MedDRA Made Easy with SAS/IntrNet

Sy Truong, Meta-Xceed, Inc, Fremont, CA

Abstract

Working with a new thesaurus dictionary that contains multi-level hierarchy such as MedDRA can be a challenge. The decisions go beyond just matching a verbatim adverse event term with a preferred term. This paper explores tools developed with SAS/IntrNet to create a user friendly interface that helps make mapping decisions easy within MedDRA. It will describe how auto coding, manual coding, and the process of creating a mapped dataset can be accomplished through simple mouse clicks rather than writing programs. An example of a hyperlinked dictionary view will also show how easy it is to navigate the hierarchy of terms. Innovative use of email integration demonstrates how the tool communicates with the users. In addition to technical examples, this paper will also share the lessons learned in various project management tasks, as well as methodologies of working with a web-based application.

Introduction

There has existed a gap between technology and clinical data. Clinical information affects people in life and death situations, therefore creating a conservative environment resistant to change. Technology, on the other hand, progressively changes very rapidly. MedDRA attempts to bridge this gap by dynamically managing a dictionary of clinical terminology in a constantly changing environment.

This paper shares experiences of using SAS/IntrNet to develop a system to help users work effectively with adverse event clinical terminology managed by MedDRA. The client I collaborated with is a small pharmaceutical company who had used other thesaurus dictionaries in the past such as Costart and is transitioning to using MedDRA. They are a relatively small shop and do not use large database management systems such as Oracle Clinical or ClinTrial. These larger systems have companion technologies that work with MedDRA.

MedDRA is delivered in ASCII format on a CD. It can only function as a powerful and searchable dictionary with additional software. This created a perfect opportunity for the development efforts of the Thesaurex System™. We decided to use SAS/IntrNet since the data managers who interacted with MedDRA did not have SAS on their desktop. This allowed us to develop a system that only required a web browser on the client machine. The fact that we worked in a small organization also made it ideal, since there was little bureaucracy in deciding how to move forward with our development efforts. We collaborated by meeting once a week to review development progress. The project started with the use of Costart and WHODRUG. Since the industry is moving towards adopting MedDRA, it was a logical step for us to develop technologies and processes for MedDRA.

What is MedDRA?

A common example of clinical data which is collected is adverse events. That is, if a patient has a headache during the conduct of a clinical trial, this information is recorded into a database. The information collected is usually solicited from the patient. The patient may say various things such as: "head pain" or "pain in the head". Depending on the patient, the information collected can vary even though it is intended to express the same thing. When the data is analyzed, terms that have the same meaning need to be mapped in order for the aggregate statistics to make sense. This is when a thesaurus dictionary comes into use to consistently translate the information collected from the patient (referred to as verbatim) into a common meaningful preferred term.

MedDRA is one such thesaurus dictionary that contains clinically validated medial terminology used by regulatory agencies in the biopharmaceutical industry within the United States and internationally. The acronym stands for **Med** (Medical), **D** (Dictionary), **R** (Regulatory), and **A** (Activities). It is one of the more comprehensive databases containing terms collected in other dictionaries including: Costart (fifth edition), Who-ART (98:3), J-ART (1996) and HARTS (Release 2.2). The dictionary is also constantly being updated with new terms, so it is one of the most comprehensive dictionaries available.

Another aspect of MedDRA that sets it apart from its predecessors is that it is organized in a hierarchy. You can think of it as if the English dictionary were reorganized so that besides words being listed alphabetically, words were also categorized by subject. For example, consider the subject of computer science as a subject category. A sub-category of this can be the SAS software. In this example, the hierarchy consists of:

- Subject – Computer Science
    - Software – SAS
        - Alphabetical Terminology

In this fictitious example, there are three levels to the hierarchy. In the case of MedDRA, there are five levels to the hierarchy. This gives much more specificity and sophistication to searching and finding clinical information. In the Costart example, the hierarchy consists of the body system as categories such as: "Cardiovascular", "Digestive", "Nervous System", etc… This allows terms to be categorized so that interpretation and searches can be applied more accurately. The body system categories are helpful, but this creates a hierarchy of only two levels. MedDRA is organized with the following levels in its hierarchy:

- *System Organ Class (SOC)* – Anatomical or physiological systems
    - *High Level Group Term (HLGT)* – Categories such as: coronary artery disorder, cardiac arrhythmias and cardiac valve disorders
        - *High Level Term (HLT)* – Categories such as: cardiac conduction disorders, rate and rhythm disorders, supraventricular arrhythmias
            - *Preferred Term (PT)* – Single medical concept including: Arrhythmia NOS, Bradycardia NOS, Tachyarrhythmia
                - *Lowest Level Term (LLT)* – It relates to preferred terms in forms such as: synonyms, lexical or cultural variants

MedDRA is used to help pharmaceutical and biotech companies speak the same language when communicating with regulatory agencies. This is particularly useful when the company is submitting data to international agencies other than the FDA. MedDRA was started by the International Conference on Harmonization or ICH. It was then handled by the Maintenance and Support Services Organization or MSSO. It is currently being maintained by MSSO and new releases of the dictionary are released about twice a year. Companies constantly send new terms to MSSO and the dictionary is a living database that is updated to reflect the most up to date information.

More in-depth clinical information pertaining to MedDRA, beyond the scope of this paper, can be found at: www.meddramsso.com. This paper will focus on how SAS interfaces with the dictionary to in facilitate the looking up and mapping of verbatim adverse event terms to the desired preferred terms.

Web Application and Client Server are Different

Before I delve into the specifics of MedDRA, there were some challenges in developing a web based application that needed to be resolved. There are aspects of the thin client application that are very different from the client server application developed by tools such as SAS/AF. The following list describes some of the differences and presents suggestions and approaches for working with them.

- Authentification – Any multi-user application requires the users to log in or authenticate. This normally requires a user name and password. There are security features built into the HTTPS protocol that encrypt the communication between the client computer and the server. This ensures that passwords are secured even if the system is rolled out across the Internet. There are also nice built in HTML field types that can blank out your password as you type your password.

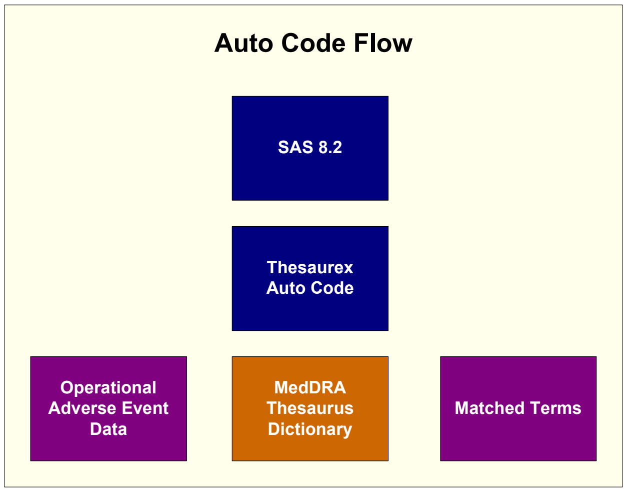
  ```
  <input type="password"
  name="password" size="20"
  tabindex="2">
  ```

  The user's name and password can be stored in a password protected dataset on the server. This ensures that the user can be authenticated through a browser while maintaining a secured system.
- Statelessness – The state of a user refers to where the user is at within the application. This usually refers to which screen or dialog box the user is currently interacting with. In a client server application, it is obvious since a user can only navigate from one screen to the next. In a web application, a page can be cached which means a user can click on the back button and enter through an old screen. This leads to possible interactions that are out of sequence. One strategy is to have a name assigned to each screen. Each user would then have a dataset stored on the server similar to a SASUSER profile. It would then store the latest information so that you can track each user at any point. You may choose to allow users to enter the system out of sequence or kick them back to the login screen if they enter through an older cached page.
- Navigation – The use of buttons is common among client server applications. A web application also contains buttons within the browser. The back button, however, can lead the user to navigate out of order. In addition to buttons, there are hyperlinks. These are usually underlined text or images that allow users to drill down with a single mouse click. The use of hyperlinking can make an application much more navigational. I suggest using both buttons and hyperlinks to make the navigation as easy as possible for the user. This paper will describe ways of using hyperlinks that make an application as friendly as a good website.

There are many differences between web applications and client server applications. Web applications development may be challenging and daunting at first, but there are many advantages.

MedDRA Auto Coding

One of the first steps in working with MedDRA is to perform an auto coding. This is when the source data which comes from the operational database is merged with the MedDRA dictionary to capture matches.



In the simplest scenario, the merge occurs between two tables by the verbatim terms.

```
data matched;
   merge meddra (in=A)
         source (in=B);
   by verbatim;

   *** Find matches only ***;
   if (A) and (B) and verbatim ne '' then
```

The MedDRA dictionary does not have a field named "verbatim" so the merge is actually accomplished through the lowest level term. The dictionary can also be very large so using an IN option to capture only items that have matches is a good idea.

Usually, the percentage of exact matches is very small. The majority of the verbatim terms are usually synonymous to existing matches but they are worded slightly differently. In this case, a decision is made to match the synonymous terms to their preferred term and then populate this decision to another internal dictionary. The synonymous tables form a knowledge base so that the next time an auto code is applied, the terms will be matched. The synonymous tables are considered a dictionary similar to MedDRA but it is "internal". The MedDRA dictionary is an external dictionary managed by MSSO.

In a more realistic example, there are multiple internal dictionaries, one for each project. It is possible that the decision to map one verbatim term differs from one study to the next. Clinical studies also have their own hierarchies. Different companies manage them in slightly different ways, but in general, it is usually:

- *Global* – This is the highest level that can be differentiated either by therapeutic groups or molecules of drugs.
  - *Project* – Projects contain a series of studies which fall within a drug molecule. Perhaps they are grouped by what the drug is trying to cure, also referred to as the indication.
    - *Studies* – These are individual studies conducted for each clinical trial.

This forms a hierarchy where there are multiple studies within a project. There are also multiple projects within a drug molecule.

It is common to define an internal dictionary for each instance in the hierarchy.

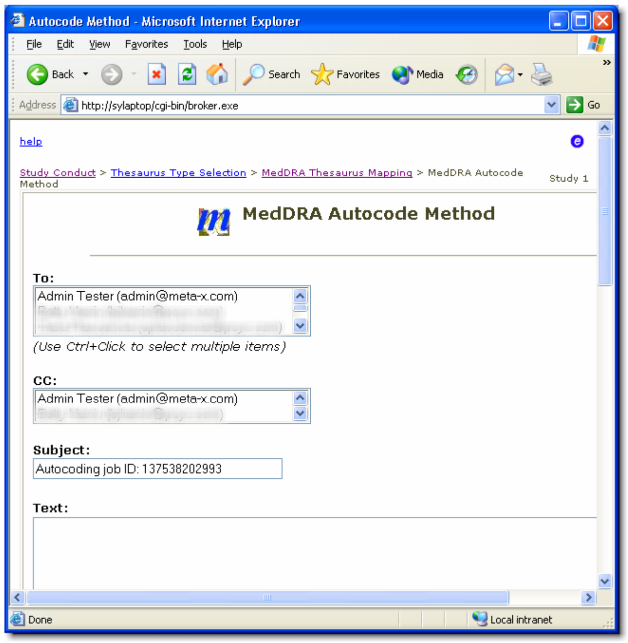
Auto coding is not just a single merge as shown above. Usually, it is a series of merges starting with the external MedDRA dictionary and then again with the global, project and finally study specific dictionaries. The above code is placed in a subroutine that gets called for each dictionary defined. As it loops through, the first time it finds a match, it uses that matching decision. That is to say, if it finds a match at the global level, it would not continue the merge down the hierarchy to the project for that matched term.

The entire auto coding process is encapsulated into a macro called *%medcode*. There are several parameters specifying things like which thesaurus dictionary to be merged against and in what order.

```
%macro medcode (thesname=, path=, datname=,
   varname=,
   thesin1= , thesin2= , thesin3= ,
   thesin4= , thesout=, outfile=, maploc=);

   *** Autocode ... ***;
%mend medcode;
```

The tasks of auto coding can be performed through a SAS batch program via a macro call. Many parameters make this process difficult for novice users, so a web interface is used.


Autocode Method - Microsoft Internet Explorer

Address: http://sylaptop/cgi-bin/broker.exe

help

Study Conduct > Thesaurus Type Selection > MedDRA Thesaurus Mapping > MedDRA Autocode Method   Study 1

MedDRA Autocode Method

To:
Admin Tester (admin@meta-x.com)
[redacted]
(Use Ctrl+Click to select multiple items)

CC:
Admin Tester (admin@meta-x.com)
[redacted]

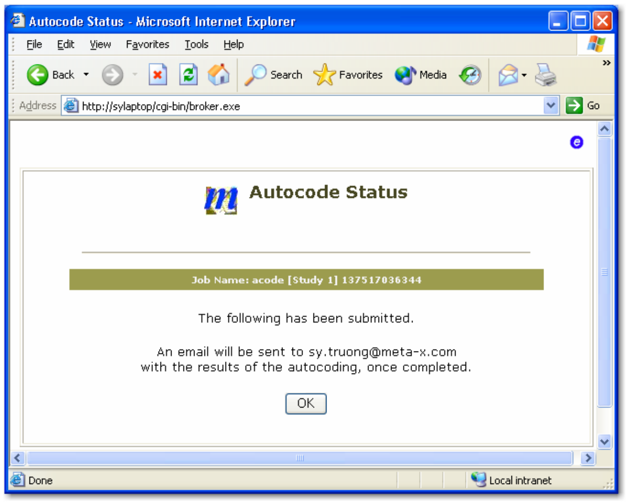
Subject:
Autocoding job ID: 137538202993

Text:

Done   Local intranet

In this case, the selections of dictionaries and data have been pre-selected in a mapping session. The auto code can therefore be repeatedly requested with just a few clicks.

When the source data is large and there are many terms, the merges can take some time. I have a general rule for web applications when it comes to response time. If the task takes more than three seconds, do not make the user wait. When the user clicks on a request to perform an auto code, they do not want to sit there and stare at the spinning browser connection icon. In this case the user will receive a confirmation that the request is placed.


Autocode Status - Microsoft Internet Explorer

Address: http://sylaptop/cgi-bin/broker.exe

Autocode Status

Job Name: acode [Study 1] 137517036344

The following has been submitted.

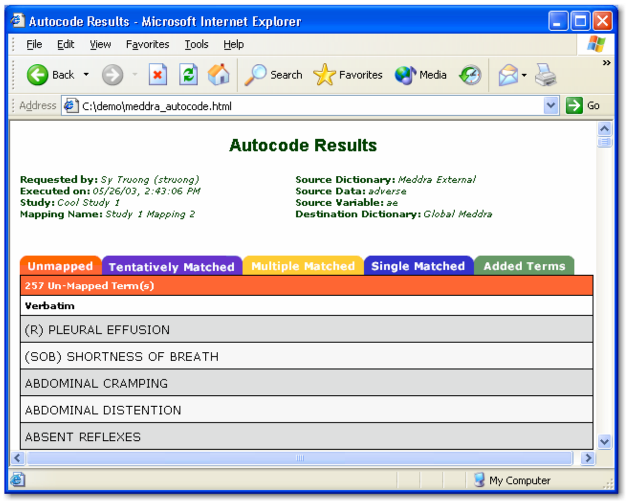
An email will be sent to sy.truong@meta-x.com
with the results of the autocoding, once completed.

OK

Done   Local intranet

An email is later sent along with the result of the auto code. This is an asynchronous approach to processing that is suitable for a web interface pertaining to tasks that take longer than three seconds.

The email delivered contains an interactive HTML report. Most email clients support the use of HTML as part of the body of the email message. It is therefore a good idea to use HTML rather than plain text when delivering content rich messages.


Autocode Results - Microsoft Internet Explorer

Address: C:\demo\meddra_autocode.html

Autocode Results

Requested by: *Sy Truong (struong)*
Executed on: *05/26/03, 2:43:06 PM*
Study: *Cool Study 1*
Mapping Name: *Study 1 Mapping 2*

Source Dictionary: *Meddra External*
Source Data: *adverse*
Source Variable: *ae*
Destination Dictionary: *Global Meddra*

Unmapped | Tentatively Matched | Multiple Matched | Single Matched | Added Terms

257 Un-Mapped Term(s)

Verbatim
(R) PLEURAL EFFUSION
(SOB) SHORTNESS OF BREATH
ABDOMINAL CRAMPING
ABDOMINAL DISTENTION
ABSENT REFLEXES

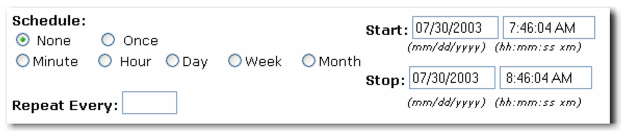
My Computer

The use of colored tabs helps organize the findings of matched and unmatched items. The HTML report is actually one long page with the tabs acting as local links to the portion corresponding to the clicked tab. This is accomplished by the use of some Java Scripting.

```
function linkHandler(loc){
   window.location.href = loc;
   if (loc=="#Page1")
   document.myForm.b_ok.focus();
   return;
}
```

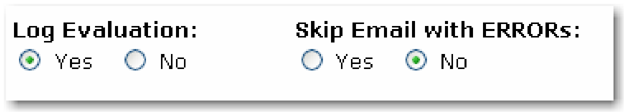
This gives the appearance of an interactive application even though it is just a static HTML file with no connection back to the server. This is an example of how a simple hyperlink in conjunction with arranging GIF images can turn static email into an interactive report delivery mechanism. Using email as a method for delivering information also allows you to send information to other users. By default, the email for autocode is used as a status report. The selection of additional recipients can also turn this process into a method of delivering information to fellow team members.

Besides email, there are other features that can be implemented when using an ascynchronized processing technique. Upon request, the task can be scheduled to be repeated.



Schedule:
[✓ None] Once
Minute Hour Day Week Month
Start: 07/30/2003 (mm/dd/yyyy) 7:46:04 AM (hh:mm:ss xm)
Stop: 07/30/2003 (mm/dd/yyyy) 8:46:04 AM (hh:mm:ss xm)
Repeat Every:

This allows auto code to be scheduled in the event that the source data or dictionaries are updated frequently. Upon completion of the scheduled job, the server will issue an email. Since the program executes like a SAS macro, a SAS log is produced. In this case, it is useful to have the log evaluated for errors and warning messages.

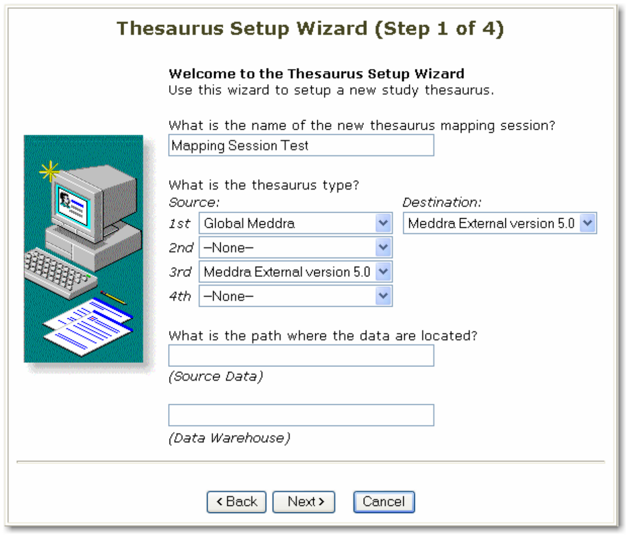


Log Evaluation: [✓ Yes] No
Skip Email with ERRORs: Yes [✓ No]

The log evaluation is a text file that gets attached to the email that is sent out. This is a useful feature for the recipient to quickly assess if the job was successful.

Manual Coding

After auto coding is performed, the user works with the list of unmatched terms. This involves matching unmatched terms to their preferred terms, also referred to as manual coding. Before manual coding can be performed, dictionaries have to be registered to the system. This involves a data entry screen that requires edit checks to ensure that the fields entered are valid.



Thesaurus Setup Wizard (Step 1 of 4)

Welcome to the Thesaurus Setup Wizard
Use this wizard to setup a new study thesaurus.

What is the name of the new thesaurus mapping session?
Mapping Session Test

What is the thesaurus type?

	Source:	Destination:
1st	Global Meddra	Meddra External version 5.0
2nd	–None–	
3rd	Meddra External version 5.0	
4th	–None–	

What is the path where the data are located?
(Source Data)
(Data Warehouse)

One of the requirements is that the thesaurus name cannot exceed 80 characters. If the process were to take place on the SAS server, the code would look like:

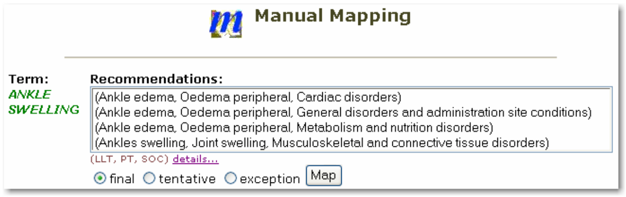
```
*** Verify maximum length ***;
if length(thesname) > 80 then do;
   put "WARNING: Length of thesaurus
   mapping session exceeded maximum
   length of 80 chars.";
end;
```

In this particular case, the application has to send a message back to the server for this evaluation. The results are then sent back to the browser. For such a simple check, it is much more efficient to do this in a JavaScript right on the client browser.

```
<!-- Begin
function Check() {
   if
(document.myForm.thesname.value.length >
80) {
   alert(
"WARNING: Length of thesaurus mapping
session exceeded maximum length of 80
chars.");
   document.myForm.thesname.focus();
   return false;
  }
// -->
</script>
```

The response is much faster since there is no network involved. The script also prompts the cursor upon the thesaurus field after the error message is displayed. SAS is on the server and it delivers more power compared to JavaScript. However, there are instances where it is more efficient to have edit checks performed on the browser via a script.

Thesaurex assists the user during the manual coding process. It makes an educated guess as to possible matches and displays a list of recommendations.



Manual Mapping

Term: *ANKLE SWELLING*

Recommendations:
(Ankle edema, Oedema peripheral, Cardiac disorders)
(Ankle edema, Oedema peripheral, General disorders and administration site conditions)
(Ankle edema, Oedema peripheral, Metabolism and nutrition disorders)
(Ankles swelling, Joint swelling, Musculoskeletal and connective tissue disorders)
(LLT, PT, SOC) details...

[✓ final] tentative exception

The recommendation list is accomplished through a series of logical steps.

- Strip Punctuation – The tool does not consider punctuation during searches, because it returns false search results.

```
if index('~!@#$%^&*()_+`1234567890-
=[]\{}|;:<>,./?',curword) > 0) then
do;
   *** Strip the punctuation... ***;
```

- Remove Trivial Words – Similar to punctuation, there are words that are not significant. Words such as “a”, “the”, “and” are considered trivial words. These words are also removed from the verbatim during the search for recommendations.
- Break up the Phrase – The verbatim phrase is broken down to individual words.

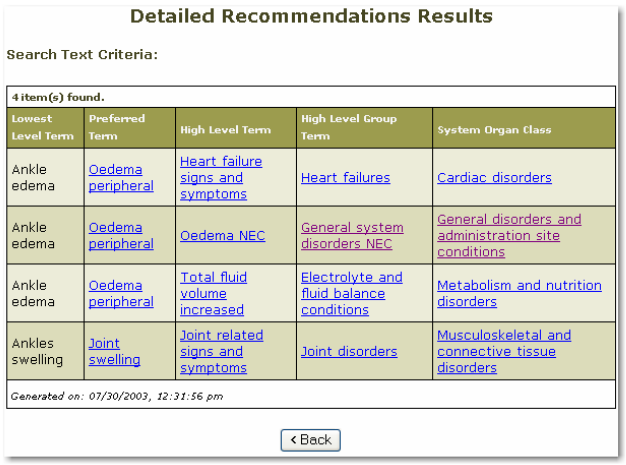
```
start = 1;
letters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ";
do i = 1 to length(verbatim);
   curchar = substr(verbatim,i,1);
   if index(letters,upcase(curchar))
   = 0 or i=length(verbatim) then
   do;
      curword =
      trim(substr(verbatim,start,(i-
      start) + 1));
```

```
        start = i+1;
        ...
```

- Search by Words – Individual words are searched against the specified dictionary for matches. These matches are then used to form the list of recommendations.

The recommendations are generated in a condensed and abbreviated list. This allows the user to quickly identify the correct match and make a selection. There are times, however, when a detailed review is needed.

(LLT, PT, SOC) details...



Detailed Recommendations Results

Search Text Criteria:

4 item(s) found.

Lowest Level Term	Preferred Term	High Level Term	High Level Group Term	System Organ Class
Ankle edema	Oedema peripheral	Heart failure signs and symptoms	Heart failures	Cardiac disorders
Ankle edema	Oedema peripheral	Oedema NEC	General system disorders NEC	General disorders and administration site conditions
Ankle edema	Oedema peripheral	Total fluid volume increased	Electrolyte and fluid balance conditions	Metabolism and nutrition disorders
Ankles swelling	Joint swelling	Joint related signs and symptoms	Joint disorders	Musculoskeletal and connective tissue disorders

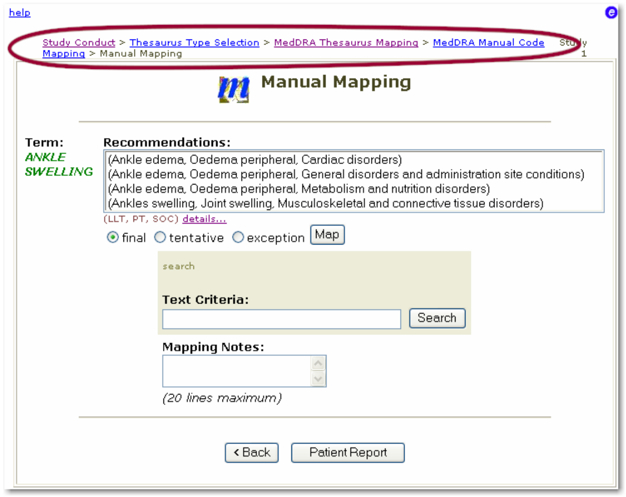
Generated on: 07/30/2003, 12:31:56 pm

‹ Back

For performance reasons, the detailed report is not generated as another page but rather a local link to the same page. Since all the information needed to generate the details are already there, the detailed view is generated further down the same page. Simple local links are very efficient ways of creating interactivity.

The use of hyperlinking is one of the unique characteristics of the web that makes it easy to navigate. When an application has several levels of drill down, it is essential to have a key path displaying where the user is at within the application. The following example shows the path to the manual mapping screen.

Study Conduct > Thesaurus Type Selection > MedDRA Thesaurus Mapping > MedDRA Manual Code Mapping > Manual Mapping



Study Conduct > Thesaurus Type Selection > MedDRA Thesaurus Mapping > MedDRA Manual Code Mapping > Manual Mapping

Manual Mapping

Term: *ANKLE SWELLING*

Recommendations:
(Ankle edema, Oedema peripheral, Cardiac disorders)
(Ankle edema, Oedema peripheral, General disorders and administration site conditions)
(Ankle edema, Oedema peripheral, Metabolism and nutrition disorders)
(Ankles swelling, Joint swelling, Musculoskeletal and connective tissue disorders)

(LLT, PT, SOC) details...

[✓ final] tentative exception

Text Criteria:

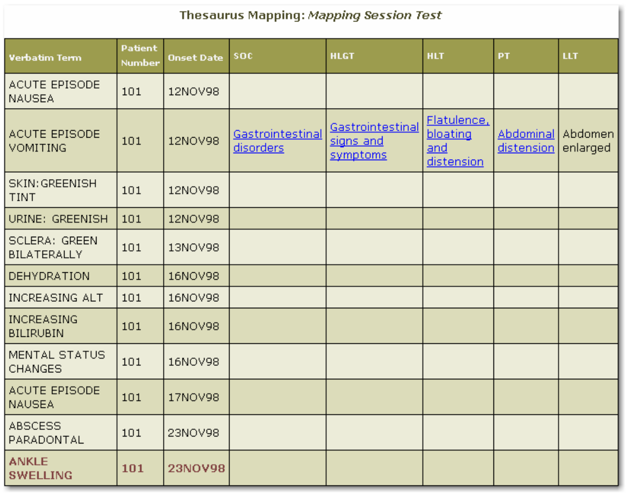
Mapping Notes:

(20 lines maximum)

In this case, the current screen name is listed as the last item on the path. It is not hyperlinked because the user is already there.

This is analogous to leaving bread crumbs on a trail so it marks a path from where you came. The user can navigate to any of the other screens by clicking on one of the above links. This allows you to prevent users from using the browser back button while still having the ability to navigate to their previous screens. It is actually visually clearer than the back button since it shows the path all the way from the top level screens. The hyperlinks create a new page which represents the previous page. It may be a new page, but to the user, it looks like the previous page where they were before.

The use of colors is a very useful form of communicating results to users. You can generate customized reports with the use of HTML coding to highlight important information. During manual coding, the user can select a patient listing report.



Thesaurus Mapping: ***Mapping Session Test***

Verbatim Term	Patient Number	Onset Date	SOC	HLGT	HLT	PT	LLT
ACUTE EPISODE NAUSEA	101	12NOV98					
ACUTE EPISODE VOMITING	101	12NOV98	Gastrointestinal disorders	Gastrointestinal signs and symptoms	Flatulence, bloating and distension	Abdominal distension	Abdomen enlarged
SKIN:GREENISH TINT	101	12NOV98					
URINE: GREENISH	101	12NOV98					
SCLERA: GREEN BILATERALLY	101	13NOV98					
DEHYDRATION	101	16NOV98					
INCREASING ALT	101	16NOV98					
INCREASING BILIRUBIN	101	16NOV98					
MENTAL STATUS CHANGES	101	16NOV98					
ACUTE EPISODE NAUSEA	101	17NOV98					
ABSCESS PARADONTAL	101	23NOV98					
ANKLE SWELLING	**101**	**23NOV98**					

This report uses two different HTML coloring techniques. The first one is to alternate the background color of each row of the report. This is accomplished through customized SAS and HTML coding.

```
*** Set background color ***;
if mod(cnt,2) = 0 then bgcolor = '#DCDCBA';
else bgcolor = '#ECECD9';
```

The CNT is the counter variable representing each row of the report. The variable BGCOLOR is used to generate the HTML.

```
<div align="center"><center>
  <table bgColor="#000000" border="0"
  cellPadding="4" cellSpacing="1"
  width="600">
  <tr>
    <td bgColor="&bgcolor">&curfont1
    . . .
```

Another example of this technique is to highlight the term displayed on the report. This is a form of traffic lighting.

```
*** Set color for selected term
appropriately ***;
if curterm = thesterm then do;
   curfont1='<font face="Verdana, Arial"
   size="2" color="#804040"><b>';
   curfont2='</b></font>';
end;
else do;
   curfont1 = '<font face="Verdana, Arial"
   size="2">';
   curfont2='</font>';
end;
```

The HTML code will then utilize the variables defined to change the colors and font.

```
<tr>
  <td bgColor="&bgcolor">&curfont1
  &curterm &curfont2</td>
  <td bgColor="&bgcolor">&curfont1
  &ptval &curfont2</td>
  <td bgColor="&bgcolor">&curfont1
  &curdate &curfont2</td>
```

SAS ODS can generate traffic lighting in conjunction with SAS formats. For many situations, ODS is probably easier and more elegant. However, there are situations when a custom report can deliver more impact.

Creating Mapped Data

The final step in working with MedDRA is to deliver a mapped dataset to the user. The data delivered looks very similar to the original data except for the fact that new columns are added containing the preferred matched terms. This allows users to perform analysis upon the preferred terms to derive summary statistics. This final mapping process is referred to as “mapex” since the new dataset will contain the same name as the original with an additional “mx” added at the end.

A strategy used to manage auto coding, manual coding and mapex is to store mapping session information in a dataset. This stores unique names, thesaurus dictionaries used and related source data.

Variable	Label
THESNAME	Thesaurus Mapping Name
THESTYP1	Thesaurus Type (1)
THESTYP2	Thesaurus Type (2)
THESTYP3	Thesaurus Type (3)
THESTYP4	Thesaurus Type (4)
THESOUT	Thesaurus Destination
DATNAME	Data Source Name
DATPATH	Data Path
VARNAME	Variable Name
PTVAR	Patient Variable Name
DATEVAR	Onset Date Variable Name
USRNAME	User Name
DATETIME	Date Time

The mapping name is a unique identifier. The thesaurus type captures the dictionaries which are used as the source data for the merge. Once mapping decisions are made, these terms are populated to the destination thesaurus. The system uses the thesaurus name in this dataset as a key to get to all the other attributes. There are two additional administrative variables which capture the user name and date time of each transaction. This is a good way of keeping an audit trail of the last user who had set up or modified a particular mapping session. The analysis file or data mart approach to managing mapping information improves efficiencies in performance and operations throughout the process.

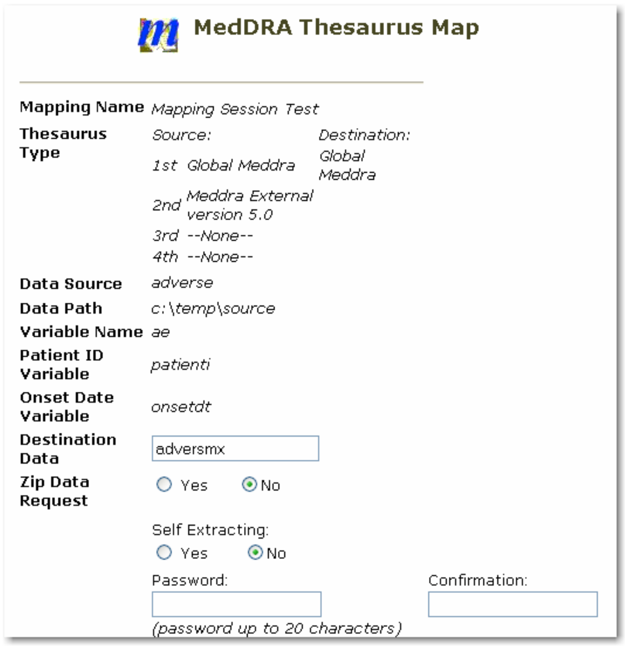
Similar to the auto coding process, the function of the mapex process is encapsulated within a macro. The macro references the thesaurus name to get to all the information it needs. This simplifies the parameters needed.

```
*** Macro to perform "MAPEX" ***;

   *** create new data... ***;
%mend ;
```

The MAPNAME refers to the mapping name and the newly created dataset is referenced through the OUTDATA parameter.

An accompanying web interface is available to perform mapex. The interface captures the values specified from the mapping session and displays them to the user as confirmation.


MedDRA Thesaurus Map

Mapping Name	Mapping Session Test
Thesaurus Type	Source: 1st Global Meddra; 2nd Meddra External version 5.0; 3rd --None--; 4th --None-- — Destination: Global Meddra
Data Source	adverse
Data Path	c:\temp\source
Variable Name	ae
Patient ID Variable	patienti
Onset Date Variable	onsetdt
Destination Data	adversmx
Zip Data Request	Yes [✓ No]
	Self Extracting: Yes [✓ No]
	Password: Confirmation: (password up to 20 characters)

After the user places a request for mapex, the process takes a few seconds. Thesaurex initiates a call to the macro and runs that job asynchronously. This allows users to do other tasks while the new data is being created. An email is sent to the user upon completion.

Asynchronous processing presents opportunities to perform other tasks since the user does not have to wait for the results. One example is to create a compressed zip file of the output data upon completion. This allows the resulting zipped file to be a smaller attached file so that the email is small. The confirmation email can therefore be used as more than just a status update, but also as an information delivery tool.

The compressed file also has the ability to be password protected. This gives the user an extra layer of security if the dataset were to be sent outside the organization. In the mapex example, the password can be entered as any text. In another similar example, the file transfer utility randomly generates the default passwords. In this example, the password must include special characters along with letters and numbers. The algorithm was devised to consistently accomplish this requirement.

```
*** Handle initial password ***;
if zpasswd = '' then do;
   *** Generate a random password ***;
   chars
="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNO
PQRSTUVWXYZ0123456789";
   special = "!@#$%^*()_+-={}[]\:;<>,.?/";
   allchars = chars || special;
   seed = round(datetime());
   curpass = "";
```

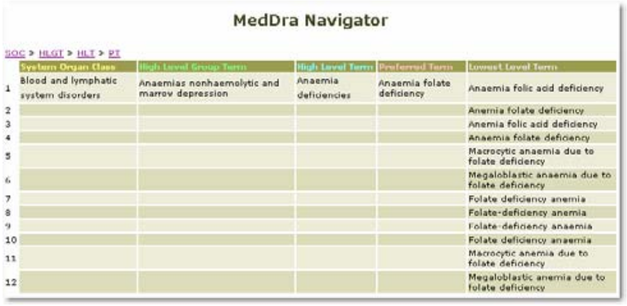
```
    do i = 1 to 7;
       rand = abs(round(normal(seed +
       i)*44+i));
       if rand > 62 then rand = i*8;
       curlet = substr(allchars,rand,1);
       curpass = curpass || curlet;
    end;
    zpasswd = curpass;
    vpass = curpass;
end;
```

During password entry, there is usually one entry for the password and a second to confirm the entry. The reason is that when the user types the password, it is not shown on the screen. The confirmation password is also tracked and pre-populated through a variable named VPASS.

Hyperlinked Dictionary View

The MedDRA thesaurus dictionary delivered from MSSO comes as a series of ASCII files. There are instructions describing how they are normalized tables with key fields used for merging. Once merged, it forms a database with a sophisticated five level hierarchy as described at the beginning of this paper. MSSO also delivers a MedDRA browser which allows you to search for a particular term. One of the challenges confronted by novice users is that once a term is identified, how does it then fit into the hierarchy. For example, for a specific preferred term, what are its related lowest level terms and highest level terms? It is useful to see how a particular term links all the way back to the System Organ Class. If you can visualize this as branches on a tree, it would be nice to have a view that visually allows you to navigate from any "leaf" back to the root of the tree. This is what the MedDRA Navigator accomplishes.

The MedDRA Navigator is a series of HTML pages that display the MedDRA terms in their hierarchy with the ability to navigate up and down the hierarchy.



MedDra Navigator

SOC > HLGT > HLT > PT

	System Organ Class	High Level Group Term	High Level Term	Preferred Term	Lowest Level Term
1	Blood and lymphatic system disorders	Anaemias nonhaemolytic and marrow depression	Anaemia deficiencies	Anaemia folate deficiency	Anaemia folic acid deficiency
2					Anemia folate deficiency
3					Anemia folic acid deficiency
4					Anaemia folate deficiency
5					Macrocytic anaemia due to folate deficiency
6					Megaloblastic anaemia due to folate deficiency
7					Folate deficiency anemia
8					Folate-deficiency anemia
9					Folate-deficiency anaemia
10					Folate deficiency anaemia
11					Macrocytic anemia due to folate deficiency
12					Megaloblastic anemia due to folate deficiency

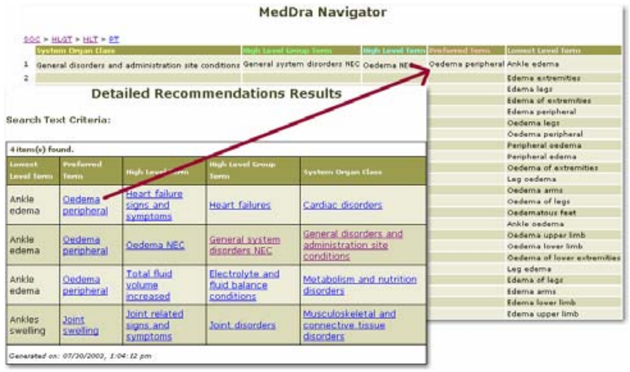
The pages are created in batch mode via SAS programs so that they reside on the web server as static HTML pages. When the MedDRA dictionary is updated, the series of pages are re-produced to reflect the most up to date information. Even though the pages are static, users interact with them in a very interactive way. You can see where you are at within the hierarchy by a little navigational path at the top.

SOC > HLGT > HLT > PT

These also act as hyperlinks so that you can click on them to navigate up the hierarchy. The lowest level terms in each diagram also act as navigational links if the user wants to drill down the hierarchy specific to that term.

All the hyperlinking is coded in plain HTML without any use of CGI. This means that there is no processing required during navigation. The result is that the user can drill down upon the large hierarchy with very fast response.

Each term in MedDRA contains an associated unique identifier coded value. For example, the SOC term "Blood and lymphatic system disorders" has the code value of "10005329". This code is used in the file name as the URL for MedDRA Navigator, "meddra_10005329.html". If the user were to drill down on one of the high level group terms such as "Haemoglobinopathies", the associated code for this is "10018902". The Navigator combines these two codes to derive at a new file name of "meddra_10005329_10018902.html". This naming convention is used throughout the entire hierarchy. Using a systematic approach allows the Thesaurex System to link to MedDRA Navigator anywhere in the hierarchy if it knows the associated codes. An example of this is links the detailed view of the recommended terms to the navigator during manual coding.



MedDra Navigator

SOC > HLGT > HLT > PT

	System Organ Class	High Level Group Term	High Level Term	Preferred Term	Lowest Level Term
1	General disorders and administration site conditions	General system disorders NEC	Oedema NEC	Oedema peripheral	Ankle edema
2					Edema extremities
					Edema legs
					Edema of extremities
					Edema peripheral
					Oedema legs
					Oedema peripheral
					Peripheral oedema
					Peripheral edema
					Oedema of extremities
					Leg oedema
					Oedema arms
					Oedema of legs
					Oedematous feet
					Ankle oedema
					Oedema upper limb
					Oedema lower limb
					Oedema of lower extremities
					Leg edema
					Edema of legs
					Edema arms
					Edema lower limb
					Edema upper limb

Detailed Recommendations Results

Search Text Criteria:

4 item(s) found.

Lowest Level Term	Preferred Term	High Level Term	High Level Group Term	System Organ Class
Ankle edema	Oedema peripheral	Heart failure signs and symptoms	Heart failures	Cardiac disorders
Ankle edema	Oedema peripheral	Oedema NEC	General system disorders NEC	General disorders and administration site conditions
Ankle edema	Oedema peripheral	Total fluid volume increased	Electrolyte and fluid balance conditions	Metabolism and nutrition disorders
Ankles swelling	Joint swelling	Joint related signs and symptoms	Joint disorders	Musculoskeletal and connective tissue disorders

Generated on: 07/30/2002, 1:04:12 pm

Note that specified terms have the ability to link to the Navigator. The link for the above example looks like:

```
<font face="Verdana, Arial" size="1">
<a href="meddra_10005329_10018902.html"
target="_blank">Haemoglobinopathies</a>
</font>
```

The target is set to "_blank" so that a new window is opened. The detailed report becomes much more interactive giving the user the ability to drill down to each specific term. Once in the Navigator, you can go up and down the hierarchy. This really helps users get a mental handle on this huge hierarchy.

One of the core technical aspects of the Navigator is to use the most fundamental element of HTML. That is, the ability to hyperlink from one page to the next. It relies on a logical yet simple naming convention. From the user's perspective, the experience is enhanced with greater interactivity. However, if you look under the hood, the tool does not use any fancy technologies but goes back to the basics to deliver a powerful yet simple solution.

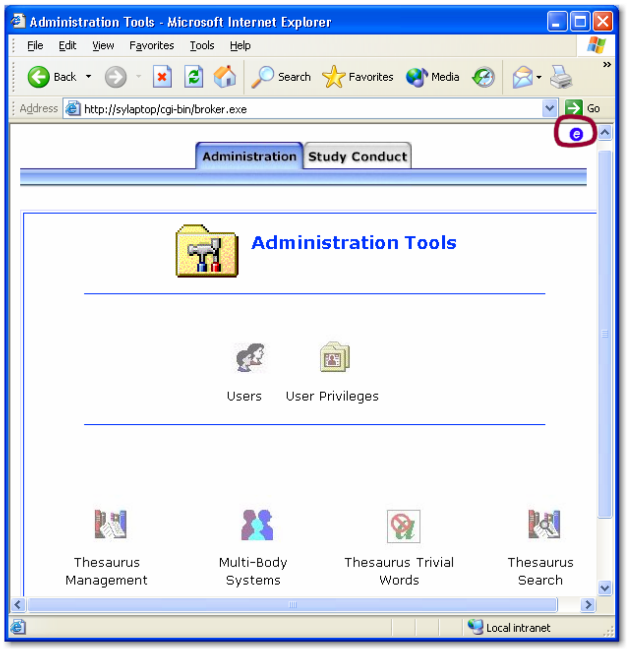
Project Management

There are many challenges in developing software in addition to technology issues. Many times, communication plays a significant role in the success of a project. The content of the communication relates to users' requirements or wish lists. Some are technical while others are just warm and fuzzy. The following are some web-related strategies to optimize the communication to smooth out project management tasks.

- Post User Requirements – Some documents need to be reviewed and updated often. In the case of user requirements or functional specifications, there is some back and forth. The latest version of a document can be posted on a dedicated website on an intranet or extranet so that all the users can view it.
- Links to Application – Since the application is a web application, how does the user get to it? You can send the URL to all users via an email. Once the intranet site

is established, a link from this site to the application is even more effective. This allows users a central place to start.

- Cut and Paste HTML – Since the interface of the application is in HTML format, it is recommended that whenever you communicate to users, cut and paste the screens capturing the HTML code. Most email clients support HTML. A screen shot can be too large if it is a graphic image. The HTML cut and paste equivalent is much smaller and more efficient.
- Discussion Groups – The use of emails is useful but the conversation is not kept historically. The use of electronic discussion groups or bulletin boards is a good way of retaining a knowledge base that is searchable for future reference. The Thesaurex System has a link on the upper right corner of most screens to allow users to ask a question.


Administration Tools - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://sylaptop/cgi-bin/broker.exe Go
Administration Study Conduct
Administration Tools
Users
User Privileges
Thesaurus Management
Multi-Body Systems
Thesaurus Trivial Words
Thesaurus Search
Local intranet

This starts a discussion thread which defaults the subject to the name of the screen. This is useful for technical questions or enhancement requests.

The web related technologies enhance communications. However, there are times when old fashion communications techniques are required. The following examples demonstrate this.

- Document First – It is common that a programmer would work on the program before generating any specifications or user documentation. If you can use the documentation as a communication vehicle, it would greatly increase efficiency. For example, we needed a macro which updates all the mapping sessions in the event that an external dictionary is updated. We called this macro *%mupdate* and documented the reference manual first. This captured all the parameters and described the details of what the macro does.

```
/*-----------------------------------------*
* program:      mupdate.sas

* Description: Update MedDRA external
*              dictionary for all
*              existing mapping sessions.
*
* Parameters:  olddict=old dictionary
*              newdict=new dictionary
* By:          Sy Truong, 6/26/2003
*
*----------------------------------------*/
%macro mupdate (olddict=, newdict=);
   *** Hold off on coding … ***;
```

The user would review the reference documentation as if they were about to use it. Comments and suggestions were made and the documentation was updated. Once this cycle is finished, coding starts. The changes in the documentation were made more concisely and faster compared to changing things with SAS code logic.

- Continuous Communications – The programmer and the user need to meet on a regular basis. The meeting does not need to take a long time or be very often. I found that once a week is good. It is optimal for the meeting to be face to face but once in a while, a Webex or teleconference will also work. This allows users to express feedback and the programmer to give development updates. This keeps an ongoing dialog that is essential in any project.

The technical challenges are often interesting and not too difficult to solve. The issues that need to be resolved with users can take longer and sometimes be more challenging.

Conclusion

MedDRA's sophisticated hierarchal structure presents opportunities for software to help users work more effectively. SAS/IntrNet is able to fully utilize the power of SAS on the backend while presenting a friendly user interface. By using SAS/IntrNet, it made it possible to develop a system that is powerful yet simple to use. There are actually many different languages used to develop the Thesaurex System including: BASE SAS, SCL, SQL, HTML and Java Script. The best language to use is not necessary the most sophisticated or powerful. Sometimes the use of a simple HTML hyperlink is the most elegant and powerful solution. HTML creates an interactive user interface with instant feedback to users. However, there are times when a batch program using a macro call is more effective. These two approaches can be combined in conjunction with email to form an asynchronous interface to "batch" macros. This asynchronous approach actually presents opportunities to add more features to the macro. Involving the user at the very beginning and continuously maintaining the communication is very important to a successful software project. From the software development perspective, user interface is not just a graphical screen, but rather an intimate face to face relationship.

References

MedDRA information
http://www.meddramsso.com

Thesaurex System™
http://www.meta-x.com/thesaurex

About the Author

Sy Truong is a Systems Developer for Meta-Xceed, Inc. They may be contacted at:

Sy Truong
48501 Warm Springs Blvd. Ste 117
Fremont, CA 94539
(510) 713-1686
sy.truong@meta-x.com