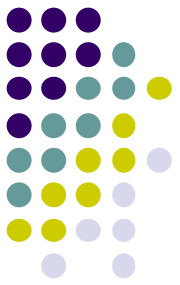


RDF123: from Spreadsheets to RDF

Lushan Han, Tim Finin, Cynthia Parr, Joel Sachs, and Anupam Joshi

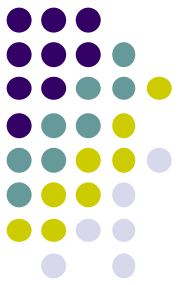
UMBC
AN HONORS
UNIVERSITY
IN MARYLAND





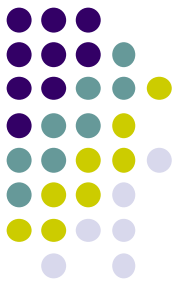
Road Map

- Motivation
- Related Work
- Translation Design
- Incorporating Metadata
- RDF123 Application
- RDF123 Web Service
- RDF123 Map Layer
- Problems and Future Work



Motivation

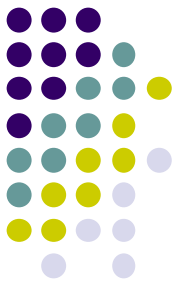
- One bottleneck of the Semantic Web is lack of data. We hope end users can participate in building the Semantic Web by contributing their own data.
- On the other side, a significant amount of the world's data is maintained in spreadsheets.
 - easy to understand and use
 - representational power adequate for many common purposes
 - online spreadsheets support collaboration.
- Thus, Spreadsheets provide a good media that can be directly maintained by end users and automatically translated into RDF



Related Work

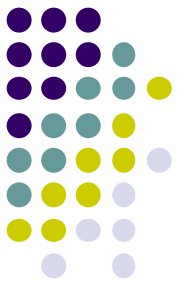
- Existing programs to convert spreadsheet to RDF, such as ConvertToRDF
 - map only to star-shaped RDF graphs, not flexible enough for general purpose spreadsheets
- GRDDL
 - Spreadsheet $\rightarrow$ XML $\rightarrow$ RDF :Involve an additional step to push the spreadsheet data to XML
 - XSLT transform, which GRDDL relies on, is hard to create for users who are not XSLT specialists.

Translation Design – overview



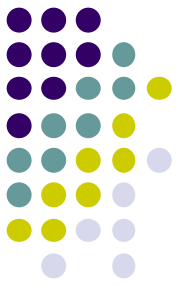
- RDF123's translation from a spreadsheet to an RDF graph is driven by a map which permits
 - a rich schema to apply to a row, rather than just creating a single instance of a RDF/OWL class.
 - allows different rows to use fairly different schemata

Translation Design – In detail



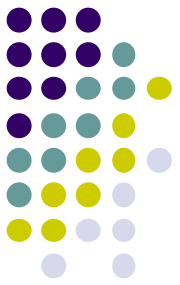
- Every row of a spreadsheet will generate a row graph.
- the RDF graph produced for the whole spreadsheet is the merge of all row graphs, eliminating duplicated resources and triples.
- If we overlap these row graphs by unifying vertices and edges, we end up with a graph that is a super graph of every row graph, with similar vertices/edges in different row graphs converging on a single vertex/edge.
- We name the super graph as the *map graph*.

Translation Design – In detail



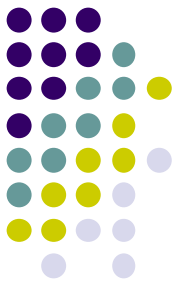
- When the *map graph* should produce different labels for a converged vertex or edge in different row graphs, an expression is used for the vertex or edge rather than a static label.
 - Expressions can use if-then-else sub-expressions and string manipulation operators to compute a label
- Since the *map graph* is a super graph of every row graph, for those vertices and edges which are in the *map graph* but absent from a row graph, the expressions will output empty strings, which signal that no vertex or edge should be created.

Translation Design – how to find the map graph

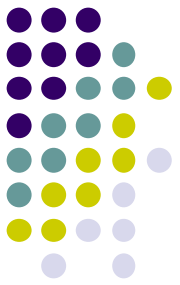


- Typically the map graph resembles a diagram of entities and their relationships that captures what users have interpreted from a spreadsheet.
- Spreadsheets provide a convenient way for users to capture the similarity of data, group and store similar data together in a succinct, informal but intuitive schema.
- RDF123 map graph can be a template that copies the intuitive schema of a spreadsheet and allows subtleties and dissimilarities within similarity to be expressed with RDF123 expressions.

Translation Design - Expression

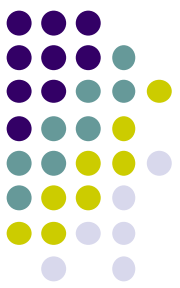


- The role of an RDF123 expression is to produce a final label for a converged vertex or edge.
- Has a context-free grammar and is able to do branch, arithmetic and string processing operations.
- While string concatenation and equality use an infix notation, other operations employ a functional notation. such as `@If(arg1; arg2; arg3)` and `@Add(arg1, arg2)`
- expressions can be recursively embedded in other expressions



Translation Design - Vertex Type

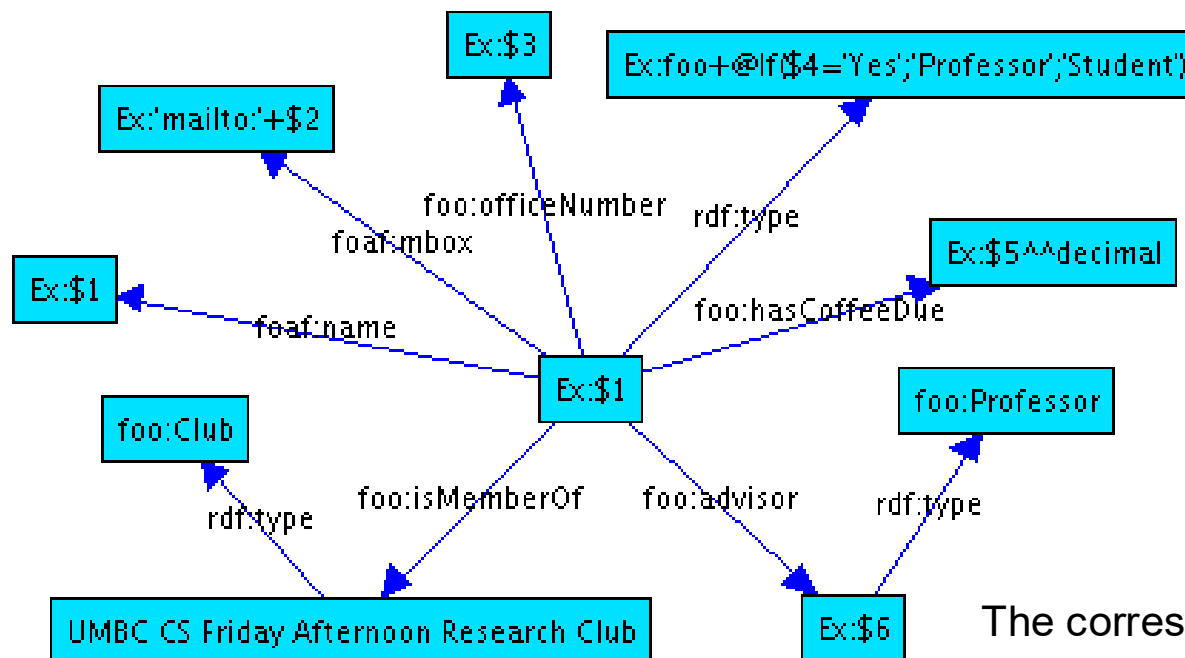
- We need know the RDF element type for the converged vertex before we can output the data as RDF.
- The potential type could be one of several data types (e.g., `rdf:Resource`, `rdf:Literal`, XML data types) or even composite data types like RDF container and collection
- We allow users to explicitly append a vertex type at the end of the static label or RDF123 expression. For example, `Ex:$1^^integer`.
- When lacking an explicit data type, we take the following heuristic: For those vertices which have outgoing edges, we make it `rdf:Resource`. For those leaf vertices, if the final label is a valid URI, we make it a `rdf:Resource` otherwise a `rdf:Literal`.



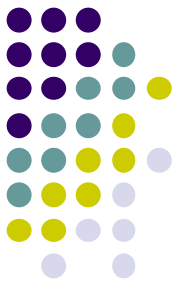
Translation Design – Example

- A simple spreadsheet for the members of a research club

Name	Email	Office	Faculty	Coffee Due	Advisor
Tim Finin	finin@umbc.edu	ITE329	Yes	\$10	
Lushan Han	lushan@umbc.edu	ITE377	No		Tim Finin
Wenjia Li	wenjia@umbc.edu	ITE377	No		Anupam Joshi



The corresponding map graph

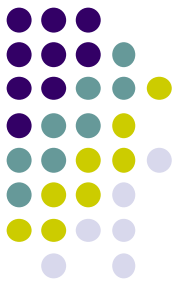


Translation Design – Example

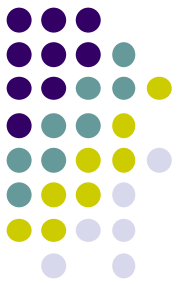
- This is the map graph serialized in RDF/XML syntax

```
<rdf:RDF
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:foo="http://www.foo.org/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <foo:Club rdf:about="#UMBC CS Friday Afternoon Research Club"/>
  <rdf:Description rdf:about="Ex:$1">
    <rdf:type rdf:resource="Ex:foo+@If($4='Yes';'Professor';'Student')"/>
    <foaf:mbox>Ex:'mailto:'+$2</foaf:mbox>
    <foaf:name>Ex:$1</foaf:name>
    <foo:officeNumber>Ex:$3</foo:officeNumber>
    <foo:hasCoffeeDue>Ex:$5^^decimal</foo:hasCoffeeDue>
    <foo:advisor>
      <foo:Professor rdf:about="Ex:$6"/>
    </foo:advisor>
    <foo:isMemberOf rdf:resource="#UMBC CS Friday Afternoon Research Club"/>
  </rdf:Description>
</rdf:RDF>
```

Translation Design - Summary

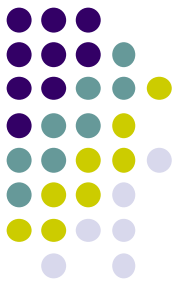


- high expressiveness since the map graph can be arbitrary graph.
- More intuitive than an XSLT transformation because it is expressed as a graph and can be visualized and authored with RDF123 graphical application.



Incorporating Metadata

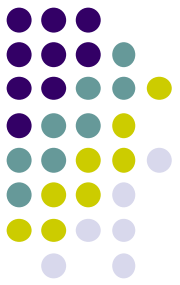
- RDF123 allows users to specify metadata both in map file and in spreadsheets.
- The metadata serves two functions.
 - One is to provide parameters to the translation procedure, such as the spreadsheet region containing the table to be translated, the map file's URI and etc.
 - The other is to add RDF descriptions to the produced RDF graph, such as title, author, and comment. Besides functioning as annotations, the descriptions also provide an identifier via a map file or spreadsheet template to facilitate search.



Metadata in a spreadsheet

- Spreadsheet metadata is embedded into a contiguous and isolated tabular area with two columns and a header 'rdf123:metadata'. This way of specifying metadata is preferred when you are the owner of the spreadsheet

File Edit Table				
	A	B	C	D
1		employee	email	phone
2		Mary Smith	msmith@foo.com	410-455-1000
3		Bob Jones	bjones@foo.com	410-455-1001
4		Bill Gates	dtrump@foo.com	410-455-1002
5				
6				
7		rdf123:metad...		
8		title	employee table	
9		creator	Lushan Han	
10		start row		1
11		end row		4
12		start column		2
13		row head	yes	
14		comment	this is a test example	
15		type	abc:EmployeeSheet	
16		abc	http://abc#	
17		map file	http://userpages.umbc.edu/~lushan1/employeeMap2.rdf	
18				
19				

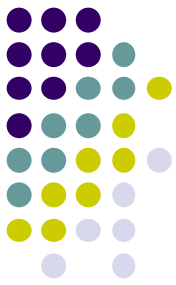


Metadata in the map graph

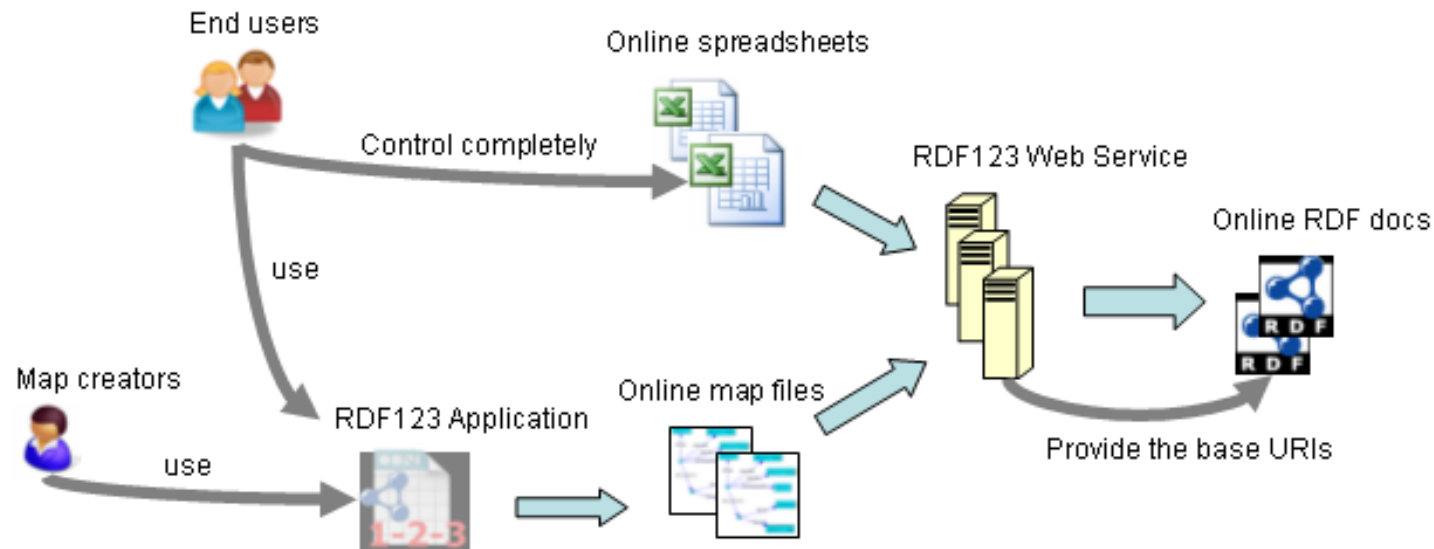
The RDF123 expression 'Ex:?' stands for the base URI of the online RDF document to be translated to. The properties 'rdf123:startRow' and 'rdf123:endRow' are used to specify the translation metadata. This way of specifying metadata is used when the map file is applied to other people's online spreadsheets

```
<rdf:RDF
  xmlns:emp="http://emp.example.org/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf123="http://rdf123.umbc.edu/ns/"
  <rdf:Description rdf:about="Ex:?">
    <rdf123:startCol>3</rdf123:startCol>
    <rdf123:startRow>6</rdf123:startRow>
    <rdf123:endRow>9</rdf123:endRow>
    <rdfs:comment>use metadata in a map file</rdfs:comment>
  </rdf:Description>
  <foaf:Person>
    <foaf:name>Ex:$1^^string</foaf:name>
    <emp:supervisor>
      <foaf:Person>
        <foaf:name>Ex:$4^^string</foaf:name>
      </foaf:Person>
    </emp:supervisor>
  </foaf:Person>
</rdf:RDF>
```

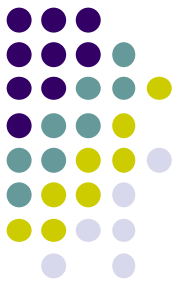
RDF123 Architecture



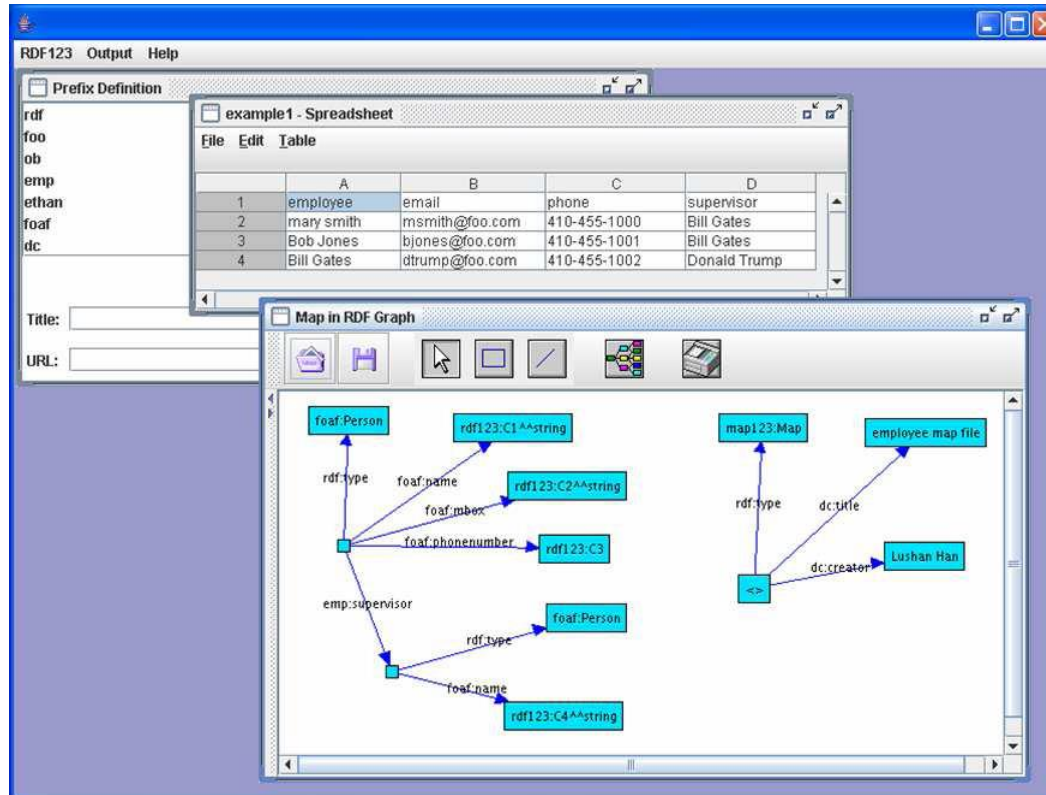
- RDF123 consists of two components, the RDF123 application and the RDF123 web service.
- The application provides a graphical interface for authoring RDF123 maps.
- The Web service is designed to automatically generate RDF documents from online spreadsheets either by specifying the location of RDF123 maps in the service or the spreadsheet itself.

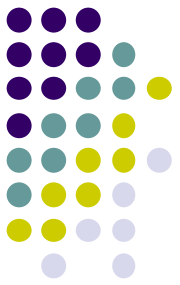


RDF123 application



- RDF123 application provides a graphical interface for creating, inspecting and editing RDF123 maps and using them to generate RDF documents from local spreadsheets





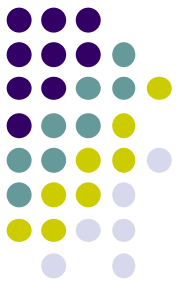
RDF123 Web Service

- RDF123 web service has a simple syntax.
- The service URL is <http://rdf123.umbc.edu/server/> and it takes three basic parameters: 'src', 'map' and 'out'.
- If a spreadsheet has an embedded link to its online map file, we just need to specify the URL of the spreadsheet with the 'src' parameter.
- The parameter 'out' is used to specify the output syntax. Default one is rdf/xml.
- Currently support two spreadsheet format: CSV and Google Spreadsheet

Example:

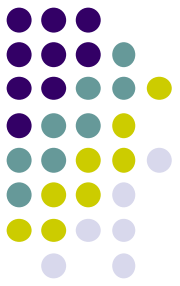
<http://rdf123.umbc.edu/server/?src=http://rdf123.umbc.edu/csv/office4.csv>

RDF123 Map Layer

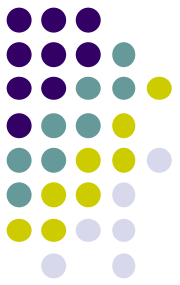


- Adding a map layer between the original data in spreadsheets and converted data in RDF can smooth data reusability and maintenance.
- By using different RDF123 maps, the same data can be available in different domains just by associating it with different map files.
- Data maintenance is eased, since data is directly maintained by spreadsheet owners and the RDF data is always rendered current.

A easy way to publish and Harvest RDF Data from Spreadsheets



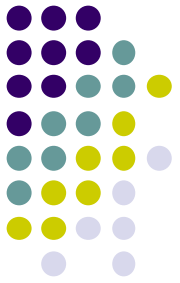
- Many RDF123 spreadsheet templates about different subjects can be distributed among end users.
- End users can fill in their own data and publish the instantiated spreadsheet.
- Query Google for spreadsheet files using keywords that are particular to RDF123 metadata like 'rdf123:metadata' and the identifiers in templates
- Convert them to RDF through RDF123 Web service



Problems and Future Work

- Although drawing a map graph in the RDF123 application is not hard, choosing proper Semantic Web terms and dealing with URI would be very hard for end users.
- Different people, without communication between them, may use different set of terms in authoring a map graph although the concepts in their spreadsheets may be the same. This makes data integrating very hard.
- Future work: We are developing a system allowing users to simply use English names in authoring the map graph and the system can map the set of English names to the set of the most standard and consistent Semantic Web terms in spite of slightly different ways people may give names to their concepts.

End



- Thank you!!!
- Questions?