

- Generating **Surrogates** to Make the Semantic Web **Intelligible** to End-Users

Fabien Gandon

INRIA – ACACIA Lab.

Nice – Sophia Antipolis, FRANCE

■ Introduction

- Semantic Web = **Web augmented** with formalized knowledge annotating it; machine-understandable for automation, interoperability, etc.
- Intelligibility problem for formal knowledge:
 - Need to dynamically **generate interfaces** for conceptual structures coming in contact with the users;
 - Gap between **conceptual structures and user-interface**;
 - Tension **generic** interfaces vs. **customized** interfaces;
- Generating representations for conceptual structures:
 - Dynamic selection of conceptual patterns;
 - Focus on initialization phase of **surrogates**.

Query Matching vs. Intelligible Results

SPARQL-like **query**

01	?d	rdf:type	ex:Document
02	?d	ex:author	?a
03	?a	rdf:type	ex:Person
04	?a	ex:name	?n
05	?n	~	"aiman"

HTML-formatted **result**

- **Novel** (<http://isbn.nu/0380789035>)
 - author** Man (<http://www.neilgaiman.com/>)
 - name:** Gaiman
- **Article** (<http://www.asee.org/jee/papers/content.cfm?name=STEPHEN-209.pdf>)
 - author** Woman (<http://www.mgt.ncsu.edu/faculty/busmgt/laiman-smith.html>)
 - name:** Aiman-Smith

Information = yes there **exist**

- a novel written by a man called “Gaiman”;
- an article written by a Woman called “Aiman-Smith”.

Title, first name, etc. = **natural requirements** \Rightarrow automatically included;

The usefulness of these properties is **ontological knowledge**.

■ Plan of the talk

- Introduce the notion of **surrogates**
 - A motivating scenario; ✓
 - Two meanings in information retrieval.
- Link between **ontologies and interfaces**
 - Needs for semiotic logics of representations;
 - Features of a surrogate and identity conditions;
 - Deriving candidate surrogates from rules.
- Different **tests** and implementations
 - Early attempts;
 - Latest approach based on equivalence rules.



■ Surrogate problem in information retrieval

- **Duality** of the use surrogates of information resources
 - First kind : highly synthetic and representative structure reduced to features of the resource that are **relevant for the intended processing** (e.g. a vector of terms);
 - Second kind : highly synthetic and representative structure reduced to features relevant **for the users to identify the resources**. (e.g. title, extracted text, thumbnail, etc.).
- First kind influences **performances** ; second kind influences **usability**; both critical.
- Problem: how to we **build** these surrogates?
- The **relevance** of features is domain-dependent.

■ Ontology and Interfaces

- Ontologies are at the **frontier** between the system conceptualization & users' conceptualisation:
 - to be understandable both to humans and to machines;
 - we focus and user-interfaces must focus with us.
- User-interfaces to bridge the **gap between internal conceptualizations and day-to-day use**:
 - support **interpretation** of models, inferences and results.
 - choosing labels introduces ontologies in field of **semiotics**;
 - interface & ontological problems to be tackled in **parallel**;
- Build surrogates as & from ontological knowledge:
 - e.g. person : **name, surname, ~~age, height~~**;
 - distinguish **key vs. non key** properties.

■ Identity Conditions and Surrogates

- Identity of instances (e.g. acquaintance networks).
- Representation is tightly **linked to identity** : recognize / distinguish an instance from other instances by means of **unique characteristic properties**.
- Identity relies on a rigid property [Guarino & Welty]
 - Rigid means necessarily **holds for all the instances**;
 - e.g. person vs. student;
 - Group of properties that establish the identity;
 - e.g. family name + first name plus + date of birth.
- Use **overlap** between properties used in the identity condition and properties used in its surrogate.



■ Building surrogates

■ Surrogate properties are **inherited along the hierarchy**
e.g. first name relevant for person and for woman;

■ Production rules to automate the detection of **equivalences**

```
01  IF    [Person: ?p1] -> (name) -> ?n
02      -> (firstname) -> ?f
03      -> (birthdate) -> ?d
04  AND  [Person: ?p2] -> (name) -> ?n
05      -> (firstname) -> ?f
06      -> (birthdate) -> ?d
07  THEN [Person: ?p1] -> (equivalent) -> [Person: ?p2]
```

name, firstname and birthdate for surrogate of person

■ Rules encoding sufficient conditions of concept **definitions**

```
01  IF    [Person: ?p] - (govern) - [Country: ?c]
02  THEN [President: ?p]
```

govern relevant for surrogate of president

■ Union of all candidate templates to **build maximal template:**

president may use govern, name, firstname and birthdate

■ Plan of the talk

■ Introduce the notion of **surrogates**

- A motivating scenario; ✓
- Two meanings in information retrieval. ✓

■ Link between **ontologies and interfaces**

- Needs for semiotic logics of representations; ✓
- Features of a surrogate and identity conditions; ✓
- Deriving candidate surrogates from rules. ✓

■ Different **tests** and implementations

- Early attempts;
- Latest approach based on equivalence rules.

■ Early attempts (1 & 2)

- (1) Complete queries with **optional parts** to display
 - Burden on the users
 - Possible only for queries authored by a human
 - Evaluations showed that users were expecting the system to **volunteer additional information** when available.
- (2) Systematically add **all** properties

```
01  ?d      rdf:type          ex:Document
02  ?d      ex:author       ?a
03  ?a      rdf:type          ex:Person
04  ?a      ex:name         ?n
05  ?n      ~                "aiman"
06  ?d      option::cos:Property ?p1
07  ?a      option::cos:Property ?p2
```

- May retrieve too many properties and generate **noise**
- May not be sufficient (URI & **holophrasting**)

■ Early attempts (3)

■ (3) Try to identify relevant properties in the ontology

- Top-property `surrogate_property` for surrogate properties
- e.g. `designation` sub-property of `surrogate_property` and parent property of `title`, `name`, `firstname`

```
01  ?d      rdf:type      ex:Document
02  ?d      ex:author    ?a
03  ?a      rdf:type      ex:Person
04  ?a      ex:name      ?n
05  ?n      ~            "aiman"
06  ?d      option::ex:surrogate_property ?p1
07  ?a      option::ex:surrogate_property ?p2
```

- Additional **burden in the engineering** and maintenance of the ontology.
- Some properties (e.g. `date`) are relevant for some concept types (e.g. `document`) but **not for some other types** (e.g. an organization).

■ Still using this technique when only option.

■ Need: attach templates of surrogates to every type...

■ Latest Attempt Using Rules (person)

■ Equivalence of instances: **Person**

```
01 <cos:rule>
02   <cos:if>
03     <ex:Person rdf:about="?person1">
04       <ex:name>?name</ex:name>
05       <ex:firstname>?firstname</ex:firstname>
06       <ex:birthdate>?birthdate</ex:birthdate>
07     </ex:Person>
08     <ex:Person rdf:about="?person2">
09       <ex:name>?name</ex:name>
10       <ex:firstname>?firstname</ex:firstname>
11       <ex:birthdate>?birthdate</ex:birthdate>
12     </ex:Person>
13   </cos:if>
14   <cos:then>
15     <ex:Person rdf:about="?person1">
16       <owl:sameAs rdf:resource="?person2" />
17     </ex:Person>
18   </cos:then>
19 </cos:rule>
```

```
01 <cos:rule>
02   <cos:if>
03     <ex:Person rdf:about="?person1">
04       <ex:email>?email</ex:email>
05     </ex:Person>
06     <ex:Person rdf:about="?person2">
07       <ex:email>?email</ex:email>
08     </ex:Person>
09   </cos:if>
10   <cos:then>
11     <ex:Person rdf:about="?person1">
12       <owl:sameAs rdf:resource="?person2" />
13     </ex:Person>
14   </cos:then>
15 </cos:rule>
```

■ Latest Attempt Using Rules (doc)

■ Equivalence of instances: **Document**

```
01 <cos:rule>
02   <cos:if>
03     <ex:Document rdf:about="?doc1">
04       <ex:title>?title</ex:title>
05       <ex:author rdf:resource="?author" />
06       <ex:date>?date</ex:date>
07     </ex:Document>
08     <ex:Document rdf:about="?doc2">
09       <ex:title>?title</ex:title>
10       <ex:author rdf:resource="?author" />
11       <ex:date>?date</ex:date>
12     </ex:Document>
13   </cos:if>
14   <cos:then>
15     <ex:Document rdf:about="?doc1">
16       <owl:sameAs rdf:resource="?doc2" />
17     </ex:Document>
18   </cos:then>
19 </cos:rule>
```

■ Latest Attempt Using Rules (templates)

- XSLT stylesheet to **transform rules into surrogate templates**

```
01 <ex:Person>
02   <ex:name>?name</ex:name>
03   <ex:firstname>?firstname</ex:firstname>
04   <ex:birthdate>?birthdate</ex:birthdate>
05 </ex:Person>
06
07 <ex:Person>
08   <ex:email>?email</ex:email>
09 </ex:Person>
10
11 <ex:Document>
12   <ex:title>?title</ex:title>
13   <ex:author rdf:resource="?author"/>
14   <ex:date>?date</ex:date>
15 </ex:Document>
```

- Merge templates targeting same types

■ Latest Attempt Using Rules (result)

■ SPARQL-like query

01	?d	rdf:type	ex:Document
02	?d	ex:author	?a
03	?a	rdf:type	ex:Person
04	?a	ex:name	?n
05	?n	~	"aiman"

■ HTML-formatted result

- Novel (<http://isbn.nu/0380789035>)
title: American Gods
date: April 30, 2002
author Man (<http://www.neilgaiman.com/>)
name: Gaiman
first name: Neil
- Article (<http://www.asee.org/jee/papers/content.cfm?name=STEPHEN-209.pdf>)
title: Algorithm for High Technology Engineering and Management Education
author Woman (<http://www.mgt.ncsu.edu/faculty/busmgt/laiman-smith.html>)
name: Aiman-Smith
first name: Lynda
e-mail: lynda_aiman-smith@ncsu.edu

■ Summary

- Represent conceptual structures (e.g. query results);
- Users will not enter each and every representation;
- Early attempt to initialize template and just a test:
 - Parallel between **surrogates and identity conditions**;
 - **Relevant features as ontological knowledge**;
 - Integration rules to implement first tests;
 - Equivalence criteria as good surrogate properties;
 - Necessary conditions as good surrogate properties.
- Opportunistic approach:
 - Candidate template = union of candidate properties;
 - Further differentiation requires human intervention

■ Discussion (1/2)

- Did not take into account the **way properties are rendered** (text, graphics, speech, etc.);
- Quality of surrogates can **deteriorate when complex** equivalence rules are introduced;
- Semi-automatic way:
 - Reviewed and **tuned by the ontologists** or
 - Refined /selected by **learning techniques**.
- Subtypes frequently require fewer properties (e.g. Economist John Adam Smith)
 - Additional properties may be poor surrogate e.g. the ISBN;
 - Maximum candidate \neq perfect candidate.
- Explore **hierarchy of surrogates**

■ Discussion (2/2)

- Recursive application of templates
 - **Holophrasting** to request additional information;
 - Start a **disambiguation dialog**;
 - Promising **heuristic** :
 - (1) apply full surrogate template to each node of the query;
 - (2) apply the “data type properties” to the added resources;
 - (3) holophrasting technique for “object type properties”.
- Improving visualization \neq 100% unicity in surrogates;

■ Perspectives

- Other **sources** than rules & other **formalization** means;
- Need for a richer model of the links between **conceptual structures and semiotic level**;
- Need to model influence of **pragmatic** context;
- Dynamic generation of interfaces for the semantic web: conceptual structures + ...
 - the users' **profile**;
 - the **context** and history of interactions;
 - **semiotic modelling** primitives in meta-models;
 - signs **linked to ontological** primitives;
 - **logics of semiotics** and surrogate generation;

■ Thank you.



Fabien.Gandon
@sophia.inria.fr