Using Domain Specific Hierarchical Good Practice for Ranking Service Composition

Anca Marginean, Ioan Alfred Letia, Sergiu Zaporojan

September 22, 2014
Outline

1 Fundamentals
   - Notation N3 and N3Logic
   - RESTdesc - Formal Representation of Services

2 Domain-Specific Hierarchical Model
   - GoodPractice for Primitive Tasks
   - GoodPractice for Compound Tasks

3 Candidate Compositions Analysis
   - Operational semantic of primitive tasks
   - Operational semantic of compound tasks
   - Compositions Ordering

4 Conclusions
### Notation N3 and N3Logic

<table>
<thead>
<tr>
<th>symbol</th>
<th>ex:book</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>?a</td>
</tr>
<tr>
<td>universal variable</td>
<td>@forall :x, :y</td>
</tr>
<tr>
<td>existential variable</td>
<td>@forsome :x, :y</td>
</tr>
<tr>
<td>statement</td>
<td>ex:product ex:price 100</td>
</tr>
<tr>
<td>blank node</td>
<td>[ex:name “George”]</td>
</tr>
<tr>
<td>quoted formula</td>
<td>{ex:product ex:hasPrice 100}</td>
</tr>
<tr>
<td>statement with quoted formula</td>
<td>ex:client :says { :product :hasPrice :high}</td>
</tr>
</tbody>
</table>

**Table:** N3 elements
N3 descriptions

Example of offer in N3 notation

@prefix gr: <http://purl.org/goodrelations/v1#> .

ex:offer
   a gr:Offering ;
   gr:hasBusinessFunction gr:LeaseOut ;
   gr:hasPriceSpecification ex:rental_price1 ,
                           ex:rental_price2 .

Price description

ex:rental_price1
   a gr:UnitPriceSpecification ;
   gr:hasCurrency "USD"^^xsd:string ;
   gr:hasCurrencyValue "2"^^xsd:float ;
   gr:hasUnitOfMeasurement "DAY"^^xsd:string ;
   gr:valueAddedTaxIncluded "true"^^xsd:boolean .
N3 rules

Example of rules

\{
=> { ?book ex:quality ex:high } (1)
\}

\{
=> false. (2)
\}

ex:john ex:hasFriend _:f. (3)

Reasoners: CWM, Euler (EYE)
Listing 1: GetTopic service
Anca Marginean, Ioan Alfred Letia, Sergiu Zaporojan

Using Domain Specific Hierarchical Good Practice for Ranking Services

**RESTdesc**

```json
{
  ?topic a dbpedia:Book .
  ?topic dbpedia2:name ?name
}
=>
{
  _:request http:methodName "GET";
  tmpl:requestURI ("/englishBook/" ?name);
}.
```

Listing 2: GetEnglishVersion service

```json
{
  ?photo flicker:id ?id.
  ?photo foaf:primaryTopic _:work }
=>
{
  _:work dbpedia:author ?author. }
```

Listing 3: Rule for user goal
Domain-Specific Hierarchical Model

Good Practice:
Hierarchical Rules with Importance value

Infringed Rules.
Infringement degree.
Partial Order over compositions

Candidate composition
N3 reasoning Module
GoodPractice for Primitive Tasks

- \langle \text{name}(o), \text{precond}(o), \text{effects}(o) \rangle
- \text{effects}(o) = \bigcup_i \langle \text{cond}(i), \text{effect}(i), \text{value}(i) \rangle
- \text{value}(i) \in V \text{ with partial order: } v_i \succ v_j

1) Task \ htn : hasPrecond \ ?Precond
2) Task \ htn : hasCons \ r_i.
3) r_i \ htn : hasCond \ ?Cond.
4) r_i \ htn : hasEffect
   \{ _: \text{request} \ http : \text{resp} \ ?Response. \}
   \text{ConstraintsGraph}_i[?Response].
5) r_i \ htn : hasValue \ \alpha_i.
6) r_i \ htn : hasComment \ TextualDescription.
GoodPractice for Compound Tasks

Method $m = \langle name(m), task(m), precond(m), network(m) \rangle$

$CT \ htn : isDescribedBy$

$\{[htn : hasContext Context_i;$

$htn : hasDecomp(T_i^1, T_i^2, \ldots, T_i^i)]\}$
Example: Good Practice for e-Commerce

**Figure:** Description of GoodPractice for *getOffer* primitive task through three rules
Cont. Example - Primitive Task

@prefix gr: <http://purl.org/goodrelations/v1#> .
:getOffer htn:hasConstraint :r1;
    htn:hasConstraint :r2;
    htn:hasConstraint :r3 .
:r1 htn:hasPrec {_:product a gr:SomeItems} .
:r1 htn:hasConc
    {_:request http:body _:product;
        http:resp _:offering.
    _:offering a gr:Offering.
    _:offering gr:includes _:product .
    _:offering gr:hasBusinessFunction gr:Sell .
    _:offering gr:hasInventoryLevel [a gr:QuantitativeValueFloat]}.
:r1 htn:hasComment "The response should include a selling offer for a product and it should indicate the stock quantity".
Other two rules and their value

:r2 :hasPrec {_:product a gr:SomeItems; _:product a : notDigital}.
:r2 htn:hasComment "The response should include offer's availability time."

:r3 htn:hasComment "The response should include the eligible regions of an offering".

:r1 htn:hasImpDeg :alpha.
:r3 htn:hasImpDeg :alpha.
:r2 htn:hasImpDeg :beta.
:alpha > :beta.
Part of Good Practice for a buying Compound task

```
:buy  htn:isDescribedBy
    [htn:hasContext  {_:person  a :RegisteredClient};
     htn:hasDecomp  (:search :authenticate :pay)].
```

```
:buy  htn:isDescribedBy
    [htn:hasContext  {_:person  a :NewClient};
     htn:hasDecomp  (:search :setPersonalData :pay)].
```
Candidate Compositions Analysis: Representation

- RESTdesc compositions ⇒ candidate composition \( C \) that includes \( S_1, S_2, ..., S_n \)
- Representation: \( C \ htn : includes \ S_i \)
- Semantics: the conjunction of the semantic of each service

\[
\{\_\_d \ e:findall (?Sem \{ \ ?Cand : includes ?S .
  ?S log:semantics ?Sem\} \\
  ?List). \\
\]
Operational semantic of primitive tasks

When a task is proposed for execution, its rules become active and the candidate composition that is checked must satisfy all these rules.

\[
\text{\{ :state } \text{ htn:do } ?\text{PrimitiveTask} .
\text{ ?PrimitiveTask htn:hasConstraint } ?\text{rule } \}
\Rightarrow
\text{\{ :state } \text{ htn:hasActive } ?\text{rule} \} .
\]

Inference of infringement

Infringement of a rule is true iff antecedents are true but the consequences are false.

4 ?C log:notIncludes ?Conseq \}
5 => {?C htn:infringe ?rule}
Inference rule for choosing a decomposition

1. \{\text{state } htn:do \ ?M. \ ?M htn:isDescribedBy \ ?Descr \}.
5. \Rightarrow \{\text{state } htn:do ?Decomp\}.

Execution of a decomposition

A recursive process for execution of all the included task

\# the one element list
 \ ?L rdf:rest rdf:nil \} 
\Rightarrow \{\text{state } htn:do \ ?Act\}.
\# the recursive access to the list
 \ ?L rdf:rest ?R \} 
\Rightarrow \{\text{state } htn:do \ ?Act . \ : \text{state } htn:do \ ?R\}.
Compositions Ordering

- In possibilistic logic \( ? \), a world \( \omega \) is all the less possible as it falsifies formulas of higher degree.

- A candidate composition is less recommendable as it infringes a rule with higher importance degree.

- Infringement degree \( \text{InfD} \) is the maximum degree of the rules from the set of infringed rules \( \text{InfR} \)

\[
\text{InfR}(C) = \{ r \mid (BP \cup SS) \models ^\pi \{ C \ htn : \text{infringe} \quad r \} \}
\]

- A composition \( C_1 \) is preferred to composition \( C_2 \) stated as \( C_1 \succ C_2 \) iff \( \text{InfD}(C1) < \text{InfD}(C2) \).

- A composition \( C \) is considered the most preferred composition iff it is safe and there is no other candidate composition better than it \( \nexists C_i \text{ s.t. } C_i \succ C \).
user goal involves the tasks *getOffer*.

three candidates are found with: \( \text{InfR}(C_1) = \{r1\} \), \
\( \text{InfR}(C_2) = \{r2\} \), and \( \text{InfR}(C_3) = \{r2, r3\} \)

\( r1 \): stock quantity, \( r2 \): availability time, \( r3 \): eligible regions

\( r1, r2 \rightarrow \alpha, r3 \rightarrow \beta, \alpha > \beta \)

\( C_2 \succ C_1 \) and also \( C_2 \succ C_3 \),
Conclusions

- A hierarchical model for Good Practice in service compositions
- A method in Notation N3 and N3Logic for using this knowledge in analysis of service compositions
  - Identification of infringed rules
  - Order relation over composition based on the infringed rules
- A hierarchical process that can easily be adapted for hierarchical task networks