Lesson Learned from Integrating OpenClinica with Other IT Systems

Tomas Skripcak
IT scientist at DKTK
DKTK

- German Consortium for Translational Cancer Research
  - Building efficient translational research units focused on cancer research
  - CCP (Clinical Communication Platform) in DKTK handles recruitment of uniquely annotated and stratified cancer patients into translational research projects and trials

- RadiationDosePlan-Image/Biomarker-Outcome platform provides sustainable radiotherapy specific IT infrastructure
  - Large scale clinical trials
  - Collection of imaging data and treatment plans
Why healthcare systems integration

- Create a uniform layer which connect federated clinical and non-clinical IT systems so they can:
  - Leverage each other functions
  - Exchange and use each other data
Integration and interoperability

- Interoperability
  - Syntactic = ability to exchange data
  - Semantic = ability to meaningfully interpret data and use it

- What makes it tick:
  - Protocols (HTTP, …)
  - Services (REST, SOAP, …)
  - Data formats (XML, JSON, …)
  - Terminologies, ontologies, data elements and information models (ICD10, UMLS, CDASH, CDISC ODM)
Common integration artifacts

- SingleSignOn (SSO): unify user credentials
- Enterprise service bus (ESB): unify communication
- Common identifiers (patients, specimens, …)
  - Global IDs
  - ID list services
- Reverse proxy: hide the complexity of underlying infrastructure = one public IP, one domain, one identity
- Common systems in clinical research IT environment:
  - HIS, Pseudonym generator, CTMS & EDC, PACS, LIMS…
OpenClinica integration abilities

- LDAP users
- SOAP web services
- REST
  - Web services
  - RESTful URLs
- eCRFs enhancement (HTML + JavaScript)
- Alternate ways
  - Program web access with HTML document parsing
  - Direct access to OC database
SingleSignOn

- LDAP users feature: make SSO possible BUT!!!
  - LDAP user can not use SOAP web services

- Solution 1:
  - Standard OC user as the primary account
  - OC SOAP study ws (listAll) for authentication or,
  - OAuth OC REST services

- Solution 2:
  - Auto-generated OC user account
  - Navigation to OC from external system (auto-login)
    - POST user credential
OC Web services - SOAP

- Reliable and secure
- Deployed as separate web app
- Limited set of features
- REST seems to be the future direction
- Some features not working e.g.
  - Secondary ID field ignored in subject creation
  - StudySubjectID generation properties ignored
    - Auto StudySubjectID: empty string required
  - Gender property always mandatory
Web services REST

- Originally designed for OC Designer
  - *designer is freshly open sourced :)
- Currently very limited features
- RESTful URLs
  - POST/GET approach (user credential, JSON/XML formatted CDISC ODM clinical data)
  - User credentials in clear text

```python
import requests

session=requests.Session()
loginData={"j_username": "ocuser", "j_password" : "pass"}

r=session.post("http://server/OpenClinica/j_spring_security_check", loginData)

r=session.get("http://server/OpenClinica/rest/clinicaldata/json/view/S_DEFAULTS1/SS_XXY/*//*/includeDNs=y&includeAudits=y")
```
eCRFs enhancement

- JavaScript make it possible to
  - Programmatically work with data from eCRF
  - Trigger external web service/web application

- Make any eCRF data field easily navigable:
  - `RIGHT_ITEM_TEXT (<div id="uniquedentifier">&nbsp;</div>)`
  - `var fieldRow = document.getElementById(divFieldName).parentNode.parentNode;`
  - `var input = fieldRow.getElementsByTagName('input')`

- Domain specific annotations of eCRF fields
  - E.g. in external DB
  - When reliable access to data needed
Alternate ways

- Whatever a user can see a program can see
  - Program access to OC and parsing of HTML DOM tree
  - *not very sustainable (depending on generated HTML)
  - OK for administration usage

- Direct access to OC DB
  - *can be a security issue
  - *can slow down the production DB performance
  - Read only please (bypassing OC application logic)

- Use only as a last resort
Common identifier

- **StudySubject**
  - Tracking patient across studies: Person ID (pseudonym)
  - Merging patient data from another system (e.g. HIS)
    - StudySubject secondary ID
  - Merging patient data from multiple systems
    - PatientID List service (keep the links between IDs separate)

- Saving IDs from different systems in eCRF
  - Make sure data fields are annotated
  - PHI flag if not exportable
Example: RadPlanBio platform

- Sustainable IT solution focused on clinical/preclinical trials
- Radiotherapy specific study features: treatment plans and imaging data
- Multi-centre data exchange and collection (national and international translational projects)
RadPlanBio components

- **Highlighted features:**
  - Virtual server infrastructure with partitioned secured areas for each partner site
  - Imaging data base on DICOM and DICOM RT standards
  - Support for randomisation in clinical trials (Randi2)
  - OpenSource
  - Deployment: web access, hosting, local installation

- **Main components:**
  - EDC & CTMS: OpenClinica
  - Patient identity management service: Mainzelliste
  - PACS server & DICOM viewer: Conquest, DWV
  - Desktop client: DICOM data upload
  - Portal: integration & single access point
Deployment example
Integration portal

- Platform infrastructure database
  - Enable systems communication
- Unify access web access
  - Single URL + SSO
- Integrate RadPlanBio components
  - OC, Conquest, Mainzelleiste
- Extra features
  - Randomisation based on Randi2
Use case: patient registration

- Separate database of patient identities
  - Patient => PID (pseudonym)
  - DB per site

- PID generation
  - 8 character string
  - (read-write fault tolerant)

- Record linkage
  - phonetic code matching
  - configurable for many languages
  - (hear-write fault tolerant)

- Technology: REST Mainzelliste + SOAP OpenClinica
Use case: DICOM data upload

- Desktop client
  - SSO – OC ws SOAP
  - Study/subject/event/item browse
  - DICOM clinical trial deidentification
    - Utilisation of patient PID
    - DICOM supl. 142
  - DICOM ROI harmonisation
    - Standard organ naming
  - Uploading DICOM data
    - Auto import to PACS
  - Import DICOM eCRF

- Technology:
  - REST + SOAP OpenClinica
  - eCRF enhancement
Use case: PACS integration

- Conquest
  - PACS extensions with Lua scripts
    - Querying PACS server
    - JSON formatted DICOM study
    - ZIP and download

- Configurable DICOM viewers
  - Communication over WADO
  - HTML5 = DWV
  - Java = Weasis

- Technology:
  - Lua
  - REST
  - eCRF enhancement
  - WADO
Wish list - now

- Unification of web services strategy
  - REST base
  - Migrate all services from SOAP (study subject, data import)
  - Preferred OAuth 2.0 authentication
  - Side effect: LDAP fully usable

- Direct support for semantic annotation in CDISC ODM
  - E.g. `<Alias Context="UMLS" Name="C1880229"> aka DICOM Study`

- Randomisation as a first class citizen
  - Subject group class
  - E.g. Randi2
Wish list - future

- Direct support for controlled terminology
  - ICD 10, ICD-O-3, ...

- Pluginable architecture for OC
  - Advanced OC modularisation
  - Dynamic loading/unloading of plugins
  - Spring-plugin...

- User changeable localisation

- Reporting
  - Custom reports
Thanks for your attention…

- IT group
  - Tomas Skripcak
  - Uwe Just
  - Marvin Schmidt
- Medical physics group
  - Steffen Loeck
  - Armin Luehr
- Medico-legal group
  - Daniel Buettner
  - Monique Simon
- Leaders
  - Michael Baumann
  - Mechthild Krause
- And the great community around open source clinical IT software