Make the Past serve your Future

How Cincom frameworks and tools let you add new functionality to complex legacy applications

Mark Grinnell, Andreas Hiltner,
Niall Ross, Dirk Verleysen

World Headquarters
Cincinnati, Ohio

SIMPLIFICATION THROUGH INNOVATION™

The Problem

- Change vs. Legacy
- Start with legacy systems – several of them
  - Custom written and poorly documented
  - Design knowledge is gone
- Add new requirements or regulations
  - Old systems are incompatible
  - But still must keep doing what they're doing
- Smalltalk to the rescue

Example: blood cancer diagnosis

- Genetics Lab: regional (central lab for many hospitals)
  - Cytogenetics (chromosomes) and Molecular DNA analysis
- Immunophenotyping: also central lab for many hospitals
  - cell surface markers and some molecular DNA analysis
- Hematology lab: one in each hospital
  - full blood count and morphology on sample
- Histo-Pathology lab: one per hospital + central re-referral
  - morphology in situ (of bone marrow trephine or other biopsy)
  - immuno-histochemistry: cell surface markers in situ

Lab Legacy Systems

- Each lab with its own un-retirable system
  - Custom front end
  - Custom relational database schema
  - Not well understood by developers
    - “Every time we make a change it breaks!”

Example Storyboard - Today

- Path lab receiving: print labels, send samples to all labs.
- Wasting time in the Genetics lab: "Probably just iron deficiency anemia – but by the time I phone the Blood lab five times, *maybe* work out what the patient's ID is on *their* system and get their result, it will have been faster just to do our test."
- Life-critical in the Genetics lab: "This bloody sample is no use. It was sent as 'probably merely ITP' - *but* the consultant has just called – Blood lab now say it may be acute lymphoblastic leukemia. *Quick*: call round to find a lab with good sample."

An episode of House? No, a normal day in the NHS!

Example Storyboard – better if ...

Path lab receiving: enter initial diagnosis in web app - prints labels, alerts labs that samples are due, so they can prepare.

Not wasting time in the Genetics lab: "That sample arrived today, but I checked the web app. Blood lab results are already in – it's just iron deficiency. We can skip this one."

Saving lives in the Genetics lab: "Blood lab results are in on the web app - it could be acute lymphoblastic leukemia. Our sample is contaminated, but the app shows analysis begun in Immunophenotyping and their sample is good. Get some fast – we're already 4 days into the 7 day window."

All you need is a web app – that integrates with the lab apps!

So we need a web app – that's easy, right?

- "In legacy IT, everything is very simple, but the simplest thing is difficult" (Clausewitz, *On War, paraphrased* ☺)
- 'Just' map legacy schemas to each other and to new one
  - little/no schema documentation
  - old sub-contracted work (Foreign keys? Try foreign table names!)
- 'Just' make new model and schema natural for domain
  - areas of mismatch to the legacy
- 'Just' do it all yesterday *and* flexibly
  - legacy has enough "one change and it breaks" code already

Is it possible?

- Read old schema
- Generate and map to an Object model
- Evolve the Object model
  - Rename classes to make them make sense
  - Accommodate multiple legacy DBs
  - Specialize classes – make the Smalltalk domain fit the business process
  - Keep mapping up to date as we evolve
- Create common web app DB from refactored mapping

1) ObjectStudio Mapping Tool

- Connect to Genetics Lab legacy DB
  - generate class model from schema (active record pattern in Glorp)
  - Generate a Glorp descriptor system that maps between model and schema
    - tables, columns, foreign keys map to classes, instvars, relationships

Legacy app schema problems

- Schema says Patients go to ‘Rassites’, it should be Hospitals!
- Patients are seen by ‘RasClientContacts’, should be Consultants!
- Patients provide ‘Folders’!
- And all this is before we look at schema relationships.

Rename Classes

- Simple refactor of
  - Folders to Samples
  - RasClientContacts to Consultants
- Write DescriptorSystem
- View current object model in Modeling Tool

Specializing classes

- Consultants still not specific enough
- Split into
  - ReferringConsultants
  - DiagnosingConsultants

Create Descriptor for ReferralDB

- This descriptor will map the same object model to a different database
- Sensible table names

Adding a new inst var to our model

- Add instVar dateOfDispatch to Samples
  - (inverse of dateOfReceipt)
- No more
  - “I could have done it yesterday if I’d known it was coming.”
  - “Where’s the result for X?” “We never received a sample for X!”

Create web app DB

- Generate schema into the new ReferralDB
- Read data from GeneticsDB and write it to ReferralDB

We did it!

Contact info

- Star Team (Smalltalk Strategic Resources)
  - sfortman@cincom.com Smalltalk Director
  - athomas@cincom.com Smalltalk Product Manager
  - jjordan@cincom.com Smalltalk Marketing Manager

- http://www.cincomsmalltalk.com

© 2012 Cincom Systems, Inc.
All Rights Reserved
Developed in the U.S.A.

CINCOM and the Quadrant Logo are registered trademarks of Cincom Systems, Inc.

All other trademarks belong to their respective companies.