Integrating Nested Data into Knowledge Graphs with RML Fields

Thomas Delva, Dylan Van Assche, Pieter Heyvaert, Ben De Meester and Anastasia Dimou

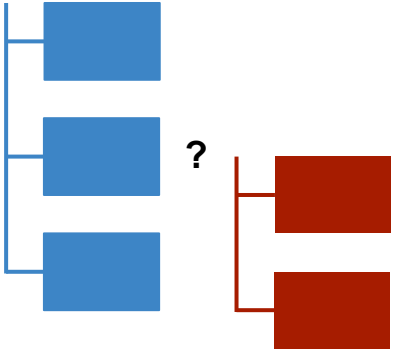
IDLab, Ghent University - imec

Thomas.Delva@ugent.be
@ThomasDelva

Current declarative mapping languages cannot always integrate nested data

Languages offer partial solutions

Fail mixing data from ≠ levels



Nested data is data with information on multiple levels

Very common

XML, file directories, RDF-star,...

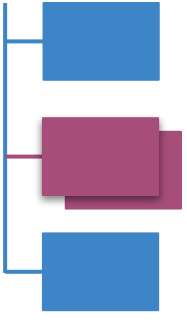
Large variety

Nested data is data with information on multiple levels

Very common

XML, file directories, RDF-star,...

Large variety (1/3)



Nested data is data with information on multiple levels

Very common

XML, file directories, RDF-star,...

Large variety (1/3)

```
nickname;items
Ash;     sword,gloves
Misty;   gloves,mittens,hat
```

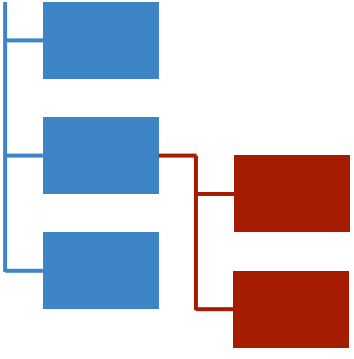
IDLab
INTERNET & DATA LAB

Nested data is data with information on multiple levels

Very common

XML, file directories, RDF-star,...

Large variety (2/3)



Nested data is data with information on multiple levels

Very common

XML, file directories, RDF-star,...

Large variety (2/3)

```
<Characters>
  <Character nickname="Ash">
    <Items>
      <Item type="sword" cost="4400"/>
      <Item type="gloves" cost="340"/>
    </Items>
  </Character>
  <Character nickname="Misty">
    <Items>
      <Item type="gloves" cost="340"/>
      <Item type="mittens" cost="400"/>
      <Item type="hat" cost="800"/>
    </Items>
  </Character>
</Characters>
```

IDLab
INTERNET & DATA LAB

Nested data is data with information on multiple levels

Very common

XML, file directories, RDF-star,...

Large variety (2/3)

```
{“characters”:[
  { “nick”: “Ash”,
    “items”: [
      {“type”: “sword”, “cost”: 4400},
      {“type”: “gloves”, “cost”: 340}]},
  { “nick”: “Misty”,
    “items”: [
      {“type”: “gloves”, “cost”: 340},
      {“type”: “mittens”, “cost”: 400},
      {“type”: “hat”, “cost”: 800}]}]}
```

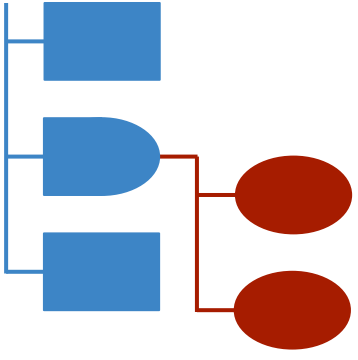
IDLab
INTERNET & DATA LAB

Nested data is data with information on multiple levels

Very common

XML, file directories, RDF-star,...

Large variety (3/3)



Nested data is data with information on multiple levels

Very common
XML, file directories, RDF-star,...

Large variety (3/3)

nick	items
Ash	[{“type”: “sword”, “cost”: 4400}, {“type”: “gloves”, “cost”: 340}]
Misty	[{“type”: “gloves”, “cost”: 340}, {“type”: “mittens”, “cost”: 400}, {“type”: “hat”, “cost”: 800}]

IDLab
INTERNET & DATA LAB

RML fields: a nested iteration model for RML

Inspired by xR2RML, ShExML
While avoiding their limitations

Gives names to values
Independent of data format

Solves related mapping challenges
Proposed by community group

RML Fields

Nested data

RML extension

Mapping challenges

RML Fields

Nested data

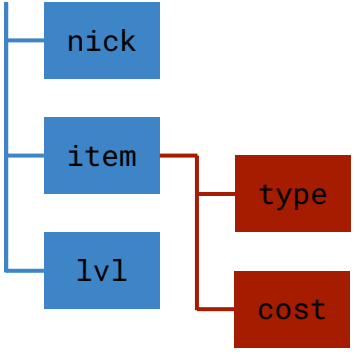
RML extension

- Extracting information
- Writing RDF
- Syntactic sugar

Mapping challenges

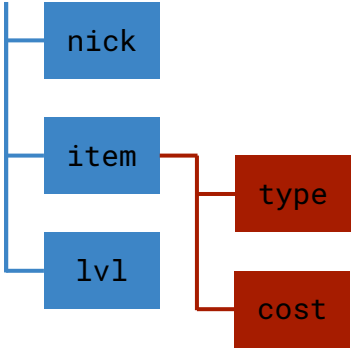
Fields give a name to values

```
{“characters”:[
  { “nick”: “Ash”,
    “items”: [
      {“type”: “sword”, “cost”: 4400},
      {“type”: “gloves”, “cost”: 340}]},
  { “nick”: “Misty”,
    “items”: [
      {“type”: “gloves”, “cost”: 340},
      {“type”: “mittens”, “cost”: 400},
      {“type”: “hat”, “cost”: 800}]}]}
```



Fields are declared with RML references

```
<LS> a rml:LogicalSource;
  rml:iterator "$.characters[*]";
  rml:field [
    rml:name "item";
    rml:reference "$.items[*]";
    rml:field [
      rml:name "type";
      rml:reference "$.type"]].
```

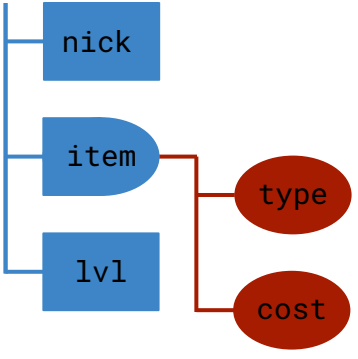


IDLab
INTERNET & DATA LAB

Fields can have different reference formulations

Inspired by xR2RML mixed-syntax paths

```
<LS> a rml:LogicalSource;
  rml:referenceFormulation rr:JSONPath;
  rml:field [
    rml:name “item”;
    rml:reference “$.items[*]”;
    rml:field [
      rml:referenceFormulation ql:CSV;
      rml:name “type”;
      rml:reference “0”]].
```



IDLab
INTERNET & DATA LAB

Fields can have different reference formulations

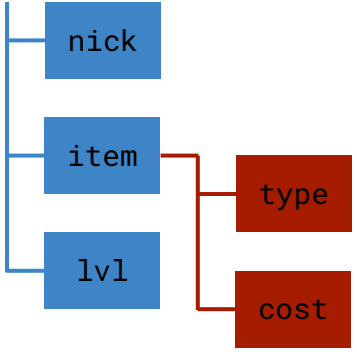
Inspired by xR2RML mixed-syntax paths

```
<LS> a rml:LogicalSource;
  rml:referenceFormulation rr:JSONPath;
  rml:field [
    rml:name “item”;
    rml:reference “$.items[*]”;
    rml:field [
      rml:referenceFormulation ql:CSV;
      rml:name “type”;
      rml:reference “0”]].
```

```
{“characters”:[
  { “nick”: “Ash”,
    “items_and_costs”: [
      “sword,4400”,
      “gloves,340”]},
  { “nick”: “Misty”,
    “items_and_costs”: [
      “gloves,340”,
      “mittens,400”,
      “hat,800”]}]
```

IDLab
INTERNET & DATA LAB

Fields define a nested iteration



nick	item.type	item.cost	lvl
Ash	gloves	340	13
Ash	sword	4440	13
Misty	gloves	340	21
Misty	mittens	300	21
Misty	hat	800	21

RML Fields

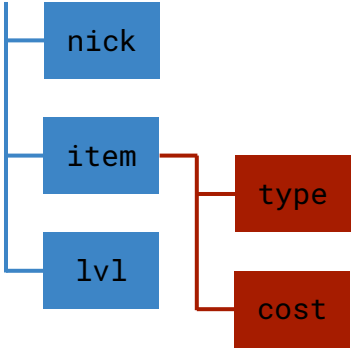
Nested data

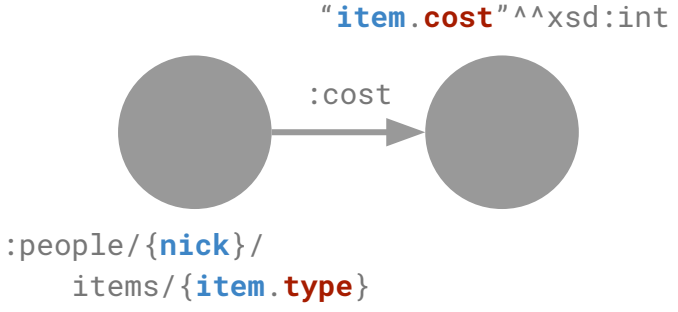
RML extension

- Extracting information
- Writing RDF
- Syntactic sugar

Mapping challenges

RML term maps use field names to create RDF

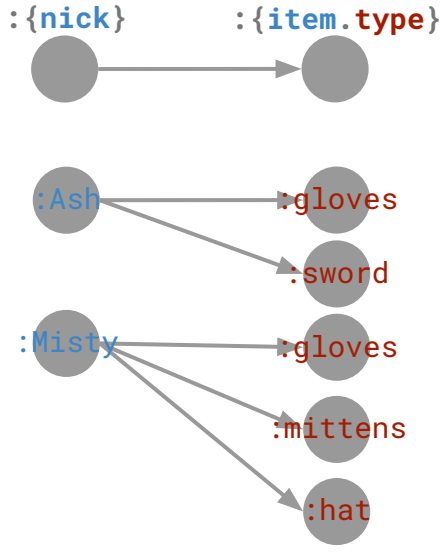




IDLab
INTERNET & DATA LAB

Per “triple template”, one triple is created per iteration

nick	item.type	item.cost	lvl
Ash	gloves	340	13
Ash	sword	4440	13
Misty	gloves	340	21
Misty	mittens	300	21
Misty	hat	800	21



IDLab
INTERNET & DATA LAB

Is performance affected?

High redundancy in denormalized iterations

Engines could make clever execution plans

nick	item.type
Ash	gloves
Ash	sword
Misty	gloves
Misty	mittens
Misty	hat

RML Fields

Nested data

RML extension

- Extracting information
- Writing RDF
- Syntactic sugar

Mapping challenges

IDLab
INTERNET & DATA LAB

RML references can be seen as syntactic sugar for RML with non-nested fields

RML with fields

```
… rml:field [
  rml:reference “$.nickname”; rml:name “nick” ].
… rml:reference “nick”.
```

RML without fields

```
… rml:reference “$.nickname”
```

IDLab
INTERNET & DATA LAB

RML references can be seen as syntactic sugar for RML with non-nested fields

RML with fields

```
… rml:field [
  rml:reference “$.nickname”; rml:name “nick” ].
… rml:reference “nick”.
```

```
… rml:field [
    rml:reference “$.items[*]”; rml:name “item”;
    rml:field [
      rml:reference “$.type”; rml:name “type” ]].
… rml:reference “item.type”.
```

RML without fields

```
… rml:reference “$.nickname”
```

```
… rml:reference “{$.items[*]}{$.type}”
```

(cannot change reference formulation)

IDLab
INTERNET & DATA LAB

RML Fields

Nested data

RML extension

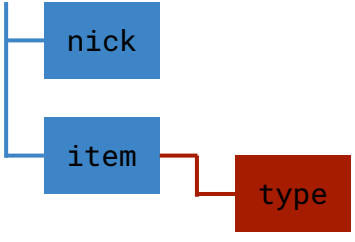
- Extraction
- Representation & writing RDF
- Syntax sugar

Mapping challenges

Challenge 1/5: access fields outside iteration

```
{“characters”:[
  { “nick”: “Ash”,
    “items”: [
      {“type”: “sword”, “cost”: 4400},
      {“type”: “gloves”, “cost”: 340}]},
  { “nick”: “Misty”,
    “items”: [
      {“type”: “gloves”, “cost”: 340},
      {“type”: “mittens”, “cost”: 400},
      {“type”: “hat”, “cost”: 800}]}]}
```

:people/Ash/items/sword



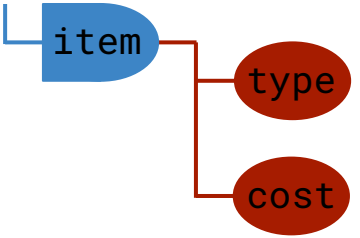
:people/{nick}/
items/{item.type}

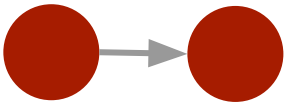
IDLab
INTERNET & DATA LAB

Challenge 2/5: process multivalue references

```
{“characters”:[
  { “nick”: “Ash”,
    “items_and_costs”: [
      “sword,4400”,
      “gloves,340”]}}]
```

```
:sword  :cost 4400 .
:gloves :cost 340 .
```



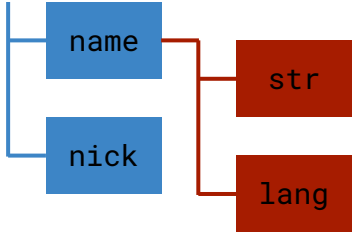


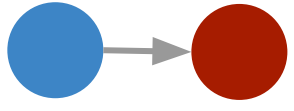
:{item. type} “item.cost” ^^xsd:int

Challenge 3/5: generate multiple values

```
{“characters”:[
  { “nick”: “Ash”,
    “names”: [
      {“name”: “Ash”,
       “lang”: “en”},
      {“name”: “Cendre”,
       “lang”: “fr”}]}]}
```

```
:Ash  :name  “Ash”@en,
             “Cendre”@fr  .
```





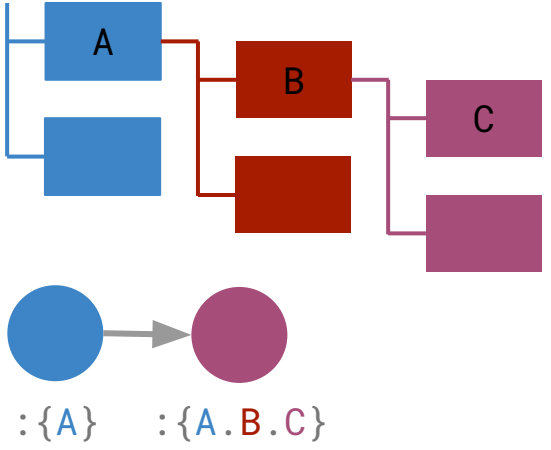
:{nick} “name.str”
@name.lang

IDLab
INTERNET & DATA LAB

Challenge 4/5: multivalue references

```
{“characters”:[
  { “nick”: “Ash”,
    “items”: [
      {“type”: “sword”,
       “elements”: [
         {“type”: “steel”},
         {“type”: “earth”}]},
      {“type”: “gloves”,
       “elements”: [
         {“type”: “grass”}]}]}
```

```
:Ash :elements :steel,
                :earth,
                :grass .
```

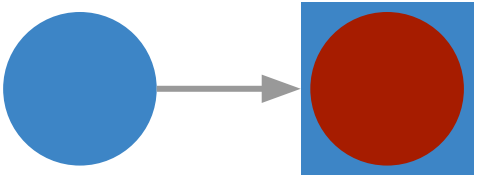


IDLab
INTERNET & DATA LAB

Challenge 5/5: RDF collections

Group values by higher field

```
:Ash    :items ( :sword
                 :gloves ) .
:Misty  :items ( :gloves
                 :mittens
                 :hat ) .
```



:{nick} xrr:nestedTermMap … :{item.type}
… rr:termType xrr:RdfList

IDLab
INTERNET & DATA LAB

RML Fields

Nested data

RML extension

Mapping challenges

RML now has a powerful nested iteration model

Can handle many cases

Shown by mapping challenges

Let’s make the best RDF mapping language possible

Open questions: syntactic sugar, performance,...

IDLab
INTERNET & DATA LAB

Integrating Nested Data into Knowledge Graphs with RML Fields

Thomas Delva, Dylan Van Assche, Pieter Heyvaert, Ben De Meester and Anastasia Dimou

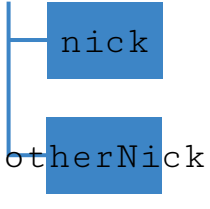
IDLab, Ghent University - imec

Thomas.Delva@ugent.be
@ThomasDelva

Challenge 1/5: access fields outside iteration (sibling)

```
{“characters”:[
  {“nick”: “Ash”},
  {“nick”: “Misty”}
]}

:Ash :companion :Misty
```



IDLab
INTERNET & DATA LAB

Challenge 4/5: process multivalue references (CSV multivalues)

```
{“characters”:[
  { “nick”: “Ash”,
    “items”: “sword,gloves”},
  { “nick”: “Misty”,
    “items”: “gloves,mittens,hat”}]}
```