ELAG 2019 Notes

https://bit.ly/ELAG2019

Day 0 - May 7

Bootcamp: Catmandu

Presented by: Patrick Hochstenbach (Universiteit Gent), Johann Rolschewski and Carsten Klee (Berlin State Library)

Catmandu is a command-line tool to retrieve data, perform transformations on it and convert it to another format. The result can be loaded into a data store or directly into an application, saved to hard drive or shown on the screen.

Website: http://librecat.org/

Terminology:

- *Item*: a row, line, MARC record, XML node, …
- *Importers*: Catmandu packages to read items
- *Exporters*: Catmandu packages to export items
- *Fixes*: to transform / manipulate items
  Paths are in dot-notation, *functions* to manipulate, *conditionals* to determine when and *blend* possibilities with other sources (VIAF, AAT, wikidata, geonames, …)
- *Stores*: to save items, make them persistent
  CQL is the preferred query language for Catmandu because this is a language that works against every DB.
  Elasticsearch is a Solr DB without a schema, so it accepts every kind of data.
  Solr uses a schema to which the data must conform, but it is more reliable than Elasticsearch, also for backward compatibility (ES is known to break when a new version is released)
  MongoDB does not use a schema, so it accepts all kinds of data. MongoDB is very good for known and exact searches, for not-known searches regex is used and this is very expensive. In that case it is better to use Elasticsearch or Solr, these are real search engines.
- *Iterables*: every stream of data is an iterator. This way the use of memory stays low.

There is a Virtualbox image for Catmandu available. This way it's very easy to try it: https://librecatproject.wordpress.com/get-catmandu/

- Start the Virtualbox environment
- Start the terminal

- Give a Catmandu command, for example:
  ```
  $ catmandu convert getJSON --from
  'http://lobid.org/organisations/search?q=location.address.
  addressLocality%3ABerlin&format=json' to YAML
  ```

Other examples can be found on the bottom of the above link and try out the advent tutorials
https://librecatproject.wordpress.com/2014/12/08/day-6-introduction-into-catmandu/

Wiki: https://github.com/LibreCat/Catmandu/wiki
Documentation: https://metacpan.org/ and search *catmandu::*
Slides: http://jorol.de/talks/2019-ELAG/#1

To install Catmandu on your own servers: http://librecat.org/Catmandu/#installation

Day 1 - May 8

Opening Ceremony / Hans-Jörg Lieder, Beate Rusch, Peter van Boheemen

- Roughly 150 participants
- 75 / 25% split male to female in participants

Change! Change! Change! On IT-Trends, Innovation and Sustainability / Prof. Thorsten Koch

Intro

- Work as professor in discrete math; works in another department in library work;
- First part of job is primarily in English; second part, in German
- When preparing this talk, slides were split in German / English
- Had to translate a bunch of slides
- AI is topic, kind of follows you wherever you go;
- For 43 years, trying to get computers to do something useful
- If you want to talk about future, start with past; what he is doing today

Definition: congram: defined by the imperfect past the insufficient present and absolutely perfect future

- See the overlap here with IT

1950s:

- “The thing machines were coming” <= topic of time period
- But how do you build those machines?
- LGP 30, historical computer

- Video: the thinking machine. 1950s time. Talking with professor from MIT. Can machines think? Professor answers that 4-5 years ago, he wouldn't think so; now different
- Small group thought computer could do things like play checkers and chess
- Field became known as AI
- 'Machines will think in our lifetime; they won't behave like humans though'
- People hadn't reckoned with ambiguity when building these early computer
- Computers were tied in time period to wanting to follow Soviet Union
- Thought computers would replace translators
- Presenter: Hype in AI now that computers will take over is not new
- Can see translators now online

What does it mean a computer is intelligent

- Standard test today is turing test
- 2 closed rooms; 1 is computer, 1 is person
- Person can't see 2 rooms; asks questions of both
- If person isn't able to distinguish the person from computer, then conclude computer is intelligent
- Intelligence is not quantified here;

Difference between strong and weak AI

- Weak AI: what we currently have. Has 1 particular purpose (play checkers or whatever); write program to solve that problem
- Strong AI: what you see in sci fi movies; we're far from this. Nowhere near having computer able to cope with unexpected things
- Press always conflates this; see weak AI example and concludes strong AI is coming

1970s

- Computers getting better
- Fast algorithms - fast computers
- People started to have great ideas about what you can do with computers
- Computers at time brought computers out of mainframe environment and into homes ; was a kind of revolution
- Computers weren't toys at that time though; very expensive
- No OSS, no internet at time

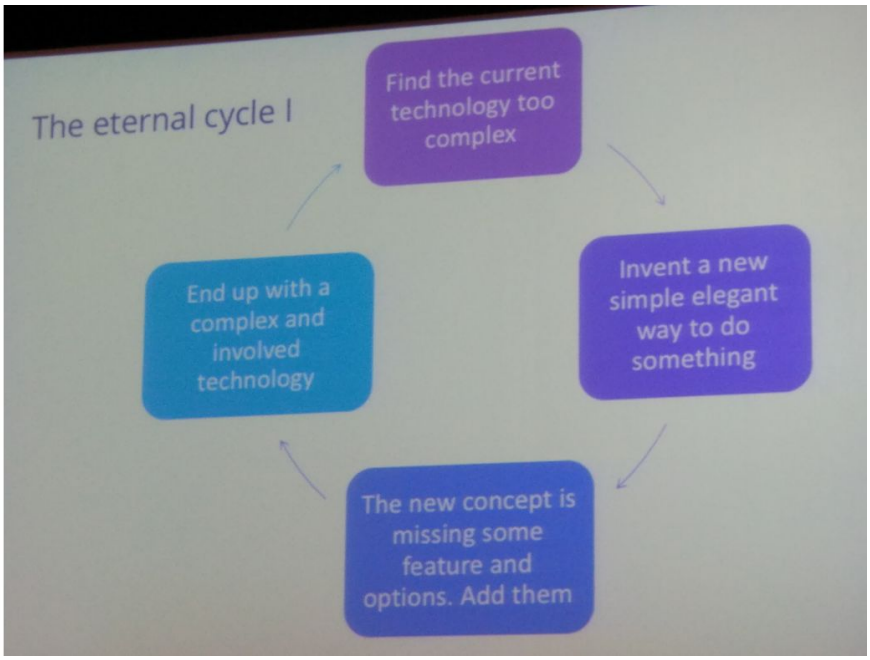
1990s

- Computers commonplace; buy them everywhere
- Once the data is entered, we will get a result
- How do we get data into computer?
- Fill out lots of forms
- No digital map of streets, for example, to help with navigation system programming
- Microprocessors doubled transistor amount in few years (Moore's Law);
- Computers in 1990s getting faster

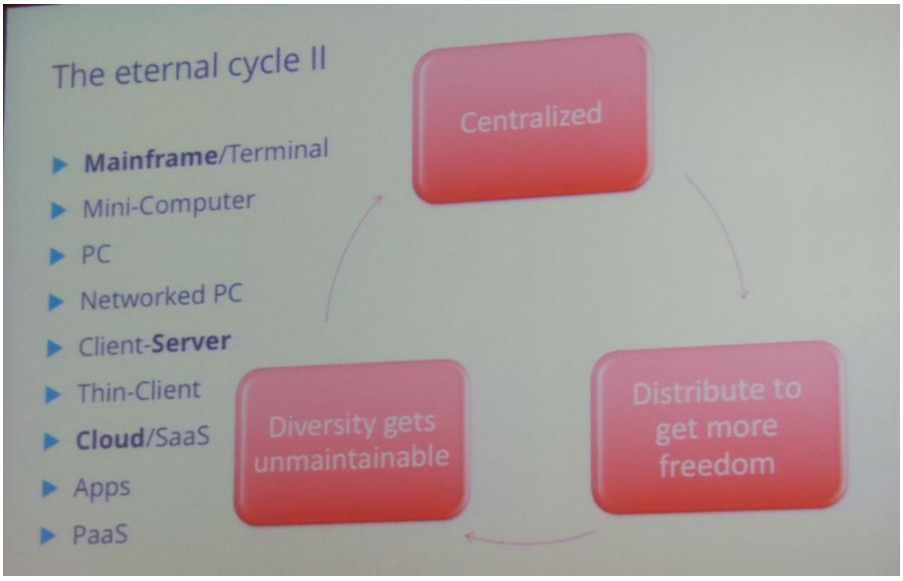
- 2000/2004 weren't getting faster: Hitting a power ceiling with large number of transistors, too much temperature to dissipate
- Direct connection between frequency and processors
- Had to switch in 2000s to increasing number of cores
- Development of internet: from 1990 to 2010, going from very few connected to majority of people connected to internet
- For speaker, time between university and career; remembers university time when you had to get a book, couldn't get information online; now, every topic he presents on has a wikipedia article
- Information is available online; we now have a selection problem
- Recommended book: Mythical man-month / frederick brooks
- About ibm mainframe system implementation in 60s
    - "No Silver Bullet" thesis: 'No single development, either tech or management technique, which by itself promises even one order of magnitude improvement within a decade in productivity, reliability, simplicity
    - No change there now

AI hype today everywhere

- People hope for miracle
- Not likely to happen
- Eternal cycle
    - Find current tech too complicated
    - Invent new simple elegant way to do something
    - Find edge case missing and add feature, options
    - End up with complex and involved technology
    - Repeat
    - PHP (and other languages, e.g. C++, Perl, Python) a good example of this



- Popularity of programming languages
    - pypl.github.io/PYPL.html
    - Main problem, from speakers PoV, lot of people learn 1 programming language
    - Then want to solve everything with that 1 language: If your only tool is a hammer, everything looks like a nail
    - But programming language is like a tool, a hammer; choose right tool for problem
    - Example: C great for system work during its time; or Python today, is a great scripting language; but neither not right for everything
    - See also example of databases; pypl.github.com/DB.html
    - Selecting database type (mysql / no sql / triplestores) choose according to what you need
    - People always think ‘it works here; then it works for everything!’
- Eternal cycle 2
    - Centralized => distribute to get more freedom => diversity unmaintainable => repeat
    - Overlay on this mainframe/terminal => mini-computer => pc => networked pc => client / server => thin client (not much more than terminal on mainframe before) => Cloud etc
    - Don’t see difference today in Cloud solution and mainframe setup; you interact with terminal and all control is on other side



2010

- Computerized information processing is ubiquitous
- How do we find answers to complex questions?
- In talking about AI, we have a problem with ‘natural intelligence’
- Saw article from yesterday; german minister of environment said ‘environmental protection must be implanted in every algorithm’
- Shows complete misunderstanding of what an algorithm information

Disruptive innovation

- Think for a moment, what is the big disruption in It in last few years
- One proposal: washing machine
    - There’s article saying washing machine done more for liberation of women than contraceptive pill and abortion rights
    - Displays bias of time of women doing washing, but looking beyond that, think about how much effort it took to manage a home before certain inventions; think about life in city where you had to bring things in fresh
    - People tried 100 of years to have mechanical devices to do things like washing, refrigerate food;
    - Not always clear what big inventions are
- Hype Cycle for emerging technology (https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/)
    - Example of self driving cars
    - Up to particular level of conditional automation, still require human driver as fallback for dynamic driving task
    - See in hype cycle for emerging tech, fully automated driving is still more than 10 years away; so when you see in news articles about self driving car, know fully self driving is far away

Wikipedia etc

- Amount of knowledge you can get at home is huge
- Thinking about what you had to do 10-20 years ago (write, proofread, send to publisher, print it, etc.)
- Desktop publishing revolutionized this
- Around 1985, multiple computer companies introduced desktop publishing
- Disruptive tech mapped on graph of performance over time; performance going up

History of Neural Networks

- Conventional deductive programming
- If you don't have problem to solve, can go to data driven + stats
- Machine learning (aka statistical modeling) does this alt approach
- But problem with machine learning is you have to choose model beforehand
- Neural networks have it where you don't have to choose model right away
- Deep learning can approximate a function; but you need lots of data

Examples

- Can train a network, can always get observed solution out of nowhere if new
- Discussion at Wuyishan workshop: what doesn't industry care about this so much? Because 90% of work is data massage
- Library community understands what it means to have a big dataset, clean and maintain over time
- There is no such thing as a free lunch here; you have to do the work on the data to get this in
- Nobody says out loud that folks at places like Google that use neural networks required a lot of dedicated work by many to prepare

Think and act problem centric

- Find right tool to solve a problem; don't select tool then fit to problem
- Progress is impossible without change; change is end result of all learning; and anyone who stops learning is getting too old (all quotes)

Questions from audience

- No time left
- Slides will be put online

What automation is from a Nigerian perspective: a challenge of sustainability / Babarinde Ayodeji Odewumi

- Here to share an experience primarily about what automation is in Nigeria experiences
- Speaker works at software development agency
- Primarily works for libraries, schools, other enterprises; builds systems
- What is library automation
    - If you don't know this, he may be in wrong room (joke)
    - Basically; application of computers to perform some traditional library activities such as acquisitions, cataloguing, etc. (islmblogblog.wordpress.com)

        - Correct to an extent; simple as this definition may be, we still take things for granted
- Issues with automation
    - Software complexity - product & user fit
        - Over years, 20-30 years in Nigeria, see complexity not in what software can do, but in users themselves
        - Think about how many people didn't have experience with computers even in 2000/2003; working with computers then becomes a challenge
        - You had then systems that were well built, but not built for users that they had
        - Cataloging a book in that world took so long because user had to do so many steps
    - Legacy bus stop
        - You selected something then were locked in
    - System crashes
        - Having years of work lost because of system crashes with no backups made
        - Example from speaker of bring server back to life in such a case; vendors not available to help with bringing server back up to save work
    - Technology envy
        - Example of software Koha; good system, as it were; but many people pick it up and run without having a process in mind
        - A bunch of libraries use it, so we should use it type of mentality
        - Boss buys such tech, brings to library, says 'make it work'
    - High cost of entry for technology, infrastructure, licensing
        - Libraries have limited funding
        - This is an issue as well
    - When you mention automation then in libraries, there is skepticism
- Consequences of the above
    - Many people have bad stories, are heartbroken, feel technology is a hoax
    - Patrons run away from library because libraries don't meet their needs; they're then on Google or other places instead of libraries
    - Libraries exist to meet needs of users
    - Example: call numbers can be opaque to many users; you see QA1XX for algebra; but user just wants to find 'algebra'
- Readable LMS
    - Has most traditional modules
    - Able to do complete cataloging; exposure of data via SRU, APIs
    - Have integration with indexing code that can get users search and access to content
- How did we get there

        - Understanding users, what their challenges are; level of exposure; etc.
        - Got the expectations down; for example, understand steps for accessioning, get steps down without lots of complexity; call back to keynote speaker's 'cycle of complexity' (simple => add features => complex => rewrite to be simple, repeat)
        - Most fulfilling thing is seeing person searching for materials and finding it
        - Migrating data and getting it into a standard format
            - Pull data from legacy systems
            - Because of infrastructure; instead of servers, decided to come up with SaaS solution that has custom deployment in cloud; can scale as choose to; able to get libraries off the ground in this work
        - Need also continuous feedback loop
            - Want this to also be a continuous learning process
- Tech stack
    - PHP web-based application; using ZF2 framework + few other libraries
    - Docker for containerization with Dokku server; Dokku allows subdomains mapped across containers;
    - MongoDB as back end
    - ElasticSearch as index for searching of resources
- Future
    - Looking to help libraries integrate with full-text databases
    - Want to integrate with other institutional resources; dspace, eprints, etc.
    - Want to integrate with these because want single point of entry, discovery for user looking for materials; user just wants to find resource + use it; good then to aggregate resources together
- Questions
    - Coming from so far away, but hearing about problems you're working with, sounds like here. How many libraries do you serve? What types of libraries do you serve?
        - 10 libraries. Have university, public and school libraries. Don't have special libraries yet.
        - Way users interface with libraries across this is quite different; interfaces need to be tailored to each
    - How many people work in your team?
        - Primarily 3 people; have other consultants, engage with library consultants to know how best to approach particular problems, to get other perspectives on approach
    - Could you give an example of how you target UI for working better for people new to computers?
        - Typical example; to some extent, for library users, it's about simple workflow in terms of integration. When cataloger is cataloging book, there are some books that don't have CIP

information; particularly if published in Africa; so probably not finding bibliographic materials anywhere
        - At point where cataloger is starting work, they do a transform of primary data from acquisitions system to metadata of sorts; do that transform so then librarians can then just do edits + corrections; have some work done for them
        - Second part of this - for patrons, simple search field is UI example of this
    - How do you feel international committee infrastructure is helping you proceed in questions you raised in your talk? Is there better way to exchange knowledge and systems?
        - There isn't much dependence on "system" in home country; do what they need with what they have to move forward;
        - Drive to push things forward relies on libraries to push the above;
        - Limit to what speaker can do in international groups without libraries driving this;
        - Want to get more people involved to help
    - Last slide said most people are using mobile phones to access systems; do you do special things to accomodate for mobile users?
        - Web Interface is responsive
        - Building mobile apps for users to engage with library resources
        - Better to have app then needing to type URL in browser on phone

Libris as Linked Data - Production Experiences / Anna Berggren, Fredrik Klingwall, Niklas Lindström

Slides: https://tinyurl.com/elagxl19

Going back in time / recap

- Been at ELAG a few times before
- In 2012; had wikileaks occur, avengers had premier; leonard cohen released album 'old ideas'
- Year the Libris project started
- Libris scope
    - Centralized National Union Catalog
    - Data is collaboratively edited
    - Publication is catalogued only once
    - Other libraries need to register holdings
    - Records export to each system
    - All above built on principles of openness, cooperation, OSS
    - Includes swedish national bibliographic
- In 2012…
    - When setting expectations of new system, swedish national library already responsible for maintenance and development of new systems

        - Want to describe more materials more quickly & handle growing data volumes
        - As internet comes along, library data was stuck in silo, so to speak
        - Second mission was to incorporate more with web; choose what others have produced
- Basic requirements
    - Had to convert existing MARC 21 data to something in JSON-LD
    - Knew they wanted RDF
    - Needed to convert not just cataloging of data but also automatic import routines of data from suppliers
    - Postgres & ES for storing & searching;
    - Had to convert data back to MARC21 for export to other systems need thing
    - Wanted to expose data via APIs
- XL
    - Core of new libris system
    - Basically, wanted to cut out Voyager system, replace with XL system & that cataloging interface
- Editing tool
    - Filtered window into libris data; not a tool for discovery or circulation
    - Link for staging environment at end of presentation
    - Editing interface can link stuff; create anonymous bnodes; have structured value descriptions
    - A lot of the catalogers still dependent on how MARC looks; have in editor conversion on the fly for seeing MARC
- Why not MARC? Why links?
    - For some, this is obvious; to others, not so much
    - From abstract PoV, it is obvious
    - You can to some extent use links in MARC, but you cannot normalize data, export links, etc (missed some of this)
    - Links enable reuse of data + sharing of data in new ways
    - Some of this is quite visionary; keen on pulling in other data
    - If other libraries go forward with their proposals around LD, they can use other authority descriptions instead of just combining
    - Another crucial aspect in this migration is getting rid of notions of how MARC21 makes us think; break those defaults of systems + mindsets
        - Other data models and vocabularies enable new views
    - This is technically hard to do; can easily get lost in the details
        - Once you get into the details, you can start to lose faith in why you're doing this
        - Had to focus on both perspectives at the same time
        - Knowing local systems won't be updated anytime soon, have to work in both directions

        - MARC is very structured + neat, but not obvious to everyone what it represents
        - Very strange pieces can show up, hard to share data / find out particular parts cannot be shared without some other detail
        - If we get rid of too much, pile of stuff export to other systems will contain more or less data, not exactly same
    - BIBFRAME 2.0: needed something they could bet on other libraries using as a common framework; RDF is just a structure for semantics
        - Build upon BF 2; fairly safe bet since LoC is behind MARC21 and is behind also BF2
        - Started before BF2, but luckily able to remap to BF2
        - Added OWL description, for more constraints + description, adding things not in BF core
        - All things in editing interface is driving by this vocabulary;
- KBV + Plan B
    - Bunch of pieces they are told people require; puts some tension in building this out
    - Many mindsets need to be change in order to communicate in terms other than MARC21
    - From outside perspective, this is scary to understand
    - Still talk about bib records in new system, though it isn't true
    - For catalogers, quite a disruptive change, especially since most don't have formal training in RDF etc.
- How to prepare for change
    - Went into production with system before totally done
    - Forced catalogers into system where they couldn't do everything they were used to
    - Needed to have work in place to move in forward though; having the UI up is where they could then discuss changes with catalogers
    - Started with a roll-out project; how to prepare catalogers for this workflow change?
    - Involved colleagues at national libraries; involved catalogers themselves, to write help text for enabling catalogers to do their work in new format
    - Create manuals + short films on how to catalog in new interface
    - Published a lot of text on principles on bibframe, new model, because communication was considered key in this plan
    - Organized tests for users before going into production
    - Have continued with user testing after going into production
    - After introducing new system, could see the libraries that prepared for chaos could do this transition in better way than those who just waited + did nothing
    - One library had flow chart preparing for chaos, i.e. where do i go if i can not do anything, if nothing works

        - Trying to get libraries to focus on learning, not just productivity + quality hits
        - This new format created a new kind of equality among catalogers between junior + senior staff, since all had same knowledge now
- Current development work
    - Project is ongoing; roadmap goes until september 2019
    - One problem; don't link to extent they'd like to
        - Normalization + link of things like types vs content or carrier
        - In RDF, can extract details and put into explicit place instead of having catalogers duplicating information; but not there yet, they still sometimes require cataloger to fill in info twice
    - Linking of component parts for things like serials + series is hard
    - Missing links for Works; the Work descriptions currently are bundled with Instance descriptions, like with MARC21 bib records
    - Hoping to make some of the above work operational this year
- Lessons learned
    - Have to move forward without getting these pieces in place
    - Trying to get at actual needs users have in libraries
    - So many details that can make you lose your way in this work
- Questions
    - What is MARCframe? BIBFRAME / MARC combo?
        - No, name of mapping tool; hoping for collaboration uptake + adaptability of this, but hasn't happened yet
        - https://t.co/1EaXLlx3Bs
    - Started work in 2012; as you have lots of experience, and given current state of BIBFRAME + preparation of MARC, what would your request be to libraries trying to change to other systems?
        - Just do it. Be prepared for tough questions. Look closely at BIBFRAME, but do not fully rely on it for being the solution to every problems. Use RDF correctly. Very dependent on resources you have as a library. Would encourage everybody to look at it.
        - Engage your users.
    - Do you perform reasoning on your data, and where is that done; you dump data as JSON to ES store, so no SPARQL endpoint
        - Formally, no, do not use OWL reasoning; using restrictions mostly in editing interface;
        - Do not expand triples
        - Reasoning may not be the right tool here; many things to talk about around this, how use shapes of data

Yet another ILS? Why and how. / Igor Milhit & Nicolas Prongué

- Studied together, working on specific project for ILS
- slides.com/ignami/rero-elag-19

- RERO
    - Library network of western switzerland
    - Rero provides products (ILS, etc)
    - Provides services - training, data processing, professional coordination, host library services for things like digital newspapers, has IT centre, etc.
- 2014
    - Important library group announces retirement from RERO
    - Next year, project started to get all libraries on 1 common ILS
    - Unsustainable for RERO to exist with reduction of collaborations between libraries, so had to reinvent network
- RERO21
    - Transform network into non profit foundation, with own OSS and in house ILS, open to whole Switzerland, targeted at schools, public, and heritage libraries
- ILS
    - Want own OSS ILS. Why OSS? Have lot of expertise in IT, so this gives them full control in how to do things; already use a lot of OSS tools too
    - Wanted a web app, SaaS but hosted on servers in office, not fare from future customers
    - There are multiple services - rero ils, mef, sonar (sonar.ch) last gathering all open / OA scientific data
    - For all services, wanted to use 1 framework: invenio
- Invenio
    - Doc.rero.ch for +15 years
    - Developed by CERN
    - Is highly modular, to easier to expand
    - They store data with postgres, map data to elasticsearch
    - Using python and flask for webapp modules
    - Jinja templates + angular application for UI
    - Use web standards in this work
- Achievements
    - Focusing now on circulation and UI
    - Using invenio-circulation; have state chart oriented on load; idea was put all the circulation information in item records; but decided then to make a new record so circulation starts with creation / load of record and ends with return
    - Put their own configuration on top of this
    - To get circulation module up and running, need to load every type, get library hours in
    - For UI, tried to be clever. Have only 1 input field, and system tries to then determine record type and return the most likely action you want to do
    - Not sure yet if librarians will be able to work with it, needs testing
- MEF

    - Multilingual entity file
    - Built MEF service because wanted to offer search + display with multiple languages for user; requirement in multilingual switzerland
    - Tried to use authority files as much as possible
    - Built map service; end user goes through another interface like ILS; ILS gets to MEF; libraries are on other end working on authority files (GND, BNF, maybe wikidata); intermediary layer gathers those sources; merges then by means of VIAF; creates for each person a French record, German record, etc.
    - MEF is only an API; services through interfaces
    - Libraries can choose independently what to contribute if they want; or just use wikidata + that is source for MEF
    - Multilingual cataloging only available in MEF, only editor that is natively such
    - Current state: only 1 entity ‘person’
    - Testing / discussing with Wikimedia for how to integrate data with wikidata
    - tinyurl.com/y2ph4ss9 <= open call for comments on wikidata work
    - Can also test service - https://mef.test.rero.ch/
- Challenge: Data model
    - Implement a full data structure? Decided to keep WEMI model, with work optional
    - Not really able to change data model without changing system more
    - Don’t care about standard because building own ils; but they do care about staying compatible for import and export
    - Data stored as JSON; inspired by BIBFRAME, but needed to complete it / give more specificity
- Challenge: Consortium
    - Will have multiple organizations with possibly multiple libraries, with big union catalog on top
    - However, an organization can decide to have library-specific views, views that vary, organizational groupings that vary
    - Still not sure what will be discovery needs in future along these lines, so trying to keep work open / extensible
- Challenge: Lots of tools that are evolving quickly, so that can be frustrating as well
- Timeline
    - Hoping to have first pilot libraries at end of 2019
    - Want customers in production by end of next year (2020)
- Takeaways
    - Doing a lot of work to to develop competencies locally
    - Wanting to stay independent from vendors + providers
    - Ils.test.rero.ch
    - Mef.test.rero.ch

        - Test it, test ILS is deployed regularly with data wiped out; would like to get that feedback
- Questions
    - How are you engaging users of prospective libraries?
        - Doing testing now, hoping to get circulation librarians feedback in particular soon
    - Did you consider any other solutions?
        - Yes; but wanted full control of software
        - Invenio works like a framework, so with different solutions can do many same things; hoping framework will save them some energy
        - Have some good faith in this approach, but waiting to see if it works
        - Otherwise, have multiple systems for ILS, IR; looking at invenio as a swiss knife
    - Found concept of cataloging in wikidata interesting; in case of multilingual search cataloging, are there strategies for data deduplication, quality control? Versioning of data?
        - Had discussion with wikimedia about this; they are very enthusiastic about this; data control is still an open question though. Wikimedia says there is no user deleting data (they're biggest concern); also worried about users changing variants or forms; option they want to have is user changing name for a good reason; and if library doesn't want that, they can choose to use instead GND or BNF; reasonable expectation to have records to compare + monitor what entities we're deleting, and this is something they'll probably have to develop
    - What are options for others to use this system?
        - They are open to work with anyone who wants to work with us
        - Not sure yet in new business model if they will have support services
        - Later, when they are more stable with work in production, they will investigate; hoping in few years to create community around this

Improving FOLIO Architecture / Julian Ladisch, Martina Tumulla

- What is folio?
    - Goal of project is to develop OSS library service platform
    - Software is for librarians to manage daily work
    - Target group is academic + research libraries for now
    - FOLIO is a product / software but also a community
    - Founded as open source project by multiple groups, stakeholders
    - EBSCO, one of stakeholders, is funding contracted dev teams e.g. 25 FTEs
    - Bring in own human resources: product management, etc.

    - Index data, software company from copenhagen, so responsible for basic technical architecture
    - Funded by Mellon foundation + membership fees
- OLF
    - Open library foundation; provides infrastructure + secures open source code for projects in higher education
    - Projects include FOLIO, OLE, GOKb, etc.
    - Project structure has a lot of committees + product groups
    - Subgroup of product council is technical council, which sets tech standards + direction
    - Working groups (SIG) each have convener that is moderator; discussing requirements + use cases; there is Product Owner gathering requirements
    - SIGs on resource management, etc.
- FOLIO Releases
    - Have quarterly releases
    - Add new features in each release
- Short overview of UI
    - Top is function bar
    - Searching functionality embedded
- FOLIO Architecture
    - Open platform: LSP
    - Platform provides infra for functional modules
    - Functional modules are self-container programs
    - Based on microservices;
    - Interfaces need to call each other if communication needed
- Tech Concepts
    - Based on various models
    - You can use a cloud hosted option or run it locally
    - For cloud system, have multi-tenancy
    - For each library, can configure to work with own tools, etc.
    - Meant to be a very flexible system
    - Want a plug and play application
    - Key thing is APIs all the way down, so any dev can interact with any layer of platform; and each module needs to stay small so easier to replace it
- Stack
    - UI toolkit - leverages React framework
    - Basic LMS Apps - erm, acquisition, cataloging, etc. Choice of programming language
    - Other apps
    - Folio gateway: Okapi, kind of switchboard; if app wants to call another app, it goes through Okapi while also ensuring tenant separation
    - System layer, with database connection, tools, indexing, logging, etc.

    - Tech
        - Javascript
        - react/redux
        - Java 8 moving to 11 soon
        - Vert.x
        - RAML
        - Postgresql with jsonb + relationship SQL
    - See slides (shared afterwards, skipped now) for more details
- Tech Evaluation
    - Platform evaluated 3 times - by OLE community members, by EBSCO, by FOLIO committee (missed it)
    - Seen as good
    - Accessibility evaluation as an example
        - Stripes is FOLIO's GUI toolkit, providing reusable components
        - Designed to be accessible
        - Accessibility checked regularly, in usability labs and in monthly power hour
        - wiki.folio.org/display/A11Y
        - https://ux.folio.org/docs/guidelines/accessibility/
    - Query Language - CQL to GraphQL
        - CQL - contextual query language, found it didn't support the more complex queries
        - Created GraphQL for these complex queries
        - Example of joining three tables + return only few fields; not impossible with CQL
    - Tenant separation example
        - Several libraries running same system on same cloud, then current idea is create new user role in database, new schema, so separate logical database; have unit tests that check this works, you can't get data from other libraries
        - OTS report found that if you create new tenant, than you pass in some super users rights which is not good, so need to change architecture
        - Needed central service to manages roles, creates, and passes info to module
        - Better design
    - Additional DBMS Support
        - In theory, can select any database
        - But currently have RAML Module Builder (RMB) that only supports postgresql (used bc reduces boilerplate code)
        - OTS recommended additional DBMS back-ends
        - FOLIO has postponed decision; asks audience what they think; should the prioritize supporting other databases systems?

- Questions:
    - If in microservices here, they are sharing databases. If I write my own circ module with own database, is there way to ensure data isn't getting across?
        - Wrote API definitions / specs; if you want to replace circulation, then it just needs to work with existing API spec
        - Concerns of module dependencies; any module cannot directly access database; always goes through Okapi to get to database, so management of access is there
    - Can use other Java (??) like OpenJDK?
        - Use OpenJDK, not certain there are implementation problems with using something like Oracle or other java implementations
        - Can always submit a bug report
    - What is the bee thing?
        - That is our logo; it is a bee; meant to be like a beehive representing a community,
    - Seems like any large, meaningful change will start propagation of issues to other microservices that could be difficult to control
        - (original question guy raised something else on microservices + database connection, missed point)
        - Another person: who is allowed to make changes to tables in database? Speaker: only module who owns database;
        - (a bunch of other discussion on applications dependencies in microservices)

Lessons learned while building the vocabulary mapping tool Cocoda / Jakob Voß

- Slides: https://doi.org/10.5281/zenodo.2677600
- Coli-conc is named of project
- Nearly 20 years ago, colleague was looking at automatic classification
    - Trying to take an existing call number and analyze it
    - An example: going from DDC to wikidata classification
    - Mappings between vocabs are then what they work with
- Goal of the tool: create tool for making + managing these mappings
    - Facilitates concordances between knowledge systems
    - In particular for classifications like DDC, RVK, BK, and the many local library schemes
- Tasks
    - Collect metadata about classifications; used BARTOC registry to find this metadata; if someone has their own classification, asks first for them to create registry there
    - Collection and publish existing mappings

        - DDC/RVK/GND/BK/STW/LCSH/IxTheo
        - Wikidata (as of 6 / 2018 ; stopped being updated)
        - One more (missed it)
- Tool for this: Cocoda
    - First prototype in Angular: allows browsing of vocabs
    - Stopped then to write project grant and receive additional funding
    - Needed to write their own data format - JSKOS (json-ld for skos)
    - https://gbv.github.io/jskos/
    - Adding additional properties for other ontologies; need more metadata on authorities & get all data into this format
- Getting data into JSKOS
    - CSV, MARCXML, SKOS => Cleanup
    - Using skos2jskos, jskos-convert, many other tools
    - Tools are less important, however, since you do it once then load it
- 2016: got grant funding and started a new implementation
    - Was a monolithic java application, threw it away afterward
- 2018: started from scratch again with Node & Vue.js
    - Much faster development this time around
    - Have a general layout
    - Coli-conc.gbv.de/cocoda/app
- Live Demo (in Browser; doesn't work with IE)
    - Search in top left corner
    - Find classification information in one box
    - Have search on top right corner, second item to find
    - In middle: can create the mapping
    - That is then saved locally in browser;
    - You can then load that mapping into Github or other providers
    - Trying to improve UI for seeing what mappings already exist
    - Trying to get as many mappings and as much data now input; then aiming for data QA as a second phase
- Infrastructure
    - Jskos-server & DANTE terminology registry (jskos-api)
    - Mapping suggestions linking to OpenRefine Recon API
    - Have OAuth login-server
    - Except for jskos api, try to follow other existing API specs where exist
    - All OAuth / Authn is managed outside of app; not configuring Authz yet until needed for roles
- Lessons Learned
    - People (still) stick to spreadsheets
        - Trying to make simple, documented JSON structure, but people still want their CSV
        - At least better than MS Word
        - All the issues with making this interoperable with Excel

        - Software dev is communication
            - Listening for what is actually wanted & trying to understand where other projects (like DCMI) fit in
            - Face to face meetings are unavoidable to find a common language, grow understanding
            - Need to explain tech decisions + aspects like URIs and Open Data
                - Try to get URIs, make sure vocabulary is open data before putting it into application
                - Examples of like english version of german classification vocab is not available openly, has to be locked down
            - Not all features developed yet, so requires explaining
        - No schema, no data quality
            - Notations and identifiers must match regular expressions
            - JSON Schema helped to find inconsistencies in JSON Data
            - Additional constraints not expressible in JSON Schema
            - Never trust any data you haven't validated! If you don't validate, you're sure to have crappy data
        - Holy decoupling
            - Went with a service oriented architecture, with APIs and data formats mattering most
            - Things will break anyway
            - Even with having thrown away 2 prototypes before, could easily transport data between them
            - Also easy to replace parts in infrastructure: RVK API => own database => Dante => jskos-server as one view
        - Look out for beneficial beta-users
            - Many librarians want to wait for a final product, then take a course how to use it
            - If you can find users willing to use an unfinished product, work with them
            - Aim for real use cases and outcome instead of click-around testing
            - Need agile users to do agile development
    - Encourages people to try out the tool and give feedback
        - github.com/gbv/cocoda
        - gbv.github.io/cocoda
    - Questions
        - Did you try to use workbench?
            - Started before workbench powerful (missed rest of answer, sorry)
        - Data is stored locally then pushed elsewhere, have you considered pushing to something like wikidata directly?

        - In browser memory doesn't require login; have work on mappings registry; but planning to add pushes to other services, with wikidata being top of that list
    - Is the demo a sandbox?
        - (missed this sorry)

How to make your cataloguers popular among researchers community? Give them – at least – a new software! / Aline Le Provost

- Paprika.idref.fr
- Last year, did a lightning talk to show you a tool developed at the time called paprika; today, they've opened it to users, have about 1500 university libraries in france using it
- Talking about this today, with focus on how to make users happy with this new tool
- Abes
    - French national agency for higher ed / academic + research libraries
    - Manage a general union catalog, dissertations, archives + manuscripts, authority files (Sudoc.fr, theses.fr, etc.)
- idref
    - Idref used to identify french research activity
    - Trying to map idref to work like ORCID, ISNI, etc.
    - Using owl:sameAs with transitivity to make these connections in their system
    - Links => Content => Identifier
    - Looking at authority file example; shows you can see all the bibliographic records from ILS or other sources; as well as links to ORCID
    - Trying to work with researcher community to show this
- Building this project + tool
    - Tried to work modularly to produce, export/reuse, etc. authority data
    - Example of the web interface of the tool
    - In web UI, you can see all the sources you can link to / from
    - There is a SPARQL endpoint for the service
    - Last year, did an application to connect to idRef interface + give key data (name + dates), if authority found off that, it allows data flow between authority systems, consumers, and idref
    - Provide a matching service - give it your data, they get idRef IDs
- Goal:
    - Making a quality team of hundreds of cataloguers becoming data curators, across institutions
    - Believe that open database has to be managed collectively
    - However, the reality of this is cataloguers are facing immensity of work

    - Trying to trigger a virtuous circle at ABES - personalized, dynamic reports (things like duplicates, missing links, etc.), dedicated tools, and authority data experts in a cycle
- Paprika is the dedicated tool for this now
    - Scope of the tool today is the union catalog - so bib records from SUDOC, especially person content; and idref, the authority database
    - Tool was built bc a bit ago, cataloging tools weren't easy to use; looking at ILS example, if you're trying to find link mistakes between bib and auth records, hard to compare across records; have multiple screens
    - So, proposed to them to use this tool
    - Tool UI shows bento style approach, with multiple authorities in boxes, and smaller boxes have bibliographic records access points; Can see where bib records have no access point link to an authority; user can then check if links correct, move bib small boxes inside auth big boxes you're wanting to link to
    - One unique screen, with extended search + focus on links
    - Zooming in + out + traceability of actions also included in tool
    - Paprika has link to a logs table (kafka?) where events are stored for created / updated connections; those are then to be displayed in another interface
- Paprika as a decision making tool
    - Name paprika came from this
    - Web UI has colors to help with links quality diagnosis (green is link is correct; red is link is wrong; yellow means needs more evaluation or manual work)
- Qualinka: how it works
    - Skipped due to time
- Live Demo
    - Clicking on small box access point, will see modal with more information on the bib record; can compare it with authority that is linked to that access point; have synthesis of authority; then seeing something that shouldn't be linked, can move (click + drag) access point to correct authority box + create that link
    - BILAN / log of actions, and those actions then update SUDOC database
        - User is required to hit 'save' button for those actions before it is written to database
    - User can also create a new authority from this web UI as well; create a new box, click a few buttons, and import data / information from existing sources
- How to get users (librarians) to play with this?
    - Did some stats analysis, and see users are actually using the tool; even breaking it with so many requests
    - About ¼ of institutions of network is using it

    - Have a few great contributors involved
    - 88 users total, from 73 libraries + 57 institutions
- To be improved:
    - Ergonomy
    - Integrating with monitoring tools (dupes, missing link reports, etc.)
    - Add more data sources beyond Sudoc, like Calames, Persée
    - Would like to add authority merging, add settings to search functions
    - Plugin suite they'd like to add:
        - Make paprika available for any database connected to IdRef
        - For this to happen, questions on how to get data + also how to synchronize from log table to external source database
- Potential sister interfaces
    - For other entities, try to expand to also corporate entities + works, or target ISNI, BnF, GnD, etc.
- Questions
    - Can you tell us about APIs you use to talk with other data / databases?
        - For SUDOC, have webservices from ILS (?), and those give the data in JSON, as well as webservices to update CVS(?) database behind sudoc
        - For databases other than SUDOC, would need those webservices but don't have them yet
    - Example includes Personal Records, do you also use it for Subject records?
        - No. At the moment, just worked on Persons. Potential future work
    - Have you worked with other / would you work with other institutions outside France?
        - Want this to be the primary researcher identifier service for research in France
        - (questioner: Was thinking about integration with something like ORCID, VIAF, etc.) Project is to use this as pilot identifier; have other activity to link idref to ORCID for french researchers

Workshop Day 1: Apache Nifi automating data flows between software systems

- Idea: want to process some PDF files, put into a good pile + bad pile
- Context for Ghent:
    - Have a lot of data across many systems that is going into Blacklight
    - Use Catmandu currently for a lot of this work right now
    - However, there is a lot of black box scripts running these
    - Hard for recovery then from failures
    - They all need a lot of integrations for various services
        - Aleph (ILS)

        - Lockers + logistic systems
        - Humans working with systems needing messages or otherwise
- Data Pipelines
    - Trying to automate as much as possible; have these pipelines do the processing for you
    - Help get staff to focus where they can work best, like cleaning up dirty marc with your favorite data tool
- Many dataflow solutions exist beyond Nifi
    - Nifi, flume, kafka, spark, etc.
    - Is dataflow programming itself something we need in the library services stack
- Hands on section (will grab slides and share here)

Workshop Day 1: IIIF

IIIF Workshop

8/5: Introduction to IIIF, demos, Q&A
9/5: World café about suggested topics, 4 tables, 15 minutes per topic, hosts record ideas and topics (servers and software, viewers and frontend integration, data conversion, data modeling)

Lightning talks suggestions: https://pad.okfn.de/p/ELAG19IIIF-Talks

(Brief introduction to IIIF)
IIIF is a framework that makes it possible to share images between different systems (Interoperability). For example: showing images next to each other to make comparison easy, place images on top of each other to show translations, bring images from different institutions together and combining them into one big image, … The possibilities are endless.

The 2 main IIIF API's are:

- Image API: for pixel data delivery
- Presentation API: for metadata in JSON-LD format

The basics needed for IIIF:

- Data: Metadata (MET, MODS, MARC, …) and images (TIFF, JPEG2000 (proprietary), OpenJPEG, …) in multiple resolutions (tiled pyramidal). You could start with existing lower quality images, but then the result will also be low quality.
- IIIF image server
- Manifests: a way to create IIIF Presentation API manifests from the data to serve to the IIIF viewer. This manifest contains where the images are stored, this can be at different locations/institutions. (Of course it contains also a lot more :))

- IIIF viewer to render the IIIF images. Not every viewer supports the same IIIF functions. For example: SVG polygons in stead of rectangles for coordinates, annotations, …
  For example: Open Seadragon

Links:

- https://iiif.io/
- https://github.com/edsilv/biiif (create static manifests, easy for testing)
- https://digital.bodleian.ox.ac.uk/manifest-editor (online manifest editor)
- https://demetsiiify.jbaiter.de/ (online tool to create manifests from METS/MODS files)

Demos:

- https://papyrusebers.de/ - long papyrus scroll, cut up over time, reconstructed through IIIF + lets you see translations through annotations ("Übersetzung an/aus"), modified IIIF viewer
- Mirador demo: http://digital.ub.uni-leipzig.de/mirador/index.php , compare Codex Sinaiticus to Codex Vat. Gr. 1209 even though they are in different institutions
- Combining images into one IIIF manifest also possible, eg. http://projectmirador.org/demo/advanced_features.html , manuscript w/ cut out miniatures, possible to reconstruct through layers interface, combines several images into one canvas - manifest: https://demos.biblissima.fr/iiif/metadata/BVMM/chateauroux/manifest.json , multiple images within same canvas can also be used for different images of same page (eg multispectral images)
- https://antlitz.ninja/ auto detects faces in images, lets you mix and match faces from different paces, pulled from server from IIIF, faces have coordinates through annotation API which lets the tool zoom to the correct height
- polygon annotations in manifest: https://api.digitale-sammlungen.de/iiif/presentation/v2/bsb00110131/manifest
- annotation overlay in writing direction: https://www.walter-benjamin.online/seite/archiv/wba_363_001, slider lets you go between layers, transcription was done in TEI-XML, implemented w/ OpenSeaDragon
- https://iiif.cloud/ - IIIF discovery that lets you see collections published by different institutions in one place (elasticSearch)
- More cool IIIF projects: https://github.com/IIIF/awesome-iiif#experiments-and-fun

Day 2 - May 9

If you share your authority file - what happens to your authority? Insights into the opening process of the Gemeinsame Normdatei to realms beyond libraries. / Barbara Fischer

- Asked to talk about authority files; not a topic that gets her going per se, and a year ago it wouldn't have been her choice of topics
- Starting with a warm-up, ask the audience some questions + move around a bit
    - If you've been to elag2019 more than once, raise hand; more than 5 years, both arms, more than 5 years, stand up (left with about 10 people standing)
- Goal of talk
    - This conference has been held for over 31 years, so it is much older than the gameboy (first in the market in 1989); older than the world wide web; older than Google (98); older than wikipedia; older than twitter; and surely older than most digital (??) in the GLAM field
    - Digital transformation is incomplete; still treated as anomaly (?)
    - Many of the audience work in technical departments in libraries
    - Asking audience how many people have digital publication structure in place; and what it means to provide digital publishing and transformation
- Digital transformation
    - Informs entire library as a system
    - Instead of an institution providing defined services; digital transformation involves APIs always open to clients; it removes the librarian as the gatekeeper
    - Alex. Berkin CEO quote
        - Library will be agile and transparent in management decisions; library will ensure flow of information in all directions; will respond, flexible to requirements of customers, and it will have a profile of its own
        - Institutional citizen protecting nand fostering democratic society it relies on for its existence.
    - Audience: raise your hand and show if you're on road to digital transformation
    - Museums and archives can learn from this as well
- German National Library building in Frankfurt Main; where speaker has worked past year
    - Has legal mandate for e-publications + data since 2006
    - Not entitled to share it by evident digital means
    - You can read this stuff by booking a computer in the library

    - Streaming practices, music industries, copyright societies remain a challenge to this access
    - Legal mandate is linked to a product by definition; in 2019, this streaming is a daily practice for increasing number of individuals; however, digital time lag is leaving libraries in dark period
- Digital transformation is coming; about linking, identifying, managing, organizing information + knowledge
- Librarians...
    - In ancient times, it was a holy office
    - Later, still respected
    - Even later, job on edges of cultural work
    - Now, as more data online, there is more need for finding the bit you are looking for
- Authority files
    - Central tool for organizing this knowledge in libraries is authority files
    - Want to talk about what authority files can be + how they force a somewhat digital character before TBL; the link codes, authority codes, encoding text strings for reading by machines + humans; but this is not enough
    - Example: think of name Greta (?); and how it links to what. A name is not enough. You need a further property to identify, to disambiguate. Link to birthdays; georeferences; works;
    - Label, id, url in 1 disambiguation property.
    - System then gets more complex. Items become properties of other items, become subtypes of other items, pointing to other databases and etc.
    - Historically-bound, complex + not always structured tool is Authority File. GND for the German speaking world
- GND
    - Uses entity relationship model;
    - Relationships are defined by code
    - Modular data structure
    - Follows international + national standards;
    - Services MARC21 + other serializations
    - Contains today 15 million entries
    - Held by German National Library
    - Provided under CC0
    - Wants to further retrieval, attribution, context, etc. of names
    - Implicit clue of authority file is it’s reliability; people have faith in the institutional contributors provided high quality data;
    - Until now, GND is quality tool produced by librarians to serve mainly their needs; For now, those who use, edit, develop the GND are librarians; exceptions confirm the rule, but things are changing
    - Experiencing impact of digital turn;

- Need for persistent identifiers, 3 reasons for this need
    - **Matter is Changing**: Used to be matter records history of universe bc slow cold form of information. Being able to store + share information on road from campsites to cities of today; for quite some time, the matter was paper + ink. Books / works were easiest way to store information. Today, matter storing information is much faster + not solid. Moving around world from one cloud to another. Data is fluid, as it swiftly transforms from one form to another; before what costs effort was storing the books themselves; today + tomorrow probably, what is more difficult is sharing. Not because it is difficult to share a file; but it is so easy that is the problem. So much more random access + data to share. Looking then for anchors + orientation
    - **Scientific apparatus is changing;** Increasing number of scientists in last centuries; terminology became a way of distinction. Knowledge is about our world, our past, and we as in humanity. Knowledge is a common asset to be shared. Today, there is much more exchange across disciplines. Scientists are not only sharing results, but working in teams, sharing methods, data. Digital methods make it easier to share across space, as well as to integrate data from places other than canonical sources in a specific discipline. For this to work, however, we need to be able to understand each other. Need a reliable mapping between terms.
    - **Digital environment is always changing**. Have been able through time to contextualize + retrieve information from stone, paper, etc. Sometimes, content was almost lost. Have faith in preserving words for our knowledge; storing data in clouds seems more redulant (?) compared to tablets. Who will be able to understand our digital born thoughts in 50 years when software + hardware changes as much as it does so quickly? We need to ensure linkage of data; effective + stable response to changing environments. A semantic web to refer to and bind data to. Scientific data could be described as an algorithm for dealing with data. Structured data is a human behavior we teach machines to copy.
- Authority files support structuring of data
    - What distinguishes GND from other authority files, it is already the product of collaboration.
    - 1000s (?) of libraries contribute to it.
    - Wikipedians don't just link articles to it, but help enrich the data.
    - In past few years, strategic program + structure of GND have changed little by little.
    - Trying more integration of data not conceptually linked to publications; and reaching to communities beyond scope of libraries traditionally
    - 12 partner associations signed contract in 2017 regulating rights + duties in GND cooperative; defines workflows + standards for data quality delivered

    - Enables new partners outside of library structures to join in future
- In 2018, started pilot project - GND for C (cultural data);
    - Organization, communication, rules + data models, + infrastructure were outlined as goals for working with these records
    - Partnering with Wikimedia Germany to understand how to share information there; see blogpost on this published this morning
    - Trying to integrate all of this into an international framework
    - Can be a challenge to reach out to other communities who don't share this framework
    - There are many data models across GLAM; sometimes, it even seems like we are not speaking the same language.
    - Feels like traveling around the world without the adapters to the outlets.
    - This involves a complex change process. This project at large is focused on requirements of new customers. What about librarian world? Does it have to change? Does it want to change?
- Authority control within library world
    - Set originally for a need; to become more efficient.
    - Has emerged from long debates
    - Only natural it brought about procedures + shared rules, but also smaller groups having shortcuts + work arounds for holes or missing rules
    - System often reacts to unwritten conventions, leading to more regulations, making the system more complex + harder to become a part of
    - On a social level, causes the establishment of an expert rule. Gives power + status to those inside, something nobody likes to give away that easily.
    - Term refers to process social systems tend to go for; not only defined in system itself, but secures influences. (missed name) described this as the 'Iron Lung'
    - To study, interpret, and apply rules, it isn't just daily routine, but a considerable part of the job; and catalogers tend to have been doing this work for years. They are both masters of the topic, and mastered by it.
- Addressing open authorities
    - Try to create new roles for all stakeholders
    - Try to create new + easy to use infrastructure
    - How do we know how to do this? Can we just take it out of the box?
    - Challenge will be to not overregulate; and sharp at the edges - define clearly the minimum, what is in, what is out. It sounds easy, but is enormous task. And all about communication
    - We need to communicate. Talk to librarians, cataloguers in other domains; create occasions for encounters and debates; and make trust, by making process open, transparent, participatory. We need time + patience with each other. But mostly, we need to talk. Agree how to

regulate relevance in authority file - what should be store in authority file as opposed to other databases?
    - What is the minimum data in authority file? Have as much as possible within GND, or link it to trusted databases?
    - We need to simplify rulesets to make it more accessible and applicable to our cross-domain colleagues. Some of this may be helped with better UIs + interfaces hiding rules, but there also will come some actual simplifications
    - Need to develop a shared understanding; how do we assure quality. Quality can be controlled by technical means to a certain extent, but it also needs new contributors.
    - To work on this, GND opened meetings, reach out to new communities on social media; This is part of why Barbara gave keynotes; wants our questions + answers. An open authority control needs an open mind.
    - If you're unable to give questions now, Barbara will be here to record it and get it into their opening process.
- Questions
    - How vulnerable is an authority file (e.g. political influences)?
        - Not going fully wikidata model, but more of a contract set up where provider is reliable for quality of data provided
        - (question asker: there is still bad data [paraphrased])
        - Asks if asker can be more precise in fears
    - If we open traditional files as open as wikidata, we will go same way as all the wikipedia have gone
        - Wikidata has its own systems, way to describe world for humans + machines. Authority file does face other requirements. If connecting wikidata to other authority files, we're doing a better job than trying to get it all into one big hub. Challenge of data maintenance is what wikidata is facing, and not possible to guarantee quality. Difference between managing quality of wikipedia as opposed to wikidata; if we used wikidata to just dump our data into it but not care about its quality, this will not lead us to a quality file
        - GND working with ORCID as well to establish links between 2 systems; thinks that is a better way to do that
    - Point: Much room for layered approach; perhaps bring all these worlds together; by complimenting each other. GND has potential for strong + reliable control, but on other hand, for many institutions, it's impossible to sign contract with this on GND consortium; lots of people contributing to wikidata but not in partnership with GND.
    - Point: wikidata moves much faster to add new links, data, unlike GND. Potential for layered approach, where core authority control for German libraries is in GND; and other stuff is in wikidata; with links between the

two. Technically, we can just join these data sources because it is Linked Data, and use it as 1 large authority with different levels of control.
    - See wikidata as partners, consortium partners. Why GND is partnership with wikimedia germany to test wikibase as well as investigate connections to wikidata; evaluating wikibase before jumping into it

Lightning Talk 1: Felix Ostrowski, skohub

Slides: https://docs.google.com/presentation/d/1kCfqafXNa0ndi_-OU_bZZa6q0Q0otAhIWNWJlkYLN-s/edit?usp=sharing

- Update on talk from last year's ELAG
- Simple Knowledge Organization hub
- Building infrastructure…
    - Building repositories + people depositing into these repositories
    - Then we have crawlers that go there periodically and ask for more stuff
- Idea of SKOhub is you connect different sources that push updates into hub
    - Basically, everything centered around a topic
    - Example, I have a topic on library science, and there's new publication on topic
    - Idea is to provide a linked data notification inbox for these topics, and show I have a new publication on this topic
    - Idea is to publish + describe, instead of publish then crawl
    - Got some funding to build a prototype this year, under umbrella of OED infrastructure but more generalized
- Right now
    - test.lobid.org/gnd/4006465-7 as example
    - Added a couple of HTTP headers pointing at a service; if you have an update about this topic, put it here; and if you want to get notifications on this topic, subscribe here
    - Built based on websockets; can go there, describe it, connect to hub
    - Still a question about what the payloads will be
    - Focusing more on machine to machine communication, getting away from source centric (crawl a repository) to resource centric sharing
    - Still having a content hub
- Next steps
    - Create a stand-alone OED metadata editor
    - Can use it to describe like a youtube video
    - Apply topics from authority files + publish to them
    - Idea started as SKOS publication
- github.com/hbz/skohub-* (multiple repos there)

Lightning Talk 2: Tesseract OCR / Stefan Weil (UB Mannheim)

- OCR / Text recognition software
- Only a few programs are used for this in libraries; and fewer of those are free software
- Tesseract is one of OSS + free ones
- Available on github;
- 1 solution (does all processing steps; including language scripts support)
- Can create text, pdf output
- Has a large, well-defined user community
- Tesseract OCR News
    - First created in 1984, still going strong
    - In 2006 was taken over by Google (currently inactive)
    - New release made after this, around 2016
- UB Mannheim actively using tesseract
    - Working on project funded by german research foundation on OCR-D (ocr-d.de)
    - Starting a new project in July called OCR-DW; focus on text recognition for archives
- Brand new tesseract 4.1 ELAG edition
    - https://digi.bib.uni-mannheim.de/tesseract
    - 32 bit version should be good to go

Lightning Talk 3: (Dorian Merz)

- Question: working on libraries, archives, museums for a while now, in semantic web branch
    - Wanted to start with small survey influenced by keynote
    - Who knows about reasoning in semantic web space? (few folks); who knows what reasoning can get you (fewer); who knows how it works? (nobody)
- Speaker studied CS on descriptive logic + can describe very quickly parts of it
    - Example: person should never have more than 1 birthday
    - Computers can do this type of validation for this; at least, they can find out where it occurred; and could help a lot of cleaning up data
- So, wonders why reasoning is so seldomly used now; and would you consider having your data cleaned or proposals for cleaning your data from a reasoning or inference system in automated fashion
    - Leaves question open to you

Lightning Talk 4: Jakob on Henriette Avram

- Henriette Avram invented MARC format
- Still working a lot with MARC + cannot get around it

        - Wants to show one project he did last year; library neural network as joined their catalog with bigger cataloger; and there are new digital formats that need to be checked
        - Idea is to validate MARC format, or to look at it; what MARC is it? There are a lot of MARC dialects + variants; everyone uses some different fields in different ways
- Idea is to create a schema format: Avram specification
    - Surprised there was no machine-readable description , so made it
    - https://format.gbv.de/schema/avram/schema.json
    - Gives fields + labels information; that information is in the LoC documentation, but it is not machine readable. So why not encode this in common JSON format + give this out; can then check if MARC format conforms to list of fields
    - Can define what you know about or now; what should exist
- Easier way to say is this is our specification of MARC
    - Haven't completed it yet bc fear that if he writes this + checks against data, there will be surprises

Lightning Talk 5: Christina on Terraform & Atlantis

(notetaker gave the talk so sorry for no notes; here are some links):

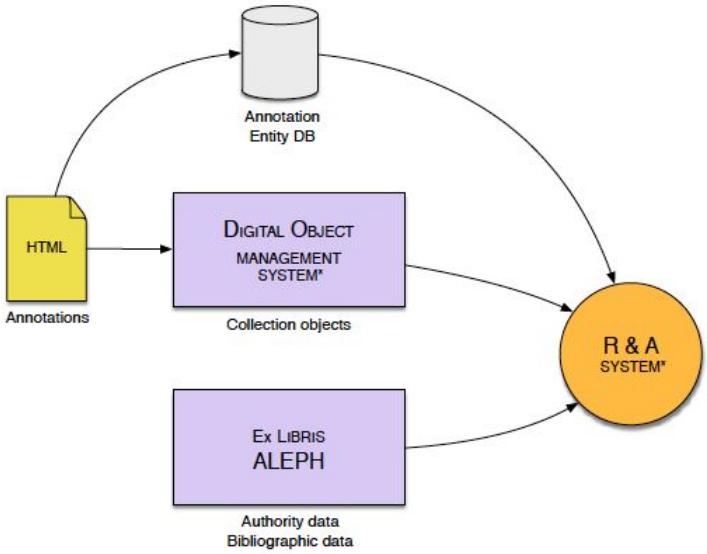
- Terraform: configuration language + set of tools from HashiCorp for configuring + provisioning infrastructure: https://www.terraform.io/
- Atlantis: small Go web app that connects to GitHub webhook for your terraform repository(ies) and runs terraform commands against your provider and state: https://www.runatlantis.io

Linked Digital Collection "Rainis and Aspazija" / Uldis Bojārs

Slides + materials: https://bit.ly/elag2019-lv-collection

- 2 Parts: Linked Digital Collection itself; then Semantic Annotation Tool that is integrated into new version
- Rainis & Aspazija
    - Important political figures of their time
    - Few years ago was 150 anniversary of them
    - Goal then to create a project to explore use of LD tech, decided to use Rainis & Aspazija as subject, good subject for this collection
    - Collection current version: runa.lnb.lv - not aimed to be comprehensive (there are lots of letters, correspondence spanning 35 years); but aimed for variety, rich multi-platform content
    - Seeing landing page, you can see the types of digital objects you can retrieve in this collection

        - There are summary pages about the people; and the organizations that provided the digital objects
            - Little bios, summaries of content, etc.
            - Different kinds of objects related to this
- Looking at platform
    - Looking at an object, see digital collection resource metadata (title, URI, link to full image, etc.) along with object itself
    - There is some Linked Data involved here as well;
    - Beyond downloading the digital object you've found, you see links to more information, additional objects, etc. in these pages (more on this later)
    - Multiple ways you can explore the collection; for example, a timeline of events
- _Linked_ Digital Collection
    - Looking at example of letters they sent to each other; their correspondence covers 75 years and covers many important parts of Latvian history
    - They've been digitized, transcribed, and were first objects they used to get to interlinked nature of the collection
    - Scrolling down, you see the textual content of the letters along with links. Links are there because the text has been annotated manually, for mentions of important entities (linked)
    - Have created collection pages for all these entities
- Where are the links in this collection?
    - Some are what you already saw; annotated documents, digital objects pointing to entities; and entity pages pointing back to digital objects that mention them
    - There are also links between objects + collections, as well as digital object management system with LD interface
    - Entity pages are simple + crude at the moment, but contains links to some external sources (e.g. VIAF, wikidata) and web sources
- Annotation set up
    - Looking for annotation tools that were easy for experts to use but also easily integrated into their system
    - In first pass, digitized objects content were HTML, and system knows how to convert links



- System
    - Uses BIBO, DC, FOAF, Schema.org vocabularies
    - Bibo:annotates for pointing from annotation to original document
    - Schema:mentions for mentions of entities in object content
    - Reuse identifiers from digital object management system, so easier to link back to source
    - Bibo:recipient for people receiving letters
- Preview of Rainis & Aspazija 2.0
    - Newly revised system, has new look + feel, more testing, improved search, with English version; new ways to look at data; responsive web; and new annotation system
    - Showing homepage with search / ways to search; seeing lists of digital objects;
    - what an entity page looks like
        - Prettier, more functional, contains more links
        - Allows you to record what links you want.
        - If wikipedia link is present, pulls in picture + abstract
    - Maps for search as well
        - Based off annotations, i.e. somewhere document mentions this location
    - Visualizations as well for entity relationships
        - Question though of how to make a useful visualization
    - Looking back at the example book page…
        - Structure of page is the same
        - See the english versions have been added
        - Annotated text part is prettier and more functional

        - Have works, persons, organizations, favorite quotations, etc.
        - Sample work is full of symbolism that princess has been sleeping on for many years; architects used that concept to create new national library building even
- Semantic Annotation Tool
    - Did not find tools that fully suited them
    - Had some idea what we are looking for
    - Worked with a private company that developed this semantic annotation tool for them
    - General requirements for this tool listed (annotations generally; be resilient; work with text)
    - Main idea for this - integrated entity database, rich annotation model, and LD interface
        - Wanted a place in entity database where they could maintain the information gathered
    - Core annotation types
        - Simple annotation: named entity mentions
        - Structural annotations: some part of text is special in some way, or role of the text (e.g. text of letter + experts commentary on letter)
        - Complex annotations: all the harder cases
    - Walk through of UI of Annotation tool
        - At least 1 annotation in grey: this is a structural annotation saying this is a quotation
        - Multiple colors for different types
        - Drop downs + entries for filling in annotation
        - Annotations can show some information for linked to entities as well
        - Why text recognition doesn't work well here: one word in text is used 4 times with 4 completely different meanings; machine can't determine
    - Entity hub
        - Every entity has a class
        - Can all links to entities of multiple types
        - Want to make open, but have some concerns about GDPR at present
        - Annotation tool is tool for creating your knowledge base
        - Integrated authority record database as well, to simplify creation of new entities in bib and auth records
    - Can see in UI the links, the metadata, and RDF versions of the data
    - Entities types described go across a range
    - Data Access
        - Annotator's server API for JSON (internal)

        - There is ability to do data export + publishing, self-contained web apps with annotation + entity information provide these
        - And LD interface for entities, but currently not public (see GDPR thing above)
- Conclusion
    - Cool tool developed based on their needs;
    - Can be easily internationalized, used with other languages
    - Integrated in their collections
    - Next steps are to make use of this gathered information
    - Interested in collaboration as well
- Questions
    - What was the experts' reactions to this tool? Did they adapt easily?
        - There were different types of experts involved
        - One type of expert are those doing the annotation + data entry for the system, quite a complex task with 2-3 people doing this annotation work
        - Idea of linking came naturally to them, no obstacles there
        - But in first iteration of this collection, there were difficulties building it; building it with internal resources + no project financing available. Getting Linked Data ideas to those dev teams was a bit of a challenge, given library dev being focused previously on maintenance.
    - Use case of using composite annotations?
        - Structural annotations used quite a lot, because can then filter in annotation tool
        - Composite not used as much, but allow to describe richer structural content of document
        - Example: event - can say when it happened, where, why, etc.
        - Another example: in compendium to works, information about literary work, can say when it was started, published, etc.

Bauhaus Open Archive / Dr. Esther Cleven, Felix Ostrowski

- Introduction of curator at Bauhaus archive; partly responsible for digitization
- Presenting project started last year - introduce museum and the problems around this project
    - Bauhaus archiv is museum around the corner
    - Opened in 1979 after moving to Berlin in 1971
    - Building designed by one of founders of bauhaus
    - Founded as archive in 1960s in another city by curator + art historian researching bauhaus and found little resources on subject
    - Bauhaus founded in 1919 just before Weimar Republic
        - In building, stayed there until 1925, until pushed out of Weimar due to Nazi's foot in the door

        - Decided to go to a different region, and got funding to get a new iconic bauhaus building
        - Couldn't stay too long due to third reich, moved to Berlin for little more than a year as private institution; then closed by Nazis
    - 100 year anniversary this year of bauhaus; have lots of plans around this
        - Building from 1979 had little space; had extension plans for it
        - Unluckily, though, closed in year of anniversary
        - So anyone using heritage they care for, the answer is no; all they can do is buy photographs, and even that is difficult
        - See diversity of materials involved; they are tiny but heavily used by anybody working on bauhaus
        - People are used to analog materials there
- Impressions of Bauhaus Archive Collections
    - Biggest + oldest collection of bauhaus materials
    - Lots of estates, personal materials, documents, etc.
    - Have cartography, architectural designs, models, teaching notes, drawings, etc
    - Have a library as well coming from estates + special collections
    - Not the only collection on bauhaus; but because bauhaus closed 2 times, there are only bits of an official archive in terms of administration
- So; big push to try to become digital given the above; start with basics and work on it
    - Have had a small collection online for a few years; so wanted to start there and build it out
    - They need smart affordable + sustainable solutions
    - They want the online resources to correspond with service orientation in their analog realm
    - Have different databases for different resource types; and archives don't even have a resources database currently; want an integrated interface
- Working in museum for 25 years now, experiences from that:
    - Not helpful to integrate external databases
    - Big plans don't work in museum; need iterative processes, and use what we have
    - Have to learn as an organization by just working on it
- Pilot Digital Project:
    - Started digitizing tiny part of Gropius archives - started with correspondence of certain time range; along with 5 thousand photographs + newspaper clippings from Bauhaus era in Weimar
    - Working with Museum Claus (sp?) so made in that
    - Funded by DigiS ; digitize images from objects now;
    - Also want to improve permanent IT infrastructure while working through this
- Digital is the new normal

    - Library vs Museum difference there
    - Scientific library works with inherently digital material + distribute copies online; museums have a different understanding of copies + sharing
    - Change in culture is a bit bigger even in a museum than a library
    - Digitization work has to be first class citizens; important to digitize + then put online, distribute them - has to be as important as getting people to visit your physical collection
    - “Beauty of translating CH into digital space because for young people, something not online doesn’t exist” (paraphrased quote from ?? on twitter)
- How do we talk about stuff
    - Identifiers in museum have roots in inventory books
    - Not great to translate to web; some resources have multiple inventory numbers; others, known
    - Let's start using identifier of IT system or database; there are a lot of workflows were we juggle with inventory numbers, so thinking from a digital system perspective, flip this
- Another difference: libraries primarily deliver something to someone outside; museum services tend to point inward
    - No idea really of OPAC within museum
    - Why start pointing outward then? To be visible, and to put digital objects in context
    - Referring to authority data as a first step; which hasn’t been done yet in that work. So integrated Getty AAT
    - Have own classification, but can map to external vocab
    - Institution has 0 IT; librarians take care of desktop computers, that’s all there is in house
    - So used these little fields in existing system + enter URI in description
    - Why not import entire AAT? It is huge + not a lot of needed
    - Someone knowledgeable on system can add the links in there; no automation here
- Second step, integrated GND into this. Limited themselves to people right now, folks around Gropius; but hope to follow up with corporate and territorial entities
    - Use a multimedia module in their system to attach multimedia objects to entries; you can add something for website - like link to external GND by adding the http URI in that module
    - Needed to automate that a bit as well; this collection grows at a much slower place; so less effort to rerun automated enhancement often;
    - How do they automate? They export a CSV, add what is necessary (first name, last name, birth or death date)
    - Then used GND reconciliation for OpenRefine; send them text strings + get back matches
    - They then import CSV via MySQL back into system; part that fields weird for speaker, feels a bit like heart surgery

        - Do pull data in; they use Calliope; that is used to catalog stuff. Can't afford or support a system specialized for archive stuff; Calliope is just the right system for archive hierarchical structure, unlike museums
        - Initially, one time import based on CSV; but this synchronization task has to be repeated again and again
            - Pull live data from MODS interface + basically what archival object as digital object into system
- Exporting data - open-archive.bauhaus.de
    - Currently being published using (missed tool)
    - Synchronization process - someone starts something and a bunch of terminal windows pop up, you leave it over night and hope it runs
    - Trying to think about new solutions for this
    - Another way they are exposing data is to get through LIDO, German National Digital Library, and Europeana
    - Trying to get 3 data formats planned; with everything data being open; and digitized stuff harder to make open;
- Simple overview of their pipeline
    - Museum => LIDO
    - Archive => MODS
    - Library => MARCXML
    - All go to Catmandu
    - Goes to JSON in ElasticSearch
    - Use ElasticSEarch as a 'cheap API'
    - Lots of bash scripts with catmandu, LIDO to JSON, using jq as a JSON processing
    - About half a day of work got to the start of a API and a UI (without styling just yet)
- Sometimes I wish I had MARC by my side
    - Feels weird going to MySQL to edit; having an exchange format would be nicer
    - Vendor lock in inhibits innovation
    - And are consultants the solution?
        - Iteratively is good, but there was so much knowledge loss; as a truly digital collection, you will ultimately need someone in house
- Question:
    - Opening up of digital resources?
        - Bauhaus has limitation of being relatively recent, so laws protecting publishing that data is complicated; also need the publishing approach that is maintainable

Data processing of ILS data to facilitate a new discovery layer for the German Literature Archive / Thomas Meyer, Felix Lohmeier

- Slides: https://doi.org/10.5281/zenodo.2678112

- GitHub repos mentioned in the slides
    - Openrefine-client (for scripting OpenRefine): https://github.com/openculturereconsulting/openrefine-client
    - Fork of OpenRefine to increase performance when joining projects with multiple-value cells: https://github.com/openculturereconsulting/OpenRefine
    - file-based OAI-PMH data provider: https://github.com/openculturereconsulting/oai_pmh
- Have been working with their OPAC for 20 years
    - Time to modernize
    - Current project started in 2016, was a prototype
    - Currently looking at this project starting in 2017; still in beta
- When starting with project, had a lot of expectations
    - Still in a state where they expect all requirements to be met by project
- Looking at what we have + what users are researching, see its focused on many authors, and brings us back to authority files
    - Authority files played a major role at German Literature Archives for 20 years or more
    - They would like to address cataloging of bibs, auths, etc in one system
    - Using GND numbers in their records
    - Have all their resources connected to this number - photographs, letters, books, objects like death masks
    - Getting all this in one system and linked to each other
- ILS
    - aStec Kallias - custom ILS for them that has all holdings together in 1 system
    - Short glimpse of current implementation
    - Can start with a search, get objects that people expect
    - Holdings have a very detailed level; archivists + librarians putting a lot of work in describing these holdings
    - When designing the catalog, decided that common formats like BIBFRAME or CIDOC-CRM didn't really address their detail level or broadness of holdings
        - Comment after the presentation: This could be done of course by enhancing e.g. CIDOC-CRM with own vocabulary. Another argument is that the internal data model is well known in the organization and will stay for many years because there are no plans to change the ILS. To support reuse of the data we provide EAD-XML (and other formats in the future).
    - Kallias system has 1200 fields, with nearly 500 being relevant in the project, so couldn't find a data model that could address all this
    - Implemented a custom non public format, following aDIS/BMS ; and reason the full code of the project has not been published open source yet

- Goals for project:
    - Open Source throughout
    - Modular
    - Able to add customizations later
    - Important to reuse data in other environments like digital projects
- Overview of architecture
    - Lefthand side is ILS ; able to do token exports of all holdings
    - Righthand side is TYPO3 extension “find” which is used for the frontend
    - Are doing exports to other systems currently; EAD exports delivered to Kalliope
    - felixlohmeier.de/dla/systemarchitektur.html
- Choose OpenRefine in prototype of project;
    - Selection of tools evolved in main projects; but OpenRefine stayed in place for pipeline
    - Preprocessing of data is done with Python Pandas (data analysis library)
    - Source data (token exports) contains codes; this preprocessing can do things like code to human readable labels;
    - Mappings are defined in CSV file stored in Github so staff of Archive can edit these via Github
    - Earlier stage of project for lookups and groupings was using Openrefine; switched that to Pandas as this was expensive in OR
    - Translated to a tabular data model for loading into OR
    - Multiple values stored in one cell separated by special character that doesn’t occur in data
    - A feature of Pandas out of the box is ability to do some nice statistics; did some early data analysis as a part of this
    - Directly import CSV files from Panda into OR
    - OR can be managed via CLI through client libraries
        - They forked and enhanced the python client library to use it: https://github.com/openculturecons ulting/openrefine-client
    - They use OR client to load file; apply the transformation mappings file; automated process;
- Source data contains internal keys to external authorities
    - Don’t want only the ID, but also the string of the name
    - Can see a library subset there with several ids of the authors
    - Openrefine doesn't support multi-value cells at the moment you can split columns beforehand but expensive; so they created a fork of OpenRefine that can do splitting + joining in same transformation; available on github https://github.com/openculturecons ulting/OpenRefine
    - Then want to provide more information for the result list
    - Another feature of this lookup, they can facet on top result records for each search query
- Facets + filters

    - Use custom rules in each subset; example, converting dates to a Solr date range field (adding ` TO ` and square brackets)
    - Complex rules for these objects allow features like a time range slider (day resolution)
- Detail pages
    - Can combine source data fields by themselves
    - Preprocessing some fields
        - E.g. differentiating authors + contributors
    - Requires complex OR transformations with for loops etc.
    - Have one field with IDs and another with the corresponding string that is index related; hacky but it works
- Authority data usage
    - Exploring what data they can make use of
    - One of the experiments is a ranking feature
    - When user opens authority data search tab they see the authority records ranked by referrals from other records
    - Extract entity_ids, into list then into project
    - In project, give entity counts,
    - Then can use this precalculated entity score
    - All using OR before hand
- Lots of GND IDs in source data
    - Using Wikidata and lobid-GND reconciliation services for OpenRefine for authority lookups
- Going back to big picture… after pandas + OR
    - result: CSV files
    - Load CSV files into Solr with customizations to solr around relevancy, sorting, etc.
    - OR running permanently in employee network to provide access to the live data in OR for colleagues. Data is used for other in-house purposes
    - Also create EAD/XML format that they serve with a file-based OAI PMH data provider https://github.com/openculturecconsulting/oai_pmh
- Bash scripts + automation
    - Can use other tools for this work as well
    - Full process can take 5 hours
    - Openrefine works like in memory database; works fine but requires lots of RAM
- Frontend is built with TYPO3 extension ‘find’
    - https://github.com/subugoe/typo3-find
    - Custom templates, etc.
    - Skipping over due to time
- Live demo
    - http://www-test.dla-marbach.de/ (available temporarily - until 12.05.2019)
    - Showing type-ahead in system

        - See 3 different media objects show up
        - One of these records selected, click search
        - Then searches in backend by ID and not string
- Lessons Learned
    - See slide
    - Would like to stress one point - although we had high effort to coordinate all this, still see that in house data literacy has really increased with this project
    - Benefit is not only having new catalog, but also having learned + talked with each other a lot on this work, internally + with partners
- Questions
    - Why OpenRefine in the workflow etc?
        - Started with OpenRefine in the prototype project. Workflow easy to adapt. If they encounter blocker (that can't be implemented with OpenRefine), they would change (possible due to file-based bash-scripted workflow)

Technical filtering of relevant articles and enrichment of data - a new service of KOBV / Nicole Heidingsfelder, Oliver Kant

- Nicole is Project coordinator for KOBV
- KOBV
    - Hosted discovery system
    - Idea for new data enrichment service
    - Past year, discussing building an index to replace Primo Central
    - Many libraries face problem of lacking data sources
    - Scientists + students then rely on databases
    - Want to offer customer service to combine journals + related articles + make them searchable
    - Provide licensing work for this service as well
- CrossRef
    - International, non-profit project for scientific publishers
    - Goal is mint as many DOIs + link them as possible
    - Collect other metadata for the resources as well
    - Possibility for direct access to articles appearing in journals etc.
    - Number of objects registered has steadily increased
    - Almost 100 million registered objects in 2018
    - From many different countries
    - Almost 77 million journal articles registered; other materials can be book related, conference papers, components, datasets etc.
    - Data can be accessed via API as well
- Over time, types of content published has expanded
    - Publishers can also store ORCID IDs, abstracts, and other data now
    - Nearly 2/3rds of documents contain full text links

    - And ⅓ of documents have licensing information
- CrossRef in ALBERT (discovery system)
    - ALBERT = in house development library system
    - No general article index at present
    - CrossRef is imported in there; have a reusable Solr-index maintained by KOBV
    - They have nationally licenced journals in there as well
    - No access to older articles at moment (which is a problem)
    - They want an open and reusable solr index with ALBERT data, can show items + data licensing
- CrossRef potential for libraries
    - You get a daily updated article indexed for licensed journals, with links to existing journals - they're planning to use this by end of 2019
    - There are numerous data displays from CrossRef; & you can enrich that data returned with things like abstracts
    - And CrossRef uses ORCIDs, so makes faculty more visible
    - Building citation network is a next step after integrating CrossRef data into the library systems
- CrossRef & DataCite
    - Want to integrate datacite data into their index by end of 2019;
    - Allows automatic linking of content of datacite;
    - Datacite + crossref also are partners
- Screenshots of work in progress
    - At moment, references / citing references are prototyped; there is the references + citations provided by publishers then used in the discovery system to add a show links section in discovery system
- Crossref + technical slides
    - Crossref-test.kobv.de (access is currently restricted)
    - Discovery system work in progress
    - About 70 million articles in the index (journal articles)
    - There are also EBZ (electronic journals library in germany) subject classifications + disciplines included
    - Search clients can use that index and the cross-ref index for lookups
- Technical architecture
    - Developed backend system using modules for data harvesting or multiple formats (java, can handle MARC-XML, DC, MAB, JSON)
    - CrossRef > kobv harvester > crossref DOI, ISSN, Data, Last update in database
    - DataCite > etc > identifiers + data
    - Unpaywall > etc > identifiers + data
    - Need the source identifiers and dates
- Backend databases then go into a data preparation transformation
    - Transform CrossRef JSON to MARC-XML

    - Choose MARC because of experience with it in library + spec is granular enough
    - Merge via DOI or ISSN
    - Transform process is enriched by other data sources mentioned
    - At the end of the process, MARC-XML created by the indexer + imported into Solr index
    - Do atomic updates against Solr for holdings, disciplines, OA (open access URLs) and datacite links
- Front end
    - Backend provides data to front end via KOBV backend to Solr
    - This is managed currently by Jenkins server running jobs
    - In practice, don't do atomic updates *currently*, so it takes more time to process + is a disadvantage
    - Want to adapt indexing engine for atomic update (i.e. only changing individual fields at a time, like holdings)
    - In summary: trying not to reinvent the wheel technologically, and have proven this work can be done in principle. Upcoming challenge is implementation of partial + daily updates;
- Conclusion
    - Contract expected by end of 2019
    - Reusability of data is ensured by MARC/XML format + possibly a base export function
    - By 2020 at latest, should be moving all of this into production operation
- Demo
    - Searching on test index
    - On left side, filters + settings
    - Have 3 partner instances (libraries) available at top
    - Facets also include keywords, materials, publishers + licensing (OA or licensed)
    - View page for a resource has linking mechanisms in top right corner - click on DOI for example to get to article
        - Can in 2 clicks, get to article or otherwise
    - Showed lookup by identifier as well
- Questions
    - Why develop this on your own and not buy anything?
        - Question of money. Things are expensive, and try not to use Primo Central due to costs. Possibility to use CrossRef data, so they thought it was the best option. They can enrich the data with information as well in this system.
    - Quality of CrossRef Metadata? Seeing much noise, duplicates, etc.?
        - Speaker not personally looking at the quality; colleagues check articles + said it was good
    - Complexity of backend - what is needed to run this thing?

        - Have built / organized the code over last 10-12 years; includes different modules
            - E.g. hierarchy analyzer for checking if parent / child relationships
            - Duplication checks
        - Machine hosted in house; infrastructure isn't so big
        - Happy to talk more about this at coffee break

Library Logistics Optimization System / Petros-Alexis Kofakis, Katerina Marinagi, Michalis Gerolimos, Eftichia Vraimaki

- Department of Logistics Management
    - Founded in Sept 2006
    - Located in central Greece
    - Wanted to talk about project done in cooperation with national library of Greece
    - National library of Greece moved from historic building in center of Athens to a new building; nearly 24000 meters squared
    - Transition + planning took 2 years
    - Transit itself took 4 months, completed last year
- Facts
    - Estimated 100 users a day in the library, when library was fully operational
    - Normally readers ask for 3-10 items; national library has mostly old books + materials from early dates of republic
    - There are 6 reading rooms with capacity of 236 seats + 6 service points
- Problem:
    - Because of structure of library, which is not like a warehouse, but instead restricted structure, the daily operation to get items to + from cells is labor intensive; no robotic system or whatever
    - Started to discuss how to solve this problem, and the library had 2 major requirements:
        - Core system should be based exclusively on OSS (not a problem; mentality + experience already there; this was not a question of money for library, but avoiding vendor lock-in or limited solutions)
        - Also, should be integrated with Koha, their ILS
    - Problem:
        - Standard planning problems
        - Tried to optimize the goal + minimize cost, with limited resources (employees, assets, time, money)
        - Under constraints also of open hours and having the appropriate staff available to move special resources, handle materials, etc.
- Solution
    - Model process as pickup + delivery logistics problem

- Capacitated Vehicle Routing Problem approach used - minimized distance walked as a time function
- Have all the components for this already in OSS
    - Koha was one part, and fully expandable
    - VRP solutions
    - Database systems
    - Software for User Interfaces
    - And a Mapping solution
- Architecture overview with software components
    - Use AutoCAD + openstreetmap to design maps for routing to then be optimized
    - Map this to an indoor environment of the building, stored in postgres
    - Koha on other hand, surrounded by boxes that solve the rest of the problems
- 4 levels to the library
    - Have assigned zones to a person, and those are ‘pick’ zones
    - Consolidate buffers for retrieved resources
    - Then split resources consolidation according to service desk where it will be prepared for visitor
- Pick sequence
    - System creates list of items to be retrieved
    - List of items grouped by pick zone displayed
    - For each pick zone, VRP solution is calculated based on avail staff
    - List of items sort
    - Staff retrieves
    - Aggregated in buffer zone
    - Grouped by destination
    - Groups transfers to service points
    - Etc.
- Using OpenStreetMap tools - like JOSM - for creating the building ground plan
    - Correct the ground map with architectural designs, but designs don’t correspond exactly with transport lines + then make human friendly (i.e. take left at this number etc.)
    - Make this ground plan more city street like
    - Shelf location encoding problems however
        - They were trying to break apart into building, level, area (roughly 36 different rooms), block (in some areas), shelf number
        - But don’t always have same number of shelves, so used geocoding (process of transforming physical address description to location on earth’s surface)

    - Location => map => navigation plan
        - Koha points to book location code, which is now a map address, with map coordinates, that get applied to a distance matrix
        - Routing, and tools like OSRM link the optimal path between two points (like google map); but this isn't solution, we need to find optimal sequence at this point
        - Still working with google optimization tools on this + organization service
        - That solution generated is then displayed to user
        - Vehicle Routing Problem (VRP) is used to find the best sequence or combination of available resources + shortest path
        - OSRM very fast, written in C++, routing engine for shortest paths
        - Use leaflet for user interface to show the map to the user
        - Vroom-project.org - open source optimization enginer in C++14
        - Vroom API
            - Gives matrix, vehicle locations, skills for routing optimization
    - Final result:
        - Optimized route provided to user - a sequence of how + where to collect at points
        - Can should 'vehicle' (trolley) information + points
- Project Extensions + Conclusions
    - Can have a solution based exclusively on OSS
    - Is solution to real problem they have; speaker even had to spend 2 weeks just getting himself oriented in the new library
    - Interoperable with Koha
    - Extension ideas
        - use indoor position with beacons (so better guidance)
        - Integration with other NLG premises + locations
        - And maybe indoor tracking using passive UHF (?) RFID
- Video demo walking through web UI with routes (Vroom output)
    - Currently used through desktop; want eventually to have these mounted on trolleys
- Questions
    - How many of your folks are using the application? Is it well received?
        - Under development, so only used by 2 librarians working with dev team
        - Making random simulations of users + discovery errors in cataloging
        - Of course it will be used, though - there isn't really any other way to get around this particular structure. Very nice from architectural point of view but not very convenient

- A lot of slides have boxes, and wonder if in workflow are you moving boxes?
    - No. Boxes are only used in final step for service desk
    - Probably saw Journals. Don't have same addresses for those. Just used on shelf for pick points; not moving themselves

Overcoming document delivery issues in WorldCat discovery / Peter van Boheemen

- Get It from WAgeningen University & Research Library
- Questions for audience:
    - what libraries are implementing WorldCat Discovery (few)
    - Who is managing deployment services (fewer)
    - Who is writing OPAC or web catalog (less)
    - Who has patrons who use library catalog to find documents (few)
- User wants something… how do they get it? Where do they find it?
    - Not necessarily in UI or library catalog
    - Presentation has example user cases of user finding some reference in a bibliography, for example; like a result set of a search on web of science
    - Showing example of embedded link resolver in item page; in link resolver, you see both paper publication in web of science as well as requests for asking for a copy;
    - Fulfillment options can exist even when they don't' have a publication
        - Can try to get it through ILL or other
    - Makes fulfillment options the same within catalog, adding those fulfillment buttons
- Proposed Order workflow
    - Stored in local order system
    - Order system contained imported ILL requests
    - Locally own materials to workflow in place
    - Remote through another workflow
- Change of the above however:
    - Dutch universities moved from national ILL system to worldshare ILL
    - OCLC introduced Tipasa for handling end user ILL requests
    - Tipasa promised to support local delivery and ILL to this integrated workflow
    - Wanted to integrate order workflow + holds in WMS better
    - Working with Delft University library to see if can benefit from a shared workflow for the above, to help return results + resources when staff is oversubscribed in one place
    - However, Tipasa + Worldcat ILL features are services in WorldCat discovery only
    - Local copy requests appear as ILL requests to the user
    - Requests services cannot be tailored by user type

    - Not want they wanted
- Solution
    - Kept the services in the link resolved
    - Did not change their get it button philosophy
    - Instead, went to work with worldcat APIs
- Examples walk through
    - In item page, there's a link to recent articles in the journal, that is pulling from ?? API
    - Then also a My Library link that uses Worldcat knowledge base API to find holdings
    - Use a few other APIs (Internet Archive, Local Repository) to see if something available there as well
    - Trying to find more APIs to integrate
    - Service that provides information of hard copies on shelves is based also on WorldCat Availability API, show if they are on loan or not
    - Looking at the catalog itself, you can see the links also does some holdings analysis to make sure there is a better fulfillment response (i.e. don't return journals that don't have the article or issue in question)
    - User sensitive: user is checked against a local user stored, used to synchronise staff or students against a local ILS
        - Means you can check type of users and show only services available to them
- Submit ILL request
    - Even though Tipasa knows which user is requesting there is no way to tailor request form to the user
    - Usually the user is a staff member + would only mention that there are no costs
- Interface in WorldShare
    - there are 3 queues: Borrowing, Lending, Document Delivery
    - In the discovery layer, have the ability to request a book via the link that is generated against WMS hold request; this is a much easier approach than the above interfaces
    - Logged in as a student, shoulds where the student can get the book themselves, since only faculty have the pick up + delivery services
- Hope of this presentation
    - Open URL is much more than a means to get you the appropriate e-version of a publication
    - The Open URL resolver is under use; gives you ways to access APIs for more services
    - Lets you better blend services into your discovery tool
- Questions

- Comment: Providing a service for print + electronic materials, using often to guide other systems that guide users to different copies. Maybe 150 systems in Germany who incorporate of systems.
    - Next challenge is to just appear OpenURL labels + bibliographies; you want to appear these links on any webpage or service that describes resources; need more work then on plugins.

Workshop Day 2: Apache Nifi automating data flows between software systems

- Groovy Example
    - https://lib.ugent.be/download/librecat/nifi/ITextGroovyExample.xml
    - https://lib.ugent.be/download/librecat/nifi/itext5-itextpdf-5.5.12.jar
- Walking through example flow from yesterday
    - Seeing isNull() function, way to access data from flow files
    - Also possible to set some global attributes;
        - right click on web board
        - Configure processor
        - Add variables there
- Looking at Provenance Events, you can see PDFs stored, processed, etc. by the example using Groovy processing

Day 3 - May 10

Lightning Talk 1: Apache Flink

Slides: http://swissbib.org/doc/elag/p.pdf

- What is Flink?
    - Apache project since 2014
    - Originally developed in Berlin
    - Until 2018, mainly developed by Berlin-based company ververica.com
    - Recently sold to Alibaba for a lot of money
- If scientific libraries want to remain visible providers of digital information in 10 years time, they need to use tools + procedures mostly only used by large companies now, especially for data sovereignty.
- Current swissbib Flink use case
    - Want to combine with kafka work they're doing already (larger presentation this afternoon)
    - Data hub provides MARC data + is pushed into kafka; data based services (like search engine) pull from this kafka topic using flink technology
    - Flink takes data, transforms it, pushes it back to kafka cluster

        - Then microservice takes that transformed data from kafka + pushes into solr
        - Use metafacture + other local tools in combination with Flink technology
- Code deep dive
    - Configure main program
    - That is then distributed onto cluster with some configuration data
    - Then that data source is put into kafka cluster
    - Then can set number of tasks that should be processed in parallel (for development for them, is 1, but in theory could be 3 to 1000 like Google)
    - Transformation function
        - Using metafacture framework
        - Data comes in (MARC) then pushed into metafacture pipeline
        - Then given back to Flink class
        - https://github.com/swissbib

Lightning Talk 2: ROSI / TIB

- ROSI = reference implementation for open scientometric indicators
- Building TIB VIVO information system
    - Hosting VIVO instance
    - Person example shown
- Scientometric indicators
    - Idea behind this is if you have a topic + you want to find an expert on it, you want targeted help via indicators or metrics
    - Look at publication numbers or otherwise of person
    - See if person is active on wikipedia, other places (many such metrics)
- Already few providers providing metrics for this system
    - Building prototype on only open data sources, with some customization of data sources themselves
    - Prototype of customizable data based out of their VIVO implementation
    - Started some data sources (registry of scientometric data sources), looking at data quality + if APIs exist)
- Doing some interviews + workshops about how to use ROSI
- Call for others with data to consider adding it to their registry
- https://labs.tib.eu/rosi

Lightning Talk 3: 20th Century Press Archives goes Wikidata / Joachim Neubert

- 20th Century press archive
    - Largest public archive of newspaper clippings
    - Collected across the 20th century
    - Have over 1500 source newspapers
    - 1 million documents online
    - Digitized in 2004-2007

    - 25000 thematic dossiers across range of subjects
- Specialized application used for access
    - Is outdated + expensive to maintain
    - How to make this sustainable without replacing application with something similar customized + special developed
- Wikidata mappings
    - Matched metadata with Wikidata using their tools
    - Based on mapping, other data can be added
    - Over 90% of persons are already linked
    - Missing wikidata items can be added automatically
- Proof of Concept example of access to this content via wikidata
    - Example: Map of Economists
    - Use wikidata to document every step of the way, including things like publish to wikidata workflows
- Invitation to look at wiki project - Wikidata:WikiProject 20th Century Press Archives
    - https://wikidata.org/wiki/Wikidata:Wiki/Project_20th_Century_Press_Archives (need to fix link)

Lightning Talk 4: Robot driven library maintenance / Stefan Weil

- Mannheim University Library
    - Work done by graduate students in CS / IT
    - Made a video also shown to the audience
- Robot scans with optic camera the book shelf numbers
    - Processes images with tesseract
    - Takes images; extracts color to find shelf label sticker & crop to that
    - Reads call number / shelf number + generates report
    - Stores results in database, noting if there are missing books
    - Compares results with data from Alma for reporting as well
- “Robot driven library maintenance”

Lightning Talk 5: MARC::Schema & Catmandu::Validator

- Jakob Voss talked yesterday on Avram specification
    - Avram schema for MARC21 developed by Péter Kiraly
    - Based on MARC documentation from library of congress
    - In Json, about 23000 lines
    - Available online at github (pkiraly)
- Péter used schema to do a metadata quality assessment
    - Validating about 100 million MARC records (a bunch of open data sets)
    - Published first results
    - Biggest issue for data quality is locally defined MARC fields
- Based on schema, created 2 tools
    - MARC::Schema creates CLI tool called marcvalidate

        - Then MARC data is processed against schema + output is problems with MARC
        - You can provide custom schema or give enhanced schema for local fields as well
    - Catmandu::Validator::MARC uses the above
        - Catmandu convert MARC to JSON --fix validate…
        - This validates MARC data and gives back errors in CLI
        - Can also use schema with condition called 'select valid(.MARC) and this will filter out invalid MARC records for processing

Link: http://jorol.de/talks/2019-ELAG/lightning.html

Lightning Talk 6: Publishing Metadata Provenance / Jana Henschke

- Slides: https://www.slideshare.net/JanaHentschke/publishing-metadata-provenance
- Taken for granted today that metadata is coming from catalog
- However, data comes from multiple other places as well
    - Machine generated
    - Derived from mappings
    - Taken from vendors
- Have need now to exchange where data actually comes from
    - Example: users of metadata should be able to establish if subject heading assigned to item by cataloger, machine process, other
- MARC bib field 883 for provenance
    - Uncertain if people publish it or use it
    - Only design for long intellectual processes
- For RDF data
    - Used PROV-O (Prov ontology)
    - Using qualified relations design pattern
    - And started to describe + identify new entities in Prov Activity & Prov Plans
- Overview of data model
- Example of data in turtle
- Data dumps available at https://data.dnb.de/opendata
    - Regularly updated with provenance data
    - Keeping separate from regular data as it is a prototype, waiting to see if someone is interested in it
    - Have wikipage describing schedule + details of plan
    - Interested in feedback + exchange

Workshop Report

Nifi Group

- Day 1: went over context of tool, how to install on our own laptops, went through 2 examples, starting with a Hello World example
- Day 2: discussed how we could use it in our libraries, IT departments, easy to use (or not)
- Screenshots of Nifi tool
    - Idea behind it is powerful + reliable tool
    - Process + distribute your data
    - Core part is processors in GUI, which do something with data
    - Have ability to stop + start processors within your workflows
    - Hello World example
        - Process generates a “flow file”
        - Showed state of resource in process
        - Many processors available for running different tools or functions
        - Nifi documentation on this
        - Process Groups can create set of processors for more complex workflows
        - Operate box for running workflows, play stop, import workflows, etc
- Patrick prepared templates for the group to jump into nifi
    - Template is ready to run process groups
    - Overview of processor configuration
    - GUI + clicking isn’t for all (some prefer keyboards, typing)
    - Properties tab on a processor shows how you put in information
    - Properties can be complex, but the setup is similar across processors
    - Has a scheduler tab for running processors
    - Bit of a learning curve in getting started
- Summary
    - Worth it to take a look at Nifi; watch a recommended conference talk video, go tutorials
    - Alternative tool is Apache Airflow
    - GUI can be helpful
        - But is it really self-explaining?
        - When you start with nifi, it's hard to find errors
    - Ghent may use it for production
    - Keep in mind: ‘there is no such thing as a free lunch’

IIIF Workshop Report

- Talk about how IIIF can be implemented, how source data can be converted, what source data can be used
- Quite a few people leading workshop from different institutions

- Schedule
    - Introductions
    - Introductions to IIIF
    - Did some Demos
    - Next day did small talks
    - Then world cafe (breaking into small groups)
- Going over concepts of IIIF
- Demos:
    - IIIF Viewers + their abilities
    - How to pull data into IIIF
        - Showed how images from different repositories can be combined
    - Annotations usage
    - Other uses of IIIF
- Discussions
    - Use cases
    - Intentions
    - What people can imagine doing with it
    - Checked out more demos + installations
        - MMMONK from Belgium
        - Software that serves IIIF Manifest from METS/MODS data https://demetsiiify.jbaiter.de https://github.com/jbaiter/demetsiiify
        - Other tools
        - Mirador 3 rewrite discussed, done by collaboration + written in React.js
    - What could go wrong?
    - World Café
        - Data conversion
        - PDFs
        - v2 / v3 parallel support
        - Servers & software
        - Data modeling

AI (Artificial Intelligence)

- Day 1: Did live coding on a sentiment analysis tool
    - Point was to demonstrate relative ease with which you could build such a tool using Python + python libraries
    - Worked with Jupyter notebooks + cloud-based programming environment
    - Started with some imdb sourced prepared data
    - Computation performed used in things like spam filters; can also be used in more complex processing as well
    - Example of running jupyter notebook
    - Did also mini, informal survey of participants
        - Most had coding experience / technical background

        - AI is technology requiring careful management - most agreed
        - Other questions on concerns of how to use (or not) AI
- Day 2:
    - What are some possible applications for using machine learning & AI
        - Indexing + tagging classifying of free text
        - Suggestions recommendations collections development
        - Fraud detection
        - Cataloging error detection
        - Better ocr
        - Chatbots
        - Deduplications in database records
        - Automated tagging of content
        - Extraction of data from images
        - Recommendation engines in discovery systems
        - Plagiarism detection, etc
    - discuss level of impact of AI in libraries
    - AI threats to libraries
        - Making some jobs obsolete
        - Funders getting idea that libraries are unnecessary
        - Means of censorship of content
        - Misuse of this data by governments
    - AI impact to libraries assessment
        - 2 groups said moderate
        - 2 groups said major (critical mass of change of types of work done by libraries)

Library in Patron's Workflow

- Assuming user is not in the library; not even on library website; library should be where user is + not other way around
- Overview of discussions had within group
    - Topics like how to help, when to offer help or not, browser extensions + risks of those

Using COUNTER 5 to track usage of IIIF resources / Dries Moreels

- Talking about work at Ghent University working with IIIF images
    - Publishing images through nice API, but how much are they getting used?
    - Can we track who (people, machines) using these images?
    - UGhent publishing 1,008,131 images in IIIF + still growing
    - More than just publishing JPEGs on the web, but also metadata, other pieces of information
    - Standardized way of usage reporting isn't the focus of IIIF, so asking how should we moderate it

    - This is becoming even more urgent not since starting small service for colleagues at other institutions too small to use IIIF, doing open IIIF hosting on the web, so question of reporting on how images are used is coming up
- Ongoing work
    - Looking for feedback on data modeling + putting this data into COUNTER
    - COUNTER has been around for a while; what we ask publishers to provide data as for reporting on usage
    - Publishers are switching from R4 to R5 reporting at the moment
    - Good opportunity here for this use case; also looking to use the momentum behind this change going on
    - Also, getting these numbers can help show importance of IIIF by showing usage to others
- ERM system used currently
    - Tracking who uses what
    - Very traditional
- Differences between COUNTER 4 + COUNTER 5
    - Reports we all know has become a bit of a zoo; each report has some of its own peculiarities
    - Release 5 pulls this together by putting out there an idea of a Master Report
    - Master Report come with idea of having filters + configuration setup
    - Nice switch also from reporting in Excel to reporting in JSON
    - COUNTER reports clustered in 4 different types
    - IIIF could be a type of database report (in COUNTER terms) or a Platform; maybe a title or an item report (like a multimedia item served to the user)
    - Assessing price of downloads, price of journal, etc. still available in COUNTER5
    - Decided to go with multimedia item reports in COUNTER for IIIF
    - Also with COUNTER 5, some of the peculiarities of the reports (like special fields in book reports) have been cleaned up + straightened out
- What are you counting?
    - Big topic in COUNTER 5
    - Data model you can work
    - In typical websites or other interfaces, there is more information served with content
    - Are users looking at abstract, content, details in metadata, other? This becomes the detailed investigation
    - Have a specific idea also of downloading a text
- Mapping out the usage
    - Starting mapping out from the Access logs

        - Mapped out investigations (just getting thumbnail or similar), items requests, no license (things that can't be shown on web)
- Instead of using an Item Master report, could have gone with platform report, title report, dataset report
    - But only have data from image server access logs
    - If you really map it out and cluster it together, you need data not in the logs to make the request report make sense
    - So went with item report
- Choose measuring IIIF image API
    - Could have used IIIF manifest or library record as the focus instead
    - Sticking with basic images, but open to suggestions for folks with good use cases to try something else
- "Real" usage
    - Filtering out…
        - the log, have to handle things like redirects, unsuccessful requests
        - Only using image API jpeg requests
        - Filtered out bots usage
    - Also, COUNTER expect concept of a session
        - Generating session IDs (ip address + user agent + transaction date + hour of day)
        - Just part of the standard
    - COUNTER also measuring double clicks
        - But if you think of an application that has zoom, you're doing double clicks there
        - Had to then filter out zoom events, which makes sense since all is 1 event, but means cutting out deep zoom from usage metrics
- TDM Usage (text + data mining)
    - How much is this data being used
    - Because COUNTER is picky about bots, this is separated out
    - In item report, mark bots as TDM, which maybe is good view on what it is doing
- Using their own IIIF service a lot
    - E.g. library catalog thumbnails are coming from IIIF server
    - OpenURL resolver that captures all requests
    - Their own viewer app uses this API as well
        - Viewer app doesn't show any changes for summer holidays
        - However, library website shows dips for summer holidays, have clean dip
    - Working with academics to make them aware of possibilities of IIIF, how they can integrate it in their environments in next year; nice to have usage measures in place during that
- Viewers

    - Item investigations are registered for looking at quite a few item thumbnails
    - Investigation also registered for zooms, with clicks rolled into same investigation
- Struggles
    - How to define + differentiate between full view, thumbnails, tiles for deep zoom
    - When you're zooming, machine is requesting tiles + partial images
    - Idea of investigation + requests + how to differentiate usage of content becomes quite difficult to reconstruct from access logs then
    - Tried to map out this usage
        - Thumbnails vs full view separation worked out
        - See people trying to get access to something no longer accessible, usually thumbnails stored in cache
        - Trying to understand how to map out seeing full images vs tile
- Open Questions
    - Should accessing JSON-LD access be counted as 'investigation' in own right?
    - Recognizing image tiles from thumbnails relies on heuristics (when is it a thumbnail? When is it a tile?)
    - This is a work in progress + they're looking for feedback from ELAG + IIIF community; see if data model makes sense + people would model this in the same way
- Questions
    - How do you explain to the chief of the library what the numbers mean? Digital world is much more fragmented; harder to see what usage means
        - Always have cartoon in mind where people complain about too many standards so get together + make a new one; interesting part of him is as the standard [COUNTER] is getting updated + somewhat more adapted to reality of internet, becomes opportunity to have this type of reporting realistically
        - How do people actually interpret numbers - it's why we're looking at COUNTER standard, so some agreement on definitions and data models to know why we're counting at all. Sticking to same kind of convention is helpful when trying to orient
    - Are you only counting item level requests, or collection level too?
        - Yes, only images.
    - Why only count images?
        - Already have book numbers, or how popular a book is on a website - those numbers already gotten from catalog website data.
    - If you look at image of book online, it is also a hit on the book itself. You're not adding to that?

        - Agrees. If you look at numbers, small number of image traffic (missed it). IIIF isn't just about images in image viewers, but also about shipping images. If you start to mix data, then you start to eclipse the machine access data information

GOKb - Cooperative Management of e-Collections for Automated Services / Moritz Horn, Daniel A. Rupp

- Seen over last few days, transformation of content to e-resources comes with new issues
- One issue is identifying e-resources in some consistent way; so want to tell you today about GOKB while looks to address this issue
- Introduction to knowledge bases
    - For things to function properly, need to be consistent in how you talk about resources in your collection. best way of doing that is having fairly rich central knowledge base in middle of that acting as an identity broker.
    - Knowledge bases are identity brokers for electronic resources
    - Goal is to identify resources, collection additional data about them; and make that data available in a structured way
    - Use cases for this include link resolvers, ERMs, discovery services, etc. Can use data in knowledge bases for fetching of bib records or stats or OA projects relying on GOKB data
- There are lot of KBs; what is special about GOKb
    - Freely accessible, reliable metadata
    - Management of packages + collections, including content + scope; titles (identifiers, history, etc. + data from bib records); and providers (hosts of URLs, hostnames, identifiers, etc.)
    - Open community; every library can participate
        - Have open APIs, CC0 license on metadata
- GOKb Organization
    - Project has been around a while, but in 2017 project joined consortium of German Institutions including ZDB; Steering Committee has many organizations including NCSU, CalTech
    - GOKb is focused on cooperative metadata on these resources
    - There are a number of curatory groups with different responsibilities
    - Cooperative management
        - If there's an error, a review request is issued
        - The groups are assigned to review
- Main components of GOKb
    - TIPP (title instance package platforms)
        - Title in package-specific context
        - Allows them to have information about a URL, Coverage, Access restrictions, etc.
        - Where to also record changes in packages

        - Import via Integration APIs
            - Mainly using JSON APIs
            - Separated endpoints for different components
            - JSON structure is very close to database model so basically can import all components
            - Important for titles especially is matching + merging functions, because want to allow updates of components with new information and also be careful not to merge something that is probably different resources
            - Via APIs, also generates review requests automatically that are then added allocated to user group for manual review
- Cross-referencing details
    - Can import very detailed information about titles; but because GOKb data is used for automated exchange, they normalize + validate quite a bit.
    - Using API also for manual imports
        - Importing KBART; KBART is tab-separated file representing package, with each line being a title or access information for title within package
        - Within KBART, there's package + title specific information
        - GOKB Title data is referenced against ZDB catalog via SRU, to get better data + more detailed information than is just provided by KBART
- Elasticsearch APIs
    - 2 main endpoints
        - Targeted component search (lots of filtering options, retrieval by UUID)
        - Type-ahead, for searching for names of components + get suggestions (uses n-gram)
    - Offers simplified JSON serializations of components
- Status of the work
    - Redeploying database in progress
    - Active pilot phase for package import
    - Coordination of some processes needing to be done around package import,
        - Have pilot institutions for this from LAS:eR and FOLIO
        - Working on this now for import coordination via JIRA
    - Development right now though focused on new user interface
        - Done at end of next year
        - In UI, want to integrate some functionality for data enrichment
    - Also would like to extend API functions (lots of things that can be made better there)
- Resources
    - https://github.com/openlibraryenvironment/gokb/wiki

        - https://openlibraryenvironment.atlassian.net/wiki/spaces/GOKB/overview
        - https://gokb.org
        - https://niso.org/publications/niso-rp-9-2014-kbart (talks about experiences with KBART files and limited functionality due to quality of data from them)
- Questions
    - Nice to have global open knowledge base, but it is a lot of work to keep it up to date; do you have any way to make sure users can expect quality from this?
        - Done cooperatively by the libraries; want to oversee this by a separate institution; hoping they'll anticipate usage of data; part of FOLIO project, open systems project, other projects, so the use will lead to the quality in the database.
    - How much data is currently in the database?
        - Not an easy question to answer because they have a kind of legacy database created from earlier work from NCSU colleagues; those are around 500 packages, and include around 30-40 thousand journal titles he thinks; they have another dataset they'll import as well from LAS:er project, with about 500 packages and similar amount of journals. Currently working on processing on demand, based on packages requested from users

Designing a new identity management system for the National Library of Greece / Nikos Voutsinas, Michalis Gerolimos

Not present; so breaking for an early lunch.

Microservices based on the Kafka Event Hub / Sebastian Schüpbach, Jonas Waeber (5 minute warning at 1:50)

- Swissbib
    - platform for data + search services
    - Aggregates repositories, networks, national licenses
    - Picks up data from more than 30 different sources on daily basis
    - Provides several interfaces from humans + machines to access data
    - Used by project as a data or service provider
- Problems with current solution of data management
    - Lots of different pipelines, which makes it harder to see what's going on
    - Pipelines are difficult to scale
    - They consist of strongly coupled components (depend on each other inside pipeline; parts of pipeline responsible for different concerns)
    - Misses central monitoring solution to see what's going on in pipeline
- Their new solution: central event hub
    - Came up with last autumn
    - Using Apache Kafka

- About Kafka
    - Platform for building data pipelines + stream based applications
    - Normally put in middle of connecting services
    - Data pushed + pulled from kafka cluster
    - Data in kafka represented as non-bounded streams of events (like a bibliographic record from one of sources)
    - Services then can react to data
    - Fault tolerant + resilient with high throughput, very horizontally scalable, lots of other aspects from big data that we love
    - Good integration with different kind of DBs + Big data frameworks
    - Apache license so OSS
    - Kafka Main APIs
        - Producer sends data to Kafka
        - Consumer API: pulling data from kafka
        - Streams API: transforming or aggregating data inside kafka
    - Typical Kafka setup
        - Partitioning of data
        - In middle is kafka cluster itself with 3 servers (called brokers in Kafka)
        - Brokers contain 2 topics, each topic is partitioned so data is more or less evenly distributed among topics
        - Left hand side is producer responsible for pushing data to kafka (in this example, pushed to topic 1)
        - At the same time, if the producer itself doesn't define a key, kafka cluster itself defines the key so every event has a key
        - Streams application within kafka cluster makes it possible to transform and work with the data in parallel, then push the data back to one of partitions
        - On right hand side, have consumer with 2 consumer instances, takes data from particular brokers and particular topics
    - Transactional Log
        - Immutable records (can't be changed after they're written)
        - Partition is transactional log saved on disk
        - Message order guaranteed within partition
        - Data temporarily kept in this log, or it is possible to have compacted log where the latest value for the key is always kept (like a log as table in database)
    - Components of stream application (aggregate + transforms data)
        - Application reads from one topic and writes to another topic
        - Tool operators responsible for steps are source processor on one hand, and sync processor on another
        - Then there are processor nodes for transformation or aggregation

        - Stick process nodes together, then you get a topology (processor chain)
        - Kafka streams APIs in 2 ways. One normally used is kafka streams DSL, build on top of processor node API. Streams DSL is good compromise between correctness + preciseness. Offers functions like map or merge, well known from functional programming
        - Kafka streams API offers 2 notions of stream of data;
            - one is a kstream (abstraction of records stream, stateless, where record is self-contained datum in dataset). Content of record is completely forgotten when it moves on
            - KTable: abstraction of the changelog stream, which is stateful, and similar to compacted log discussed before.
- Data transformation use cases
    - Started to implement 2 use cases
        - Collection MARC data from all ilses connected to them, transform this data, put into (?) to create master records; then indexed to Solr using a Flink - Kafka process
        - 30 million MARC records transformed into LOD to better facilitate how we use it. Have created a workflow that takes MARC records and builds about 100 millions bib resources, docs, items, person, orgs, works, etc. Each has a defined ontology (bibframe / frbr)
- Example of use case 2
    - Thomas Mann
    - Pulling in data from dbpedia
    - Used to do this work with scripts, a lot of them. Got to go back and fix bash scripts
    - Reworked this with Kafka event hub
- Changing to Kafka
    - Takes time to understand kafka, understand how it works, etc.
    - First part (loaders):
        - Loads data from sources via file read, goes 1 line per data, one resource per message
        - 3 kafka streams applications; transforms n-triples to json-ld, and filters out unused predicates to lessen burden
        - Put into elasticsearch vis the elasticsearch consumer
    - Second part:
        - Clustering sameAs relations
        - Read out all sameAs relationships + put together into 1 elasticsearch document
        - Allows easier processing of swissbib data by matching GNDs, also to enrich with dbpedia

        - Again, producer, consumer (consumer code is same as above; reused), and one streams application that does heavy lifting
    - Central idea of microservices here is reusing components, and see that here; refactored their consumer to do this, the elasticsearch consumer is stable
    - Looking at buildStream code, where Topology is defined
    - Everyone in development team has same language when developing things; and it is easier to reuse the code
- Advantages
    - Breaking up old workflow
    - Reusing old workflow components in new workflow
    - Reusable components (some)
    - Can use different languages (cotland (sp), scala, java, python)
    - Runs everything in parallel
- Disadvantages
    - Lots of different parts that you build, organize, etc. for things like deployment
    - Have to think about organization, otherwise you have same problem as before
    - Kafka stream is stream based; you can do batch processing with it, but not optimal way to do it, because you can often run into race conditions; so you need to think in streams and make sure use cases fits this, which isn't always the case
    - Running things in parallel can still cause problems, especially when things start to depend on each other
- Roadmap
    - Improve stability, finish implementations, testing, benchmarking
    - Logging + monitoring
    - Sometimes kafka cluster dies bc runs out of memory of disk space bc of amount of data
    - Need someone who knows how to set up these things + configure them
    - Supporting future use cases; a big one is an authority + research metadata hub where they plan to do heavy lifting of transformations with kafka + have them regularly available
    - Swissbib.org
    - gitlab.com/swissbib
    - swissbib.gitlab.io/presentataions/microservices-and-kafka
- Questions
    - Missed bc notetaker is facilitating
    - Something about what is doing transform (metafacture); comparison with nifi; and how are you coordinating devops work (check back in a year)

Building Library Information Systems in Times of Vanishing Developer Resources / David Zellhöfer, Oliver Schöner, Gerrit Gragert (5 minute warning at 2:20)

- Slides: doi.org/10.5281/zenodo.2682645
- Graphic novel artists picture hanging on his wall in his office
    - Comes from time when there was a lot of work in the department with many fewer people
    - Shows man towered over by books
    - He felt a bit like the guy in that picture
    - A bit depressing
- Have a huge amount of digital resources they are forced to deal with
    - Have to organize, save them, make them useable by patrons
    - Only have 1 human resource honestly to work on this; but they had a little luck because they started a bit late
    - When they started, there was an open world game, looking around to find existing digital resource management solutions
    - Looked at fedora, zenodo, dspace, archivematica (system more for long term preservation), opus, mycore (more german), apache jackrabbit
- Metrics they evaluated possible solutions against:
    - Needed universal solution, can take multiple forms + forms they don't know about yet
    - Metadata in all its formats
    - Most of the metadata they get they do not produce by themselves, but get from outside; there are resources they buy, license, and they get that metadata from the publishers; can get quite bad metadata at times from publishers
    - Need a system where it can integrate into other existing systems; so cannot be a monolith
    - Need authorization given some resources are behind license, so not all are world visible
    - Scalability
    - So after looking at all these solutions against these metrics, they decided to go with fedora
- Fedora
    - Decision mainly made for name; fedora is acronym that includes flexible
    - Fortunately starting with Version 4; there was major shift from fedora 3 to fedora 4; people who worked earlier with fedora 3 are forced to migrate data; but they could start with fedora 4
    - Has a simple REST API for basic CRUD; can concentrate on basic functions
    - Can talk to Fedora with programming language due to APIs

        - Fedora integrates well with Solr index; you need a Solr index bc data is not searchable within Fedora, so you need a search engine, etc.
        - All the data stored in fedora is in RDF; fedora stores triples but it is not a triplestore
        - If you need a triplestore, you can set up one and add an SPARQL endpoint; at moment, they do not have a use case to provide triples to outside, but if someone came to them and asked, they have a path to offer that
- Fedora has 2 major problems
    - Many members problem: imagine you have think books (here electronic resources with up to 40000 pages); modeling this in Fedora is a mess, because it is structured as a tree; book is a node, all pages are children of node; when someone from Fedora community is here, they'd like to address it again; no easy solution; they're working with buckets to there are fewer subcontainers with multiple pages inside it
    - Fedora is good for developers, but bad for other users because it has bad UI; when you talk with people, say fedora is great, then they ask if you can show them something and they say not really
- Fedora fits well in microservices paradigm
    - See fedora as small piece of software that can easily communicate to other pieces of software
    - Has simple REST API + fits quite well
    - Any piece of software that first with Fedora should be reusable
        - Can talk to other data providers like solr for example
        - So can easily replace components when needed
    - Access management service is important point; dealing often with licensed material, so not everyone is allowed to see the data; software build on top of fedora must be able to talk to management service; they're doing integration work with shibboleth to transform this information to JWTs that are then used by other applications
- Architecture
    - Bottom is storage (databases, Postgres);
    - because they have so many collections, they don't rely on single fedora instance, have 10 or so
    - For each instance, there is a solr pool
    - Then web front ends
- Have mainly 3 applications online
    - First is RGZ / RGSt with 30k objects; had to do quickly due to contracts; not a typical fedora use case; use Samvera stack with little code for data modeling
    - Stabirep: 1k objects (growing); used for self-deposit internally by librarians. The public doesn't see this interface, but can download files by link

        - ITR for Crossasia: ~100 million objects; users search for certain concepts and get the data back
- Remarks from management at the end
    - Problem with front end development with fedora, saw some solutions running Islandora or Samvera Hyrax solutions; problem at the time was to go with Hyrax (a ruby on rail tech stack), however rails isn't skill in house;
    - Other problem: not an option to start a new front-end for fedora
    - So had to made a hard choice; went with Hyrax as out of the box solution where just need a front end for self-deposit
    - Used tools from Samvera stack to ingest + modify data
    - Cannot give something back from their devs at the moment however
    - At least now a member of Duraspace to give something back
    - Have 3rd party funding / temporary contracts for some of these projects; have to cooperate with developers in house
    - Future is linked; quite ready for linked data applications, they are on their roadmap
    - Don't want to build *the* repository for their institutions, but instead build multiple data silos for collections + connect them with linked data
    - Repository itself is not a catalog, but a storage for digital objects; not where you search, but where you store
- Question
    - Missed bc facilitator

Closing Ceremony

- PVB
    - Great few days!
    - Everyone is leaving and sad!
    - Keeping people here to very last moment is not easy task so thanks all for staying!
    - If you have suggestions to keep more people the whole time, tell PVB!
    - Thanks to the photographer; photos are on the website (see 'impressions' https://www.elag2019.de/impressions.html)
    - If you want to join us in ELAG, we always want to have more participation; you can contact PVB or any other member of ELAG community
    - If you're capable of organizing conference in the future, also can contact PVB (peter van boheemen [sp])
    - Thanks to local committee in Berlin
- Berlin local committee
    - Looking back to good conference
    - Productive coffee breaks
    - Couldn't have happened with presenters, keynoters, + bootcamp + workshop organizers
    - (lots of thank yous to people who made it happen)