a non-linear (hyper-text), distributed environment. Since then the idea has exploded into the "World-Wide-Web", a ubiquitous intangible, replete with gratuitous commercialization and an endless stream of so-called enhancements (animated GIF images, push media, meta-tags, Java, and the like). This paper is about none of the gee-whiz, because-you-can add-ons to the WWW, but is rather an example of the power the Web has when used for it's original purpose: building an easily accessible, networked, cross-referenced information resource.

In applying for approval of any drug or medical device, the sponsor attempts to make it clear that the drug or medical device is both safe and effective. These two assertions are, of course, based on the results of research studies, which are, in turn, developed from what appeared in the data. The data itself is based on the directives and assumptions of the research protocol, interpretation of the data collection forms, and immeasurable factors influencing the patient. All of these factors will weigh in, to some degree, on the quality of the two statements, and therefore on the approvability of the drug or medical device.

Obviously, the sooner a regulatory agency reviewer can assimilate this information, and draw their own conclusions, the sooner the drug can be recommended for approval. Accelerating this process has two parts: 1) Getting the submission to the Food and Drug Administration (FDA) as quickly as possible, and 2) Making the submission as easy to review as possible. Because the Web is designed to be a document storage medium, and a clinical research report is ultimately just a large, interconnected document, creating a Web-based submission should help with both of these.

What is an Information Space?

The core of a submission to the FDA, is the new drug application (NDA), a 2-page form that is often supplemented with hundreds of thousands of pages of appendices. The new drug application has 18 sections, including a clinical data section, a statistical section, a safety update section, each of which can have dozens of subsections. Although each section contains specific, detailed information on its particular topic, the information in each of these sections often cross-references to other sections. Taken as a whole, this collection of inter-related information constitutes a good example of an information space.

In the traditional paper format of an information space, even finding the page containing the information desired in the thousands of volumes can be a significant effort. However, for more and more clinical research projects, a good share of the information is likely to already be in electronic form. The electronic data can be created in a variety of formats: Word processing documents, SAS data sets, scanned image files, etc. This makes the task of creating a Web-based information space much simpler, as we need only to map the information from one format to another, rather than create the content in the first place. Much of the content of a research report is derived from the results of the clinical trial, which in most cases is stored in SAS data sets. As SAS programmers this is to our benefit, as it allows us to do most of this mapping in SAS.

Implementing an information space in HTML can be addressed in three key steps:

- Creating the HTML documents
- Organizing the information (linking them together)
- Implementing interactivity

Creating HTML

In order to create a Web-based information space, the clinical information should be mapped to hypertext markup language (HTML) format whenever possible. The types of information appropriate for a clinical information space include:

- Protocol
- Analysis plan
- Research report
- Tables/listings/figures
- CRF Tabulations
- Scanned CRFs (or equivalent)
- Electronic database
- Programs used to generate reports

The first three are often created using a desktop word processor, and most software packages today have an option to save the document as HTML. CRFs would need to be scanned, but this is becoming common practice for archival reasons, which can simply be leveraged into Web content. The database and programs will most likely not be converted to Web pages in themselves, but will instead have an interactive form that gives the user access to them. For the remaining items (Tables/listings/figures and CRF Tabulations), HTML files will have to be created.

There are several techniques for implementing this conversion, and there will likely be a number of papers in this section presenting these and other techniques in more detail. I will give a brief description of some of them here. The table below provides a quick summary of the document storage methods described in this section.

Method	Implementation (1=hard 5=easy)	Hyperlinks?	Platform Independent?
Encoding as text	5	No	Yes
Embedding tags	3	Yes	Yes
Coding HTML reports	1	Yes	Yes
Print driver system	3	Yes	Yes
SAS/Intrnet	3	No	Yes
SAS/ODS	4	No	?
SAS/MDDB	4	Yes	Yes
Non-HTML file formats	2	No	No

Presenting SAS reports as text

This technique involves sending the data to the client browser with the MIME type "text/plain." This is usually done by giving the file an extension that the server will recognize as text, such as ".txt". This is the easiest method to implement, as it requires no modification of “standard” SAS reports, as it supports documents that rely on a fixed-pitch, space-delimited output format. The only modification necessary involves routing the output to its appropriate place on the Web server.

Embedding HTML tags in "standard" SAS reports

If most of the programming that supports your research report has already been completed, you can wrap the entire table in "pre-formatted data" HTML tags (<pre></pre>), and embed anchor tags (<a></a>) as appropriate. This allows the use of existing DATA _NULL_-based reports, but allows for the inclusion of hyperlinks. For greater convenience, a simple macro can be created that will allow minimal modification of existing reports, and, with a global macro variable, can be switched on and off depending on whether an HTML document, or a standard report is desired.

Writing HTML directly

Another approach is to write HTML documents directly from a DATA _NULL_ step. This approach is flexible, but requires a thorough understanding of the intent and use of all the applicable HTML tags. It also necessitates a complete re-write of at least the output portion of any existing SAS programs If the reports depend heavily on analysis and reporting hybrid tools like PROC TABULATE or PROC REPORT, then without SAS/ODS (Output Delivery System) a complete re-write may be necessary. This approach is capable of created reports with the highest quality and most flexibility of any of the techniques listed here.

Print Drivers

Using the system I presented at SUGI 21[1], you can use the same DATA _NULL_ step described above, although a thorough knowledge of HTML is not required. This approach has the added benefit of being able to generate paper-destined formats like rich-text format (RTF) and PostScript without modifying the report programs in any way. The majority of the listings and summaries generated for this paper were created using this system. For more detail on the interface to this system and some of the philosophy behind it, I encourage you to refer to the SUGI 21 Proceedings.

SAS/Intrnet®

SAS provides a data set-to-HTML table tool with their SAS/Intrnet product. It is likely not the optimal solution for particularly complex reports, as it is designed only to represent SAS data sets in HTML tables. However, it may have a role in creating convenient listing tables.

SAS/ODS

SAS is developing a compelling product in Version 7 that will allow SAS procedure output to be presented in HTML (and ultimately in a variety of file formats). This should make the mapping of data to a hypertext document very convenient for even the most novice SAS user. Hopefully by the time this paper is published, SAS version 7 will be available, or on its way. One limitation to this approach is that it is available to SAS procedure output only, so if the report requires more customization than is available from PROC TABULATE or PROC REPORT, this solution may not apply.

SAS/MDDB™

SAS also provides a “multidimensional database” interface that is designed to create Web pages which summarize a particular class of data. This approach may be ideal for some types of summary tables. Furthermore, like the other SAS products mentioned here, it is supported by SAS, and can generally be expected to be easier to use and more reliable than some of the custom programming required using the other techniques listed. SAS has more information on their Web publishing products on their Web page[2]

Non-HTML markup

Because of the implementation of MIME-encoding in most Web browsers, the format of the documents that make up the information space do not necessarily have to be in HTML. It is not uncommon to find Web sites that link to pages in Adobe Acrobat™ (PDF), or other document formats. Although SAS does not generate any of these formats natively, many user sites have developed, or employed third-party software, that performs a conversion of sorts on the standard SAS output into other formats.

Care should be taken with considering alternative formats, as many formats do not support embedded hyperlinking, which considerably restricts the advantages of a Web-based information space. Also, non-HTML coded documents do not automatically enjoy the platform-independence that HTML documents do. For example, a document in a PC application format is likely to not be readable by Web browsers that do not have the required application, or are on a UNIX or Macintosh platform.

Organizing your Clinical information Space

The key to creating an effective information space lies in choosing an appropriate, and fixed naming convention. The naming convention should be dictated by the way you choose to organize your information space. In my case, the space is generally organized in a top-down fashion, allowing the user to drill-down from the research report all the way to the scanned case report forms.

Of the information stored in SAS data sets, most will are presented in one of three formats: summaries (including figures), listings, and case report form (CRF) tabulations. While these distinctions have been arbitrary in the past, splitting these reports into categories aids in identifying groups to target with our hyperlinks. This, in turn, helps support our organization of the information space.

Summary tables

Summaries are reports based on the information contained in data sets. These can include anything from simple counts to comprehensive analyses. In our information space, these can be linked to the listings, CRF tabulations, or—if appropriate—derived data sets. Most of the tables directly referenced in the research report will be summaries. Some of these tables will actually appear in whole, or in part, in the body of the report. Others will be contained in one or more appendices. When choosing a target name for these links in the research report, it is tempting to use the reference numbers as table names. For example "Table 1.5: Summary of Demographics Information" may lead one to choose to store table 1.5 in a file called "table1_5.html" or the like. Unfortunately, this approach does not allow for the reality of tables being added and subtracted from a research report during its development. Changing table numbers would cause a link to connect us to the wrong table, forcing us to constantly update our links. A better method is to link to a name related to the content of the table. In this case, the name of the SAS program or macro that generated the report is probably more appropriate (assuming the SAS program was well named) and provides additional benefits when we add some interactivity, as we will see later.

In most cases, the SAS program or macro name that created a specific report are not unique, as a single program often generates more than one related table. To accommodate this characteristic, a suffix can be added to distinguish each unique table from one another. In this example, a separate directory is created for each program, and an arbitrary sequence number is assigned to each table (see figure below). Because the sequence number is a property of the SAS program, and not of the research report, it is effectively "behind the scenes," and is not affected by renumbering the appendices. To help organize these tables, an index page can be created in each program directory which contains links, by title, to each of the reports that program is capable of generating.

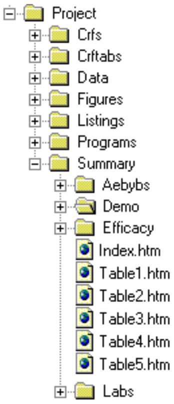

Figure 1: Directory structure

Listing tables

The summary tables created in our information space represent the distillation of information that is available in data sets. Listings are simply a human-readable interface to the data contained in these data sets. For this reason, it makes sense for parts of the summary table to have hyperlinks which target all or a specific section of the listing tables. Because of the one-to-one correspondence the listings have with the data sets, they are the easiest tables to generate. With a little foresight, and some minor programming, a listing table can be programmed so that the sort order can be specified as a parameter. This provides each value in the summary table to have a target to reference containing the information that makes up that statistic. For example, consider a summary of patient demographics by treatment group. For a specific value, say, the number of male patients in the placebo group, the value is linked to the "male" section of a listing of demographics by treatment group and sex. The number of Caucasian patients is linked to their corresponding section of the same report, sorted by race.

The listings referenced by the summary tables can, in turn, be linked to their corresponding section within the CRF tabulation reports, or the original case report forms themselves. If the listings reference the CRF tabulations (which are simply listings of patient data by patient, as opposed to by data type) They can be cross-referenced to the scanned images of the original CRFs. They can also serve as the target for extreme values in the summary tables, to address the question of "What factors was this patient exposed to that might account for this value?"

CRF Tabulations

The CRF Tabulations are arguably an information space in themselves. As a by-patient listing of every piece of information contained in the clinical database, they encompass all of the data that the summary and listing reports are based upon. Managing this volume of information requires careful organization. The CRF Tabulations are also a good example of a complex report requiring a customized DATA _NULL_ step.

This is where our fixed naming convention is most important. For multi-center studies, It is useful to separate the patients' reports into separate directories for each center. Again, the location of these files is entirely up to the designer, as long as the definition stays fixed, so that the listing and summary tools can embed their appropriate links.

In addition to the patient reports themselves, an index page can be created that contains links to each patient in the study. This link can organize the patients by center, if appropriate, and then have a patient identifier—patient number, initials or both—be a hyperlink to the corresponding page. This is a good example of the value of a Web-based information space. The CRF tabulations are often 10-20 pages per patient, and for a moderately-sized study of a few hundred patients, it is clear that these reports would easily fill dozens of binders. A user would find the ability to review the information on any patients with a few mouse clicks from their desktop far more appealing that hoisting huge binders of printed pages, and leafing to the desired pages in the central documents storeroom.

Information and Metadata

Hopefully it is clear that managing a complete information space demands a reliable naming and linking conventions. The job of keeping all of the components of the information space straight is made easier by enlisting the support of some metadata. The information space described in this paper is supported by two main tables.

The first table contains one record for each report created in support of the NDA. It associates each report with a program (in the "Programs" directory above), appropriate titles and footnotes, and an arbitrary sequence number. This sequence number is linked to the second table, which contains the settings for the parameters appropriate to generate that specific report. Each report that a single program is capable of generating will be listed with it's corresponding title(s) in the "INDEX.HTM" file within the directory assigned to that program. This will allow the user to select tables by title, rather than trying to remember which number goes with which report.

These tables are used on many occasions to help tie the information space together. In addition to providing the titles and labels for the table of contents, they also provide the Web form associated with each program the parameters that the user can change to obtain a variety of reports.

The two data sets described here are just examples of the value of metadata when creating a Web-based information space. Any amount of additional data may also be included. Metadata can, for example, identify the properties of each center in the study (location, primary investigator, etc.), name the primary and secondary efficacy variables, or record information about the study design. The more the information space relies on the metadata, the more robust and flexible it can be. Using metadata for the variable information in a study also provides for a higher degree of reusability of the programming infrastructure, as you can simply replace the data sets from one study with the data sets appropriate for another.

HTML forms, CGI, Search engines, and beyond

The SAS programs that generate the reports are themselves part of the information space, just as the summaries and listings are, and should not be overlooked. The information space would be incomplete without them. For those reports that are designed to be run with optional parameters, as previously described, a form can be created that allows the user to change those parameters to create reports that may not have been included in the NDA. This is especially useful for those reports based on flexible code, like PROC TABULATE.

If a user creates a new report, and wants to keep the report for later reference, the metadata design described above will conveniently accommodate that. When the user elects to store a custom table permanently, the system need only add a title, and store the corresponding parameters to the metadata base. The updating of the program index would be done automatically.

If all of the reports that support the NDA can be created simply by submitting the programs included with the parameters recorded in the metadata, why store the output at all? There are actually a number of reasons. One, of course, is performance. Activating a hyperlink on a summary table, and then waiting 5-10 seconds while the corresponding listing table is generated would seriously impair the usability of the information. Second, storing the output in a permanent file allows users that do not have access to SAS the ability to browse the information space in it's static form. Finally, having a collection of permanent files allows a search engine to automatically catalog the entire information space. An information space in HTML lends itself readily to cataloging by search engines. Search engine software can be used to create a table of contents for all permanently stored information, and a searchable index, where users can scan for specific types of information.

Conclusion

Creating a clinical research information space using the technology developed for use with the World-Wide-Web provides a quick, convenient, and integrated interface to the appendices of a NDA. Using the Web as an interface also allows the developer to inherit all the advantages of the Web, including the simplicity of development, simplicity of interface (reducing the training time to nearly zero), and the broadest platform independence of any software currently available. Furthermore, it's hard to imagine that the pharmaceutical and biotechnology industry will continue to ship paper-based NDAs in perpetuity. Clearly some sort of electronic storage and presentation system will take its place. With the ubiquity of the Web, and it's design as a document storage and distribution system, it is realistic to expect that it will play some significant role.

However, implementation of this approach requires some major re-thinking of business processes for companies who have been submitting NDAs on paper for some time. It also requires a certain level of acceptability by regulatory agencies. Although there are some challenges for early adopters of Web-based clinical information spaces, the benefits will be worth the investment to both the companies that sponsor them, and the patients they are trying to help.

Author:

Paul Wehr
STATPROBE, Inc.
3885 Research Park Drive
Ann Arbor, MI 48108
(313) 769-5000 x-188
pwehr@statprobe.com
paul@arbornet.org

Reprints of this paper are available at
http://pages.prodigy.com/paul_wehr/sugi.htm

[1] "%Print Drivers: Teaching SAS to Speak the Many Languages of Document Publication", Proceedings of the 21st Annual SAS User's Group Meeting, International

[2] "Web Enablement", SAS Web site
http://www.sas.com/software/web_enablement/