

Relation Extraction for Ontology Construction

Kate Byrne

Institute for Communicating and Collaborative Systems

19 January 2006



Outline

Motivation

Overview

The source data

Currently available systems

Some specific goals

Method

Basic ontology design

Automatic ontology construction

Relation extraction

Querying graph data

The ontology schema

Management

The project plan



Outline

Motivation

- Overview

- The source data

- Currently available systems

- Some specific goals

Method

- Basic ontology design

- Automatic ontology construction

- Relation extraction

- Querying graph data

- The ontology schema

Management

- The project plan

Outline

Motivation

Overview

The source data

Currently available systems

Some specific goals

Method

Basic ontology design

Automatic ontology construction

Relation extraction

Querying graph data

The ontology schema

Management

The project plan



Overview: the proposal in a nutshell

- Cultural heritage data — fixed fields, free text, thesauri
 - Query problems:
 - Querying data with fixed fields is easy
 - Querying data with free text is hard
 - Limited access to free text
 - Proposal: query application that will guide non-expert users
 - Two key elements of proposed solution:
 - User interface to guide user through application
 - Knowledge engineering to handle all data in the database



Overview: the proposal in a nutshell

- Cultural heritage data — fixed fields, free text, thesauri
- Query problems:
 - complex data structures, specific to each collection
 - specialist domain terminology
 - limited access to free text
- Proposal: query application that will guide non-expert users
- Two key elements of proposed solution:



Overview: the proposal in a nutshell

- Cultural heritage data — fixed fields, free text, thesauri
- Query problems:
 - complex data structures, specific to each collection
 - specialist domain terminology
 - limited access to free text
- Proposal: query application that will guide non-expert users
- Two key elements of proposed solution:

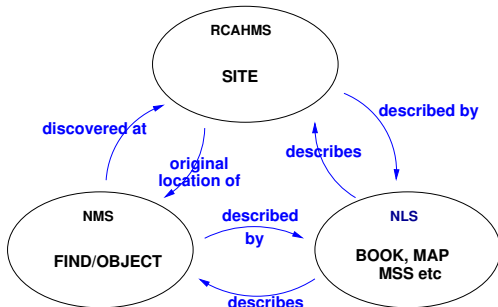


Overview: the proposal in a nutshell

- Cultural heritage data — fixed fields, free text, thesauri
- Query problems:
 - complex data structures, specific to each collection
 - specialist domain terminology
 - limited access to free text
- Proposal: query application that will guide non-expert users
- Two key elements of proposed solution:
 - *relation extraction*: two-place predicates from free text
 - *ontology construction*: combine all data in one simple format

Source Data

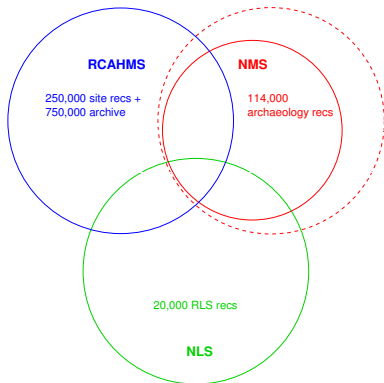
- 3 datasets: RCAHMS, NLS, NMS
- Similar topics and vocabulary: archaeology, Scottish history
- Unexploited relationships





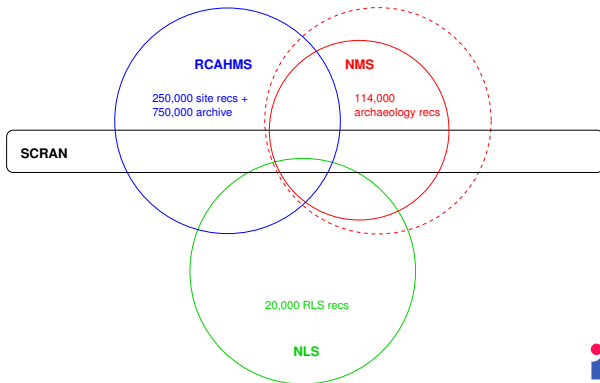
Source Data — and SCRAN, RLS

- SCRAN — founded by NMS, RCAHMS (and SMC)
- RLS — lead body NLS
- Also contributing: museums, galleries, local history societies...



Source Data — and SCRAN, RLS

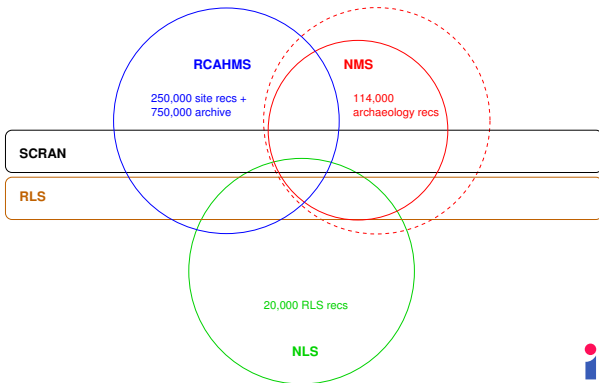
- SCRAN — founded by NMS, RCAHMS (and SMC)
- RLS — lead body NLS
- Also contributing: museums, galleries, local history societies...





Source Data — and SCRAN, RLS

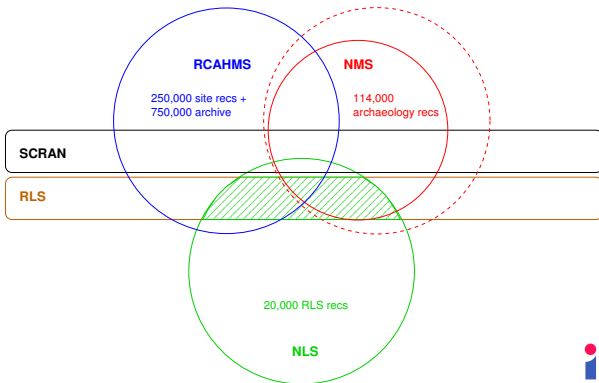
- SCRAN — founded by NMS, RCAHMS (and SMC)
- RLS — lead body NLS
- Also contributing: museums, galleries, local history societies...





Source Data — and SCRAN, RLS

- SCRAN — founded by NMS, RCAHMS (and SMC)
- RLS — lead body NLS
- Also contributing: museums, galleries, local history societies...





Existing Query Applications

- CANMORE and CANMAP — www.rcahms.gov.uk
- SCRAN/RLS — www.scran.ac.uk
- Several others...
- Problems:
 - terminology
 - data structure
 - free text searching
 - results presentation

Demo — SCRAN, CANMORE



Main Goals

- Permit hybrid queries
- Provide guided queries
- Results summarisation
- Allow cross-collection queries



Secondary goals

- Address “Hidden Web” problem
- Achieve schema flexibility
- Deal with updates and maintenance
- Provide Natural Language presentation
 - M-PIRO project (Isard et al, 2003)



Motivation

Overview

The source data

Currently available systems

Some specific goals

Method

Basic ontology design

Automatic ontology construction

Relation extraction

Querying graph data

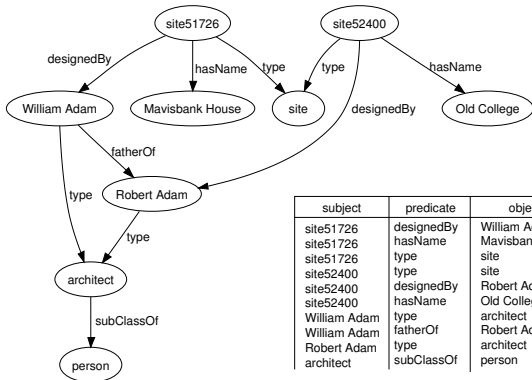
The ontology schema

Management

The project plan

What do I mean by “Ontology”?

- Graph of *Subject, Predicate, Object* triples
- Populated with instance relations
- Represented in RDF, in a triple store with database storage
- Using RDFS for *subClassOf, type, domain, range* etc





RDF Storage

- In memory or in persistent storage?
- Lots of interest at present
- Basically a 3-column table: *Subject, Predicate, Object*
- Traverse graph by self-joining this table
- ...but, each has potentially useful attributes:
 - actual literal or URI vs canonical form
 - level in hierarchy (Navigli and Velardi, 2003)
 - “parent” node (*site* for RCAHMS etc)
 - type (faster than finding RDFS *type* relation?)
- More efficient than just adding extra graph relations(?)
- In other words: indexing/denormalising for performance



Automatic Ontology Construction

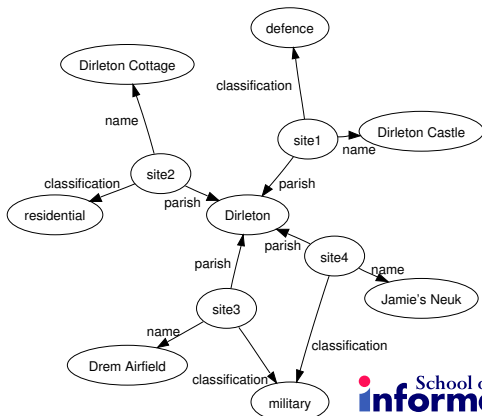
- Bottom up approach:
 - instance population first
 - then infer schema relations
- From database fields
- From thesauri
- From free text — relation extraction

Ontology From Database Fields

- Straightforward RDBMS to RDF conversion
- (Tested: around 40,000 records → 1.3 million triples)

SITE

siteNo	name	parish	classification
1	Dirleton Castle	Dirleton	defence
2	Dirleton Cottage	Dirleton	residential
3	Drem Airfield	Dirleton	military
4	Jamie's Neuk	Dirleton	military





Ontology From Thesauri

- Many available: TMT, FISH, SPECTRUM, AAT, LCSH...
- CIDOC Conceptual Reference Model
- W3C SKOS framework
- Automatic translation to RDF graph

Relation Extraction Methods

- Grammar based:
 - hand-written rules, machine learning, combination
 - problems: volume of data, lack of annotated data
 - experiments inconclusive so far
- Statistical associations:
 - frequently co-occurring terms
 - poor results in my experiments
- Planned approach:
 - pre-processing to find key **entities**
 - simple heuristics to combine them into relations
 - compare with machine learning



Relation Extraction Methods

- Grammar based:
 - hand-written rules, machine learning, combination
 - problems: volume of data, lack of annotated data
 - experiments inconclusive so far
- Statistical associations:
 - frequently co-occurring terms
 - poor results in my experiments
- Planned approach:
 - pre-processing to find key **entities**
 - simple heuristics to combine them into relations
 - compare with machine learning

Relation Extraction Methods

- Grammar based:
 - hand-written rules, machine learning, combination
 - problems: volume of data, lack of annotated data
 - experiments inconclusive so far
- Statistical associations:
 - frequently co-occurring terms
 - poor results in my experiments
- Planned approach:
 - pre-processing to find key **entities**
 - simple heuristics to combine them into relations
 - compare with machine learning

Step 1 — Entity References

1. Package free text fields as individual documents
2. Tokenise, POS tag, chunk
3. NER, using marked-up training data
 - (F-scores average 85%)
4. Co-reference resolution
5. Translate each entity to canonical form

Step 2 — Construct Relations

6. Identify candidate predicates
7. Clustering step
8. Build triples:

site101:

A chambered cairn as described and planned by Henshall.

NE, type "site_classification"



"describedBy"
predicate



surveyedBy
predicate



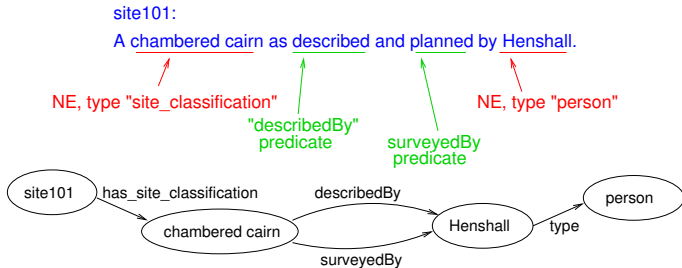
NE, type "person"



9. Merge with rest of ontology

Step 2 — Construct Relations

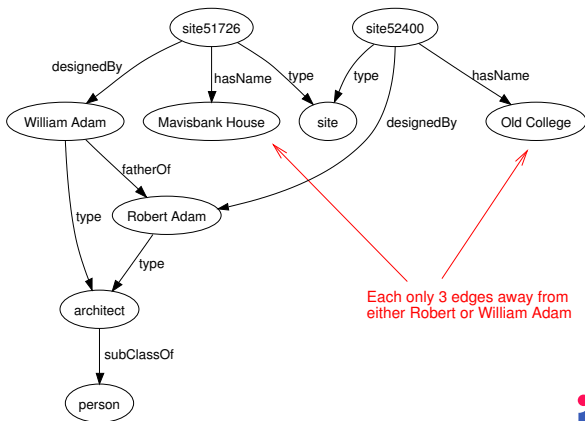
6. Identify candidate predicates
7. Clustering step
8. Build triples:



9. Merge with rest of ontology

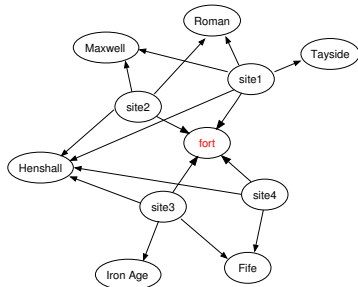
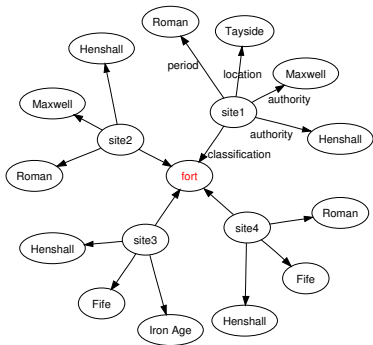
Graph vs RDBMS

- Possible to detect implicit relationships



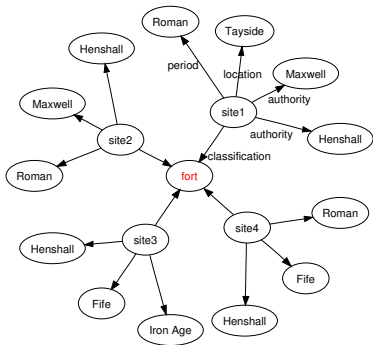
Graph vs RDBMS

- Easier to summarise under multiple headings



Graph vs RDBMS

- Easier to summarise under multiple headings

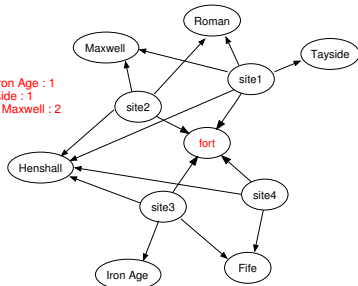


Summary for "fort":

Period – Roman : 3, Iron Age : 1

Location – Fife : 2, Tayside : 1

Authority – Henshall : 4, Maxwell : 2



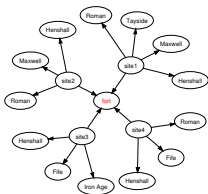


RDF Query Languages

- There are lots!
- SPARQL backed by W3C
- Limitations:
 - paths between pair of nodes
 - k -neighbourhood of node (need for “Adam” example)
 - degree of node (need for summaries)
 - diameter of subgraph
- SQL translation

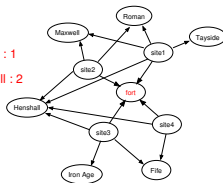
Revisiting the Ontology Schema

- Fully automatic would be nice, but...



Summary for "fort":

Period – Roman : 3, Iron Age : 1
Location – File : 2, Tayside : 1
Authority – Henshall : 4, Maxwell : 2



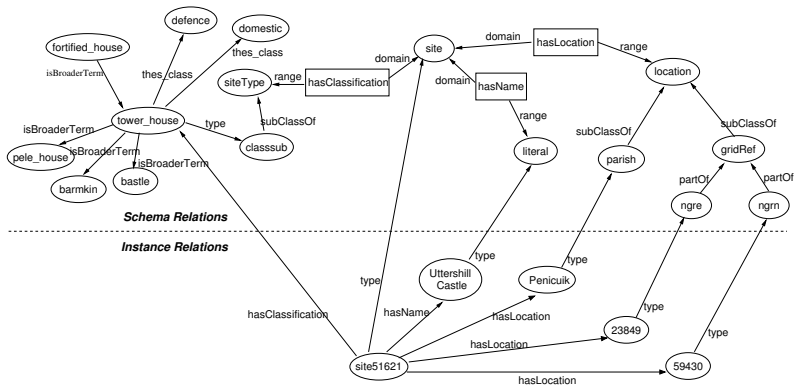
- Summarising requires:
 - finding sets of nodes by predicate label (eg all *period* ones)
 - clustering values within those sets (n Roman, m Iron Age)
- Therefore need to limit number of different predicates



Limiting Size of Predicate Set

- Can do automatically for text relations
 - predicates chosen by POS analysis, followed by clustering
- Manual task for database attributes
 - database column names are arbitrary
- Group similar attributes/predicates together
 - eg *location*: {parish, grid reference, street name,...}
- Push complexity into schema

Move Detailed Structure Into Schema





Motivation

Overview

The source data

Currently available systems

Some specific goals

Method

Basic ontology design

Automatic ontology construction

Relation extraction

Querying graph data

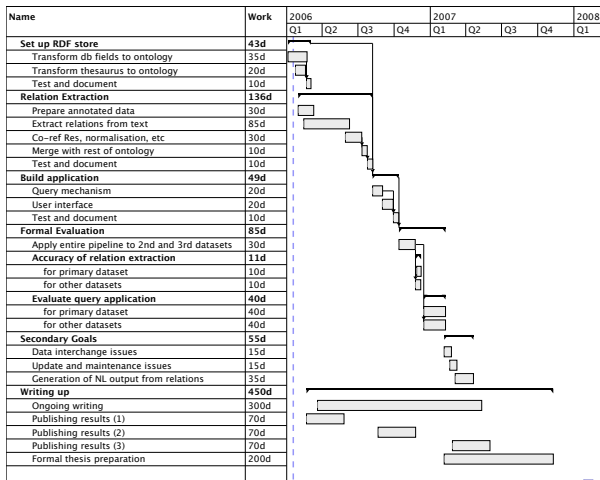
The ontology schema

Management

The project plan



Project Plan



Summary

- Relation extraction
- Automatic ontology building
- Query application:
 - search criteria tailored for each query
 - context of results provided
 - user not expected to specify criteria
- Simple format makes combining data easier