Weeding Fees
IGeLU 2010

Eevaliisa Colb
Toni Vallden
Helsinki University Library
Finland
Introduction

- HELKA library system
- Problem
- Solution
- Production Batch Runs
- State of affairs after
HELKA Library System

• Voyager 6.2.1 at the time (now 7.2.1)
• Helsinki University Library, 20 locations
• Research collection of National Library of Finland
• 5 non-university special libraries
Fees start to expire

• 2004: Act on Time-Barring of Debt
  - Debt expires in 3 yrs
  - unless reminded
  - After reminding a new 3 yr expiration period
  - Unclear legal status of library fees

• 2009 resolution: library fees expire accordingly
The Problem

• Helka 2009 after the resolution
  • Ca. 11 years worth of expired fees (from 1996!)
  • More fees were expiring every day
  • Confusion at customer service

• How to solve
  • Expired fees forgiven
  • Patrons with unpaid non-expired reminded about their fees to restart the 3 yr count-down to the expiration
Forgiving expired fees

• Voyager 6.2.1
  • Circjobs 40, 41 not available to forgive fees by patron id or create date

• Used bursar transfer job, Pbursar
  • Not run for its intended use in HELKA
  • Settings: all locations, all patron groups, interval of 1095 (3x365) days, operator id EXPIREDFEE

• Autohotkey macros
  • Pbursar cannot forgive fees if patron has no active barcode or institution id
Patrons included

• After some consideration, the criteria to send a reminding letter to a patron was defined
  • Patron has unpaid fees which are
    • Less than 3 years old
    • More than 1 year old
    • Totaling over 1 euro
  • For these patrons, all fines included
    • Also those less than 1 year old
Statement generation

- A computer program HEMULI was developed (Python 2.4.4) to produce fee statement files
  - As e-mail or
  - In a specific electronic format for paper letters
- Text templates in Finnish, Swedish, English
- Not used: Voyager's fee statement reporting functionality (circjob 14 and Reporter)
  - Complexity of the fee/patron selection criteria
  - Required format of the output
Sending of statements

- HEMULI Statement files were manually transferred to another server already set up for sending daily library notices
- Modified the existing system to also handle the expired-fee-notices
- Statements sent
  - As e-mail notice if possible
  - As a paper letter (iPost: transfer data in a certain electronic format to mail service, where letters were printed, enveloped, distributed)
- Bounced mail sent again later
Statement batches

- Statements generated and sent in batches (A, B, C, …)
- Unique statement id for each letter
  - For controlling re-sending if patron not reached
  - Id was included in the email subject field and the envelope window of the paper letters
  - file for statement ids -> patron id lookups maintained by HEMULI
Batch A: e-mails

- All eligible patrons with e-mail address
  - University e-mail addresses checked against University LDAP service. If no match, e-mail address was deleted from Voyager patron record (->notice sent in batch B as a paper letter instead)

- Bounced email statements
  - Sorted by library staff into subfolders
  - Subfolder for unknown recipients scanned with a Python script to extract the statement ids from the subject headers to be included in a re-send batch
  - Invalid email address was deleted from Voyager patron record before re-send
Batch B: Paper letters

- Eligible patrons with no email
- Bounces
  - A librarian looked for the patron's current address (student register, district registry office...)
  - cost-benefit evaluation: some fees forgiven
  - If found
    - Updated Voyager patron record
    - Added statement id to a re-send list for next batch
    - No need to open the letter, as statement id visible on the envelope
Batches C-F: re-sends

- Included
  - Bounced emails from batch A
  - Return-to-sender letters from other previous batches

- Generated by HEMULI, parameters:
  - Re-send list of statement ids
  - Statement id -> patron id lookup file

- Easier to generate new HEMULI-batches than to re-envelope/re-address the returned mail
# Summary of notices

<table>
<thead>
<tr>
<th>BATCH</th>
<th>STATEMENTS SENT</th>
<th>DATE</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7156</td>
<td>20.4.2010</td>
<td>Sent as emails</td>
</tr>
<tr>
<td>B</td>
<td>2518</td>
<td>20.4.2010</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>685</td>
<td>22.4.2010</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>180</td>
<td>26.4.2010</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>101</td>
<td>30.4.2010</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>78</td>
<td>12.5.2010</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- HELKA – a large and complex Voyager implementation in Finland
- Legally expired fees removed from the system
- Annual notices to avoid further credit loss
Thank you! Questions?

eevaliisa.colb@helsinki.fi

toni.vallden@helsinki.fi