The Current State of Research Networking

- We are well into the first generation of tools, but we are still conceptualizing what successive generations might look like.
  - As support for this, consider that the majority of participating institutions are still in the process of fully deploying their first research networking system.
VIVO

- As meme
- As project
- As platform
- As ontology

How do you tell whether you’re ‘compatible’?

- Option 1 – run VIVO as a platform
- Option 2 – field a SPARQL endpoint serving RDF data aligned with the VIVO ontology
Loki’s VIVO Stack

• D2RQ – SPARQL to JDBC translator
• Teiid – data federation layer (VIVO-like virtual database)
• PostgreSQL – Loki’s storage layer (Loki native data model)

But do you then look like VIVO?

– Validate with a bunch of SPARQL queries
– Iterate through the ontology using Jena

– For each class in the ontology
  • SELECT count(?inst) WHERE { ?inst <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <aClass> }

– For each class for each property
  • SELECT count(?prop) WHERE { ?inst <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <aClass>. ?inst <aProp> ?prop}
The Google Imperative

- The broad population of users are addicted to “just-relevant-enough” search – fast
- The problem is that distributed semantic search (i.e., SPARQL driven by an ontology) is precise, and slow

Meta-search or Federated Search?

- The two iterations of direct2experts are good examples of meta-search
  - All data stored at the edges
  - Search cost (computation) is distributed to the edges
  - Aggregation occurs at query time, with only partial information as to why a hit ranks the way it does
Meta-search or Federated Search?

- Federated search can blend multiple information sources in a unified manner, but at a cost
  - Source data must be harvested
  - Aggregated data must be stored at the federation site
  - Search cost occurs at the federation site

Initial CTSAsearch Architectural Choices

- Use SPARQL to harvest just the data to be used in indexing
- Cache data locally to support experimentation / value-addition
- Generate index from cached data, with hits pointing at person URIs
Initially harvested:

- Persons and demographic properties
  - URI
  - First and last name
  - Title (optional)
  - Email address (optional)
  - Phone number (optional)
  - Research overview and areas (optional)
  - Keywords (optional)

Initially harvested:

- Academic articles and properties
  - URI
  - Label (optional)
  - PMID (optional)
  - DOI (optional)
  - PMCID (optional)
Initially harvested:

• Authorship and properties
  – Person URI
  – Article URI
  – rank (optional)
  – isCorrespondingAuthor (optional)

Enrichment

• Academic articles (with inferrable PMID)
  – Abstract
  – Keywords
  – MeSH terms
  – Chemicals
  – Genes
Concept-driven search

- Named entity recognition pass against all suitable fields
  - Recognized UMLS CUIs added to index record for person
- Search text processed similarly
  - All UMLS CUIs and subtype CUIs added to user query

Inter-institutional Coauthorship

- Materialize
  - Article URI
  - DOI
  - PMID
- Write big query joining author-authorship-view-authorship-author chain
- Result materialization
  - Author, coauthor, count
Lessons Learned from Initial Configuration

• Based on ontology coverage analysis, many (most!) sites get very sparse outside of persons and publications
• Cross-site identification of collaboration is quite feasible once publications appear
Lessons Learned from Initial Configuration

- Many sites find SPARQL endpoint deployment “challenging”
- It’s remarkably easy to crash a SPARQL endpoint…
  - Some queries return millions of triples

Architectural Alternatives

- Have sites submit/serve dump files
  - Current approach for UCSF, OHSU
  - Added manually to CTSAsearch database
Architectural Alternatives

• Crawl public VIVO interface
  – Crawler implemented, working on data integration
  – Successfully crawled
    • ColPos (Mexico)
    • Eindoven (Netherlands)
    • Melbourne (Australia) (in progress, 9.5m triples and counting)
    • APA, Colorado, Idaho, Nebraska, New Mexico

Lessons Learned from Current Configuration

• Many sites find VIVO deployment “challenging”
• It’s remarkably easy to crash a VIVO…
  – Rate limiting the crawler seems to have helped
• Eindoven crawl:
  • 56 hours
  • 4.3 million triples
Next Steps

- Field specialized crawlers for
  - Harvard Profiles
  - SciVal Experts
  - Digital CV platforms
  - CV PDFs

Next steps

- CiteSeer-like Web crawler is running
  - Capturing DOIs, PMIDs and links to PDFs
    - Current database is 45m URLs
      - 3m DOIs
      - 800k PMIDs
      - 12m PDFs
- Further profile enrichment
  - New domains (e.g., hard sciences)
  - New sources (e.g., GenBank, EMR, …)
Questions?

• Email: david-eichmann@uiowa.edu
• Thanks to the Research Networking Collaborative