



# ETD-MARC/Perl Module Design

Presented by

**Yan Liao (Clara) & Mary Finn**

University Libraries of Virginia Tech

October 20, 2005

# ETD-MARC/Perl Module Design



- **Overview**
- **Creating ETD-MARC/Perl Module (I):**  
General Design Procedure and Preparations
- **Creating ETD-MARC/Perl Module (II):**  
ETD-MARC/Perl Module
- **Limitations**
- **Further Research and Applications**
- **References**

# Overview

---



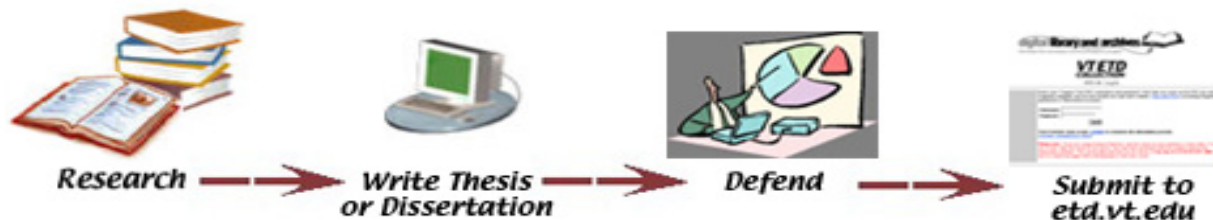
- **Life Cycle of ETD**
- **ETD Cataloging Before Perl Module**
- **ETD Cataloging After Perl Module**

# Overview



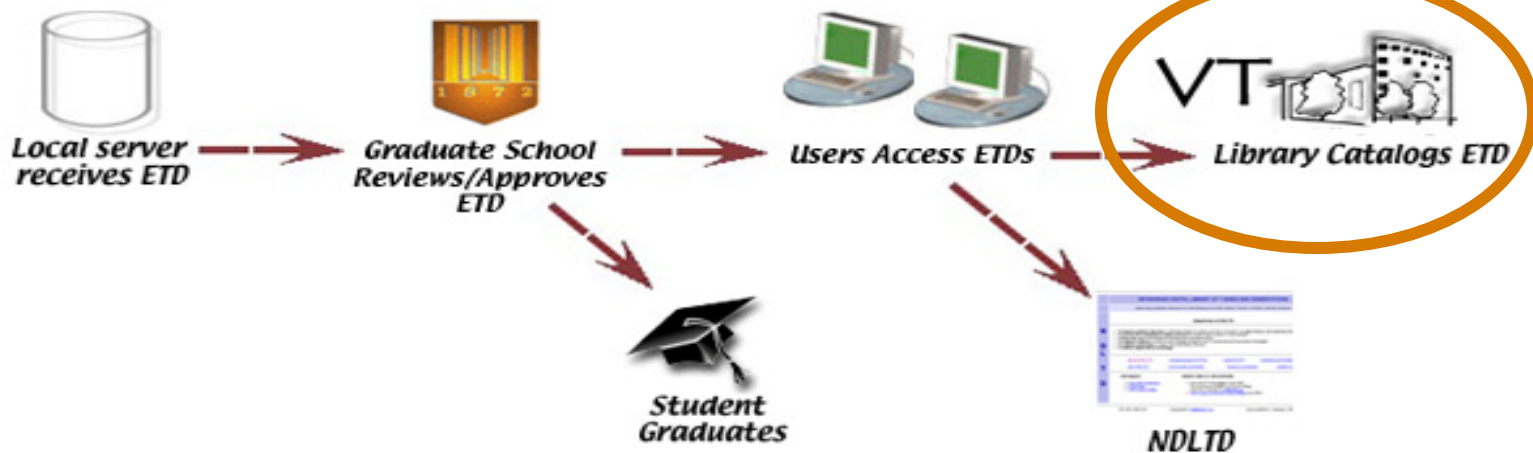
## Life Cycle of an ETD

### Student thesis work



ETD Cataloging

### Steps after submitting an ETD



# ETD Cataloging Before Perl Module



- Professional catalogers developed policies and procedures
- Cataloging from submission form
- Create new records in OCLC
- Copy/Paste data to MARC fields (5 minutes)
- Authority control
- Download to Addison



# ETD Cataloging After Perl Module

- Cataloging from submission forms
- Generating MARC records automatically
- Uploading the records to OCLC
- Authority control
- Download to Addison

# Creating ETD-MARC/Perl Module (I)



- **General Design Procedure**
- **Preparations:**
  - ❖ **ETD metadata meets MARC (leader, fixed field, etc.)**
  - ❖ **ETD-MARC/Perl module Functions**
  - ❖ **Perl**
  - ❖ **MARC::`Record` package**

# General Design Procedure



- Catalogers cooperated with system staff
- Match ETD Metadata with MARC
- Design working module
- Test and use the module



# ETD Metadata meets MARC



- Set up templates for constant data, such as leaders, some fixed fields (006, 007, 008), and variable data fields.
- Map ETD variables to MARC tags

# MARC Coding for Leader

---



## Leader

### Position

00-04: Logical record length  
05: Record status  
06: Type of record  
07: Bib level  
08: Type of control  
09: Character coding scheme  
10: Indicator count  
11: Subfield code count  
12-16: Base address of data  
17: Encoding level  
18: Cataloging form  
19: Linked record requirement  
20: Length of the length-of-field portion  
21: Length of the starting-character-position portion  
22: Length of the implementation-defined portion  
23: Undefined

### Code

n: New  
a: Language material  
m: Monograph/item  
\_: No specific type of control  
  : MARC - 8  
  
K: Less than full level  
a: AACR2

# MARC Coding for 006

---



## 006 Field

### Position

00: Form of material

01-04: Undefined

05: Target audience

06-08: Undefined

09: Type of computer file

10: Undefined

11: Govt. publication

12-17: Undefined

### Code

m: Computer file

d: Document

s: State, provincial, etc.

# MARC Coding for 007

---



## 007 Field

### Position

00: Category of material  
01: Specific material designation  
02: Undefined  
03: Color  
04: Dimension  
05: Sound on medium  
06-08: Image bit depth  
09: File format  
10: Quality assurance targets  
11: Antecedent/source  
12: Level of compression  
13: Reformatting quality

### Code

c: Computer file  
r: Remote  
  
u: Unknown  
n: Not applicable  
u: Unknown

# MARC Coding for 008

---



00-05: Date entered on file	
06: Publication status	s: single known date/probable date
07-10: Date 1 year	
11-14: Date 2	
15-17: Place of publication	vau: Virginia
18-21: Illustration 1-4	a: Illustrations
22: Target audience	
23: Form of item	s: electronic resource
24: Contents 1	b: bibliographies
25: Contents 2	
26: Contents 3	
27: Contents 4	
28: Govt. publication	s: State, provincial, etc.
29: Conf. Publication	0: not a conference publication
30: Festschrift	0: not a festschrift
31: Index	0: no index
32: Undefined	
33: Literary form	0: not fiction
34: Biography	_ : no biographical material
35-37: Language	eng: english
38: Modified record	
39: Cataloging source	d: other

# Constant data in variable MARC fields



040: Cataloging Source	\$a VPI \$c VPI
245: Title	\$h [electronic resource]
260: Publication	\$a [Blacksburg, Va.] \$b University Libraries, Virginia Polytechnic Institute and State University
500: Notes	\$a Title from electronic submission form
500: Notes	\$a Vita
504: Bibliographies	\$a Includes bibliographical references
538: System requirements	\$a System requirements: World Wide Web browser and PDF reader.

# MAP ETD Variables to MARC



<b>ETD variable name</b>	<b>MARC tag(s)</b>
urn: Universal Resource Name	035
year: release year	008, 099, 260, 440, 502
type: text Document type	099
title: Title of document	245
first_name: First name of author	100, 245
middle_name: Middle name of author	100, 245
last_name: Last name of author	100, 245
comp_file: computer file characteristics	256
degree: degree(M.S., M.A., Ph.D. etc.)	440, 502
abstract: abstract of document	520
url: URL of ETD	856
keywords	653
department:	440

# ETD-MARC Perl Module Functions



- **Queries the ETD database to extract Metadata**
  - ❖ connect ETD database
  - ❖ fetch metadata from database

→ **Perl Script**
- **Creates a MARC record for each ETD**
  - ❖ placing the appropriate metadata in the appropriate MARC tags

→ **MARC::Record Framework (MARC.pm)**



# Perl

---



- **Practical Extraction and Report Language** : a general-purpose programming language invented in 1987 by Larry Wall
- **Facilities**: text processing, database accessing, networking, etc.
- **Perl Module**: a set of related functions that are packaged together into a library file that has an extension of “.pm”, such as CGI.pm, MARC.pm
- **CPAN**: Comprehensive Perl Archive Network ([www.cpan.org](http://www.cpan.org))

# MARC::Record Package



- MARC.pm
  - ❖ a piece of open source software developed by librarians for librarians in the summer of 1999
  - ❖ contains functions to read in USMARC data; to add, remove, and modify fields; to search through data; to save MARC data
- MARC::Record (1.39)
  - ❖ Latest and enhanced version of MARC.pm
  - ❖ contains: MARC::Batch; MARC::Field; MARC::Record; MARC::File; MARC::Lint (separate package since Dec. 2004)

# Creating ETD-MARC/Perl Module (II)



- **ETD-MARC Module**

- ❖ **Core part**
- ❖ **Crucial subfunction: Encode\_USMARC**
- ❖ **Cataloging problems and Perl Script**
  - **Title problem: 245 indicators**
  - **Author name problem: 100 and 245**

# ETD-MARC Module: core part



```
our $dbh = db_connect (); → Connect ETD database  
my @urns = fetch_urns ($limit); → Fetch data  
for each my $urn (@urns) {  
  my %record = fetch_record($urn); → Fetch data  
  my $marc = encode_usmarc(\%record); → Generate MARC record  
  print OUT $marc;  
}  
$dbh->disconnect(); → Disconnect ETD database
```



## ETD-MARC Module: encode\_usmarc(I)

- Assign data value to subfunction variables, for example:
  - ❖ `my $url = $record{main}->{url};`
- Create MARC record, for example:
  - ❖ `$marc->append_fields(MARC::Field->new('856', '4', '0', u=>"$url"),);`



# ETD-MARC Module: encode\_usmarc(II)

- **MARC::Record** : primary class represents a MARC record, being a container for multiple MARC::Field objects
  - ❖ New() : `my $marc = MARC::Record->new();`
  - ❖ Leader(): `$marc->leader(' nam 2200000Ka 4500'),`
  - ❖ Append\_fields(): `$marc->append_fields(MARC::Field->new('006', 'm d s '),);`
  - ❖ As\_formated(): `return $marc->asformated();`
  - ❖ As\_usmarc(): `return $marc->as_usmarc();`
- **MARC::Field**: object for representing the indicators and subfields of a single MARC field
  - ❖ new(): `MARC::Field->new('040', "", "", a=> 'VPI', c=> 'VPI'),`

# 245 indicator problem

---



```
My @mTitle;
If ($title = ~ /A \b*/){
    @mTitle=('245', '1', '2', a=>"$title");
}
elseif ($title=~/An \b*/){
    @mTitle=('245', '1', '3', a=>"$title");
}
elseif ($title=~/The \b*/){
    @mTitle=('245', '1', '4', a=>"$title");
}
else {
    @mTitle=('245', '1', '0', a=>"$title");
}
```

# 100, 245 Author Name Problem (I)



- **Middle Name, Extra Space, and Punctuation Problem**
  - ❖ 100: \$a Last\_Name, First\_Name Middle\_Name.
  - ❖ 245: \$c First\_Name Middle\_Name Last\_Name.
  - **If Middle Name = Null:**
    - ❖ 100: \$a Last\_Name, First\_Name \_.
    - ❖ 245: \$c First\_Name \_ Last\_Name.
  - **If Middle Name with period:**
    - ❖ 100: \$a Last\_Name, First\_Name Middle\_Name..





# 100, 245 Author Name Problem (II)

## ● Check Name String Code

```
if ($mName){  
  $hName = "$IName, $fName $mName";  
  $cName = "$fName $mName $IName.";  
  unless ($hName = ~ / \b\.$/){  
    $hName = "IName, fName $mName.";  
  }  
}  
else{  
  $hName = "$IName, $fName";  
  $cName = "$fname $IName.";  
  unless ($hName = ~ /\b\.$/){  
    $hName = "IName, $fName.";  
  }  
} ...  
MARC:: Field->new('100', '1', '', a=> "$hName"),  
MARC:: Field->new(@mTitle, h=> "[electronic resource]V", c=> "$cName"),
```

→ If there is a middle name

↘ → Include middle name

→ If 100 doesn't end with "."

→ If there is no middle name

↘ → Exclude middle name

→ If 100 doesn't end with "."

# Limitations

---



- limited by the quality of the metadata input by students
- limited to descriptive metadata only, cannot accommodate classification, subject analysis, and name authority validation
- AACR2, 9.7B22 “For remote access resources, always give the date on which the resource was viewed for description”

# VT System Limitations (2004)



- System problems: → Solutions: New System (Connexion & iii)
  - ❖ Systems (2004): OCLC Passport & VTLS
  - ❖ Input problems: Character sets; long abstract
  - ❖ Workflow problems: uploading local files to OCLC
- ETD Database problems: → Solutions: System maintenance
  - ❖ Degree: 440-> “VPI & SU. \$department. \$degree \$year”  
502-> “Thesis (\$degree)\-Virginia Polytech  
Institute and State University, \$year.”
  - ❖ No index table in the database for the degree types,  
e.g. MA; M.A. ; Master of Arts;

# Future Research and Applications



- Conduct further research to determine if there are savings in terms of human resources and time to catalog
- Applications of Perl script on other digital cataloging projects:
  - ❖ import MARC data into a relational database
  - ❖ perform metadata auto-crosswalk
  - ❖ manipulate vendors' digital bibliographic records
  - ❖ ...

# References

---



- Brian E.Surratt and Dustin Hill, “ETD2MARC: a semi-automated workflow for cataloging electronic theses and dissertations”, Texas A&M University, 2004, <http://di.tamu.edu/bsurratt/>
- Anne Highsmith ...[et al.], “MARC it your way: MARC.pm”, *Information Technology and Libraries*, vol. 21, no.1, March 2002.
- MARC/Perl : <http://marcpm.sourceforge.net/>
- Comprehensive Perl Archive Network(CPAN): <http://www.cpan.org/>
- Perl4Lib: <http://perl4lib.perl.org/>
- VT Electronic Theses and Dissertations: Cataloging Instructions: <http://techserv.lib.vt.edu/Cataloging/CTetd.htm>
- VALET for ETDs : <http://www.vtls.com/Products/valet-for-ETDs.shtml>

# Thank You!

Contact: Yan Liao (Clara)

Phone: (540) 231-8845

Email: [liaocy@vt.edu](mailto:liaocy@vt.edu)

Mary Finn

(540) 231-4980

[maryfinn@vt.edu](mailto:maryfinn@vt.edu)

