Erasmus MC
Universitair Medisch Centrum Rotterdam

Generation R

Pascal@Scheffers.Net

Generation R – Project Overview

Prospective population study

We track 10.000 children from 12th week pregnancy and their mothers and fathers

Planned duration is about 20 years

Five university departments

About 25 external organisations (midwives, hospitals, laboratory, local government)

Lots of PhDs

Lots of staff

Many cultures, both respondents and participants

Technology

Tcl and a bit of Tk

Aolserver with OpenACS 3 for logistics and datamanagement

Postgresql, Sybase and Microsoft Access

LaTeX

Regsub, diff and patch

Forms5 ‘Electronic Datacapture’

Health Level 7

SPSS / SAS

Tcl/Tk Stuff

- Logistics system AOLServer/OpenACS
- Database links:
    - Postgresql/Sybase for Registration and Logistics
    - Forms5 (CSV-like) to Postgresql for EDC
    - Proxy server to access Sybase over VPN
    - HL7 to Sybase for demographics and appointments
    - MS Access to batch web requests for birth registration
    - Access to Postgresql for additional logistics
- Automated email data transfer with PGP encryption/authentication

Erasmus MC

Data gathering with Questionnaires

- Expect to issue and receive 90.000 QA’s in the first three years
- That is roughly 1.25 million pages
- Must be suitable for electronic data capture
- QA serial number on each page
- Barcode on the front page
- Accidents
- Must cost less than 10 euros a piece for the entire life

Erasmus MC

Questionnaire Options

Started in Word

- mail merges too slow
- printing too slow
- printing problems, unpredictable things in the output

Try LaTeX?

- Mail merges very fast
- Questionnaire generation can be scripted
- Templates hard to define
- Not enough time to do it

Erasmus MC

Pagemaker

- mail merges even slower than Word
- printing not much faster
- Few printing problems, excellent positioning
- Usable PostScript output

Regsub, diff and patch

1. Use regsub and diff to figure out where the numbers are:

```
set fp [open $document.ps r]
set ps1data [read $fp]
close $fp
set ps2number [string range "qwertyuiopasdfghjklzxcvbnm" 1 [string length
        $ps1number]]
set cnt [regsub -all $ps1number $ps1data $ps2number ps2data]
if { $cnt != $replace_count} {
        error "I replaced $cnt occurences of '$ps1nummer', not the suggested
                $replace_count times."
}
set fp [open document.ps2 w]
puts -nonewline $fp $ps2data
close $fp
```

Erasmus MC

diff

2. Store the diff file

```
catch {
  exec diff $document.ps  $document.ps2" > /tmp/$document.diff
} res

set fp [open /tmp/$document_id.diff r]
set diff [read $fp]
close $fp
ns_db dml $db “update psdocuments set diff=[ns_dbquotevalue
$diff] where document=$document”
```

Erasmus MC

Regsub, patch and lp

3. Regsub the diff file
4. Patch the postscript file
5. Send patched file to printer

```
regsub -all $replace $diff [rt_scan_armor_number $id] onediff
exec patch $tmp_dir/$document_id.ps << $onediff
exec lp $tmp_dir/$document_id.ps
```

And out comes a numbered questionnaire.

Health Level 7

- ISO ‘Application’ level 7 data exchange
- Exchange of patient demographics, status and medical information
- Standard set by a commercial company www.hl7.com
- Nicer than EDI, but not by much
- The ‘client’ is a TCP/IP server application

Goals:

- To capture demographics and appointment information, no more double entry and loss of synchronisation
- To transfer from the offsite database to Generation R logistics system

Erasmus MC

HL7 – TCP/IP Protocol

‘client’ listens for a connection on a TCP/IP port

In:

\xb<Message>\x1c\xd

Out:

\xb<Ack-Message>\x1c\xd

HL7 – Receive Data

```
proc processData { channel } {
        append ::data($channel) [read $channel]
        set begin [string first \xb $::data($channel)]
        set end [string first \x1c $::data($channel)]
          if { $end > $begin } {
            set message [string range $::data($channel) [expr $begin +1] \
                        [expr $end-1] ]
            parseMessage $message
            set ::data($channel) [string range $::data($channel) \
                        [expr $end + 1] end]
          }
```

Erasmus MC

HL7 – Message Format / Sample

```
MSH|^~\&|DISPADT||ASTRAIA|astraia|20030514113000||SIU^S12|59303016762|P|2.2||
SCH|123599^PAS Modelplanning|123599^PAS Modelplanning||||N||||
|^^^200306021520^200306021550|03035032^G.C.H.M.VD AA,VERVALLEN
!||||LDW||||LDW|||||PENDING
PID|1||00000310660||SCHEFFERS^""^P.R.^""^""^""||19742612|F|||Somaereenstraet^83b^
ROTTERDAM^^3123AB^""||06254646311|||""|""|||||""|""
ZPI|1|||""^^""^""^""||||||""
PV1|1|O||
RGS|1|U
AIS|1||GR02^20 WEKEN ECHO GENERATION R^L|200306021520|||30|min||PENDING
AIL|1|U|^ARECH2||0
```

HL7 - Acknowledge

```
MSH|^~\&|ASTRAIA|astraia|DISPADT||||ACK||P|2.2||
MSA|AA|59303016762||
```

HL7 Parser

Notice 6 levels:

- The ‘line’ or *segment* level
- The main segment *fields* separated by |
- Sub, sub-sub,… separated by ^, ~, \ and &

Each segment starts with a segment ID

“” in a field indicates an empty string

Only line, field and ‘^’ subfield used here

HL7 Field map

```
MSH {
    set fields {
        1 encodingchars
        2 sendingapplication
        3 sendingfacility
          …
        8 {type eventtype}
        9 msgid
        11 version
        12 sequencenumber
    }
}
SCH {
    set fields {
        1 {id application}
        16 {fillercode fillername}
        20 {enteredbycode enteredbyname}
          …
```

HL7 Set variables

```
proc hl7_set_segment_variables { segment fields } {
  foreach {fieldno names} $fields {
        if {[llength $names] == 1} {
            upvar ${seg}_$names var
            set var [hl7string [lindex $segment $fieldno]]
        } else {
            set i 0
            foreach name [split [lindex $segment $fieldno] ^] {
                        upvar ${seg}_$name var
                        set var [hl7string [lindex $subfields $i]]
                        incr i
            }
        }
  }
```

Erasmus MC

HL7 – Things Glossed Over

Insert appointments in database

Some triggers to maintain links between keys

Things to Solve

Data entry

- Data-bound widgets
- Combined with PMG?

Database Access - Cross platform issues

- Windows ODBC works
- Unix ODBC sucks
- ODBC: No stubs
- “Native” drivers are hard to come by
- JDBC?
- Tcl Remote ‘TDBC’ system?