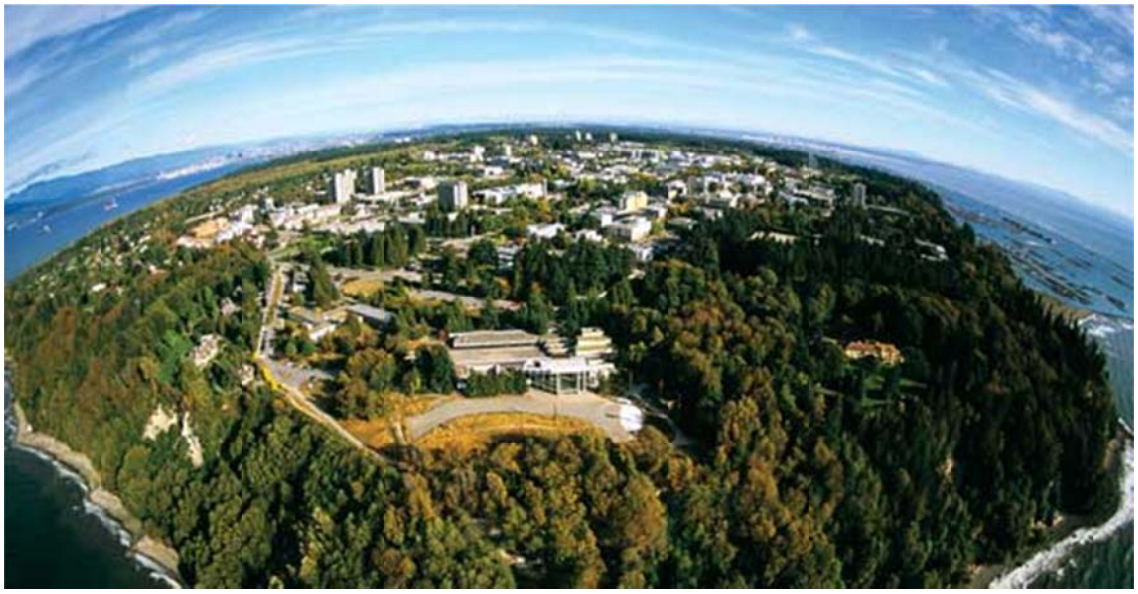


Christopher J. O. Baker, Helen Chen,

Ebrahim Bagheri, Weichang Du (Eds.)

CSWS2011

Proceedings of the 3rd Canadian Semantic Web Symposium

Vancouver, British Columbia, Canada

August 5, 2011

Organization Committee:

Christopher Baker, University of New Brunswick (Saint John)
Helen Chen, University of Waterloo
Ebrahim Bagheri, Athabasca University
Weichang Du, University of New Brunswick

Program Committee:

Abdolreza Abhari	Ryerson University, Canada
Atif Khan	University of Waterloo, Canada
Alexandre Riazanov	University of New Brunswick, Canada
Arash Shaban-Nejad	McGill University, Canada
Babak Esfandiari	Carleton University, Canada
Bruce Spencer	National Research Council Canada, Canada
Christopher Baker	University of New Brunswick, Canada
Dragan Gašević	Simon Fraser University, Canada
Ebrahim Bagheri	Athabasca University, Canada
Faezeh Ensan	University of British Columbia, Canada
Fred Popowich	Simon Fraser University, Canada
Hassan Ait-Kaci	IBM Canada, Canada
Helen Chen	University of Waterloo, Canada
Leo Ferres	Universidad de Concepción, Chile
Marek Hatala	Simon Fraser University, Canada
Marek Reformat	University of Alberta, Canada
Mark Wilkinson	University of British Columbia, Canada
Marina Sokolova	University of Ottawa, Canada
Michel Dumontier	Carleton University, Canada
René Witte	Concordia University, Canada
Vio Onut	IBM Canada, Canada
Volker Haarslev	Concordia University, Canada
Weichang Du	University of New Brunswick, Canada
Weiming Shen	National Research Council, Canada
Yevgen Biletskiy	University of New Brunswick, Canada

Preface

The Third Canadian Semantic Web Symposium (CSWS2011) is a follow-up to the previous symposia (CSWWS 2006 and CSWWS 2009). It aims at bringing together Canadian and international researchers in semantic technologies and knowledge management to discuss about various issues related to the Semantic Web.

We thank all authors for submitting to CSWS2011, the program committee for the help in the review process for the symposium program.

Christopher Baker, Helen Chen, Ebrahim Bagheri, Weichang Du

CSWS2011 Organization Committee

Table of Contents

The Social Semantic Subweb of Virtual Patient Support Groups

Harold Boley[1], Omair Shafiq[2], Derek Smith[3], and Taylor Osmun[3]

[1] Institute for Information Technology, National Research Council Canada
Fredericton, NB, Canada, harold.boley AT nrc.gc.ca
[2] Department of Computer Science, University of Calgary
Calgary, AB, Canada, moshafiq AT ucalgary.ca
[3] Faculty of Computer Science, University of New Brunswick
Fredericton, NB, Canada, {i14dy, w91pq} AT unb.ca

Abstract. Patients increasingly interact in support groups, which provide shared information and experiences about diseases, treatments, etc. Much of this interaction is mediated by the Social Web, allowing worldwide reach but lacking in semantic precision. We present an online prototype, PatientSupporter, to create a Social Semantic Subweb that will facilitate high-precision networking between patients based on ontologies and rules. PatientSupporter is an instantiation of Rule Responder that permits each patient to query other patients' profiles for finding or initiating a matching group. Rule Responder's External Agent (EA) is a Web-based patient-organization interface that passes queries to the Organizational Agent (OA). The OA represents the shared knowledge of the virtual patient organization, delegates queries to relevant Personal Agents (PAs) via a responsibility ontology, and passes checked PA answers back to the EA. Each PA represents the medical subarea of primary interest to an associated patient group. The PA assists its patients by advertising their interest profiles, employing rules about diseases and treatments as well as interaction constraints such as time, location, age range, gender, and number of participants. Profiles can be distributed across different rule engines using different rule languages (e.g., Prolog and N3), where rules, queries, and answers are interchanged via translation to and from RuleML/XML. We discuss the implementation of PatientSupporter in a use case of sports injuries structured by a part-whole ontology of affected body parts.

1 Introduction

Social Web (Web 2.0) techniques have been explored in recent years for applications in healthcare [14, 13]. Web 2.0 portals such as PatientsLikeMe[1] and (part of) samestory[2] have been developed to help patients to network with other, geographically distributed patients having similar ailments to discuss and exchange

[1] http://www.patientslikeme.com/
[2] http://www.same-story.com/sante-maladies/

information and experiences. Successful 'Patient 2.0' portals typically have good recall when searching for other patients but lack in precision.

Semantic techniques increase this networking precision, leading to the following Social Semantic Web (Web 3.0) approach to patient portals. We introduce ontologies and rules for organizing patients – here, with sports injuries – into virtual support groups around classes of an ontology of injuries – here, a commonsense partonomy for localizing sports injuries. Of course, this can only *complement* the diagnosis and therapy of diseases by medical experts – it reflects new patients' use of commonsense knowledge, rather than expert knowledge, to *find* similar patients as well as relevant literature and medical professionals.

Our initial online prototype, PatientSupporter[3], is designed to start the Social Semantic Subweb for patients by demonstrating how patients with a sports injury could be helped to find or initiate a virtual support group about that injury. Patients in an online PatientSupporter virtual organization create their semantic profile referring to classes in a disease ontology – here a partonomy of body parts affected by sports injuries. This body partonomy allows patients to base the description of their injuries on a `subPartOf` hierarchy leading to affected body parts, which is implemented as a corresponding `subClassOf` taxonomy of injury classes for those body parts. Profiles contain rules about body-part diseases and treatments as well as interaction constraints such as time, location, age range, gender, and number of participants. A patient can pose queries against the semantic profiles of other patients in his or her virtual organization to find or initiate a matching group. PatientSupporter is built upon Rule Responder [19, 9], which has also been used, e.g., in the related Social Semantic Web instantiation WellnessRules [8] and in SymposiumPlanner [12].

PatientSupporter allows patients to have their profiles expressed in either Pure Prolog [20] (Horn logic rules) or N3 [2] (graph production rules). Providing these quite different rule language paradigms permits virtual organizations or individual patients to base their PatientSupporter use on the paradigm that best suits them. Rule Responder handles the interoperation between different rule languages of patients through translators to and from RuleML/XML as the interchange format [10].

As an example, let us consider a patient with nickname Paul, who has injured part of his left leg during a rugby game. He has questions about his lesion and precautions for recovery, which others with similar lesions may be able to answer or help with. Since he lives in a small town where he knows no one else with such an injury he looks for online support.[4] Using PatientSupporter, Paul poses a query through the External Agent (EA), focusing on leg lesions. The EA submits the query to the Organizational Agent (OA), which delegates it to the Personal Agent (PA) of the leg-injury group, and checks the answers from the profiles (local knowledge bases) of its participating patients. When the OA returns many answers to the EA, Paul discovers that his query was too broad.

[3] http://ruleml.org/PatientSupporter/

[4] Other reasons for seeking support in a virtual rather than real group may include increased anonymity (via nicknames) and avoiding contagiousness (e.g., flu).

Paul, who actually hurt his left knee, thus proceeds downward the partonomy by querying PatientSupporter just for patients with knee lesions. Since no knee-injury PA exists, the OA again delegates it to the more general leg-injury PA. But within this PA's group only the knee-injury participants are eligible, hence the answers returned are less in number and more relevant. Paul picks some of the returned nicknames of patients and queries them with his interaction constraints, proposing a knee-injury discussion in a Skype-based conference call on the upcoming Saturday or Sunday any time between 10AM and 6PM EST. Paul's query returns the Skype IDs of patients who want to reveal theirs and are interested in the discussion, with the time narrowed down to Sunday between 3PM and 6PM EST. Hence, Paul invites them for a first call Sunday, 4PM to 5PM, effectively initiating a knee-injury subgroup of the leg-injury group.

It should be noted that Paul by using the PatientSupporter Social Semantic Web portal is able to initiate the virtual subgroup about his sports injury on a global scale. He also benefits from PatientSupporter's interoperation facility in the background – to transform patient profiles between Pure Prolog and N3 through RuleML/XML. The system employs a partonomy of sports-injury-affected body parts, which makes it easy for Paul to navigate hierarchically up or down, increasing recall or precision, respectively. Paul's queries invoke other patients' interaction rules, allowing him to narrow down his search in a step-wise fashion. All of this saves him from browsing through a large set of irrelevant patient profiles and permits him to efficiently converge on a first Skype call.

The rest of the paper is organized as follows. Section 2 presents the design goals of the PatientSupporter instantiation of Rule Responder. Section 3 discusses the global knowledge base used by the OA. Section 4 describes the use of local knowledge bases to represent the profiles of individual patients underneath the PAs. Section 5 expands upon the RuleML-based interoperation of Pure Prolog and N3 rules. Section 6 explains and demonstrates the use of RuleML-based querying for patients in a distributed setting. Section 7 concludes the paper and discusses future work.

2 Instantiating Rule Responder to PatientSupporter

PatientSupporter is based on Rule Responder, where this reference architecture with each of its main agent types (i.e., EA, OA, and PAs) is instantiated as described in the following. It performs virtual support group matchmaking by querying patients organized in a disease ontology – here, a body partonomy. The following design goals have been pursued while developing PatientSupporter:

1. Identify a language of appropriate expressiveness to model patient profiles from the family of RuleML languages [3], Pure Prolog [20], N3 [2], etc.
2. Identify a language for light-weight ontologies of sports injuries such as `subPartOf` partonomies mapped to `subClassOf` taxonomies in RDFS or OWL 2. The ontology language is to be combined with the rule language.

3. Allow eliciting rule and ontology definitions in human-oriented syntaxes, while translating the resulting knowledge to and from RuleML/XML, RIF/XML [7], or XCL2 / CL RuleML[5] for interchange.
4. Allow different rule engines (e.g., OO jDREW[6] [1], Prova[7] [15], and Euler[8]) to execute global and local rulebases.
5. Allow rules as well as queries and their answers to be transmitted over an Enterprise Service Bus (ESB) – e.g., Mule[9] or Apache ServiceMix.
6. Investigate the appropriateness of languages, engines, and GUIs for rules as well as ontologies, to express, process, and transform the knowledge required in patient profiles.
7. Elicit exemplary patient profiles and abstract them to generally usable profile templates for increased usability and reusability.
8. Guide students – e.g., of Computer Science, Medicine, or Kinesiology – when forming and evolving virtual sports-injury support groups with PatientSupporter.
9. Evaluate the effectiveness and usefulness of the distributed PatientSupporter architecture, based on its ESB-interconnected engines using different languages for the dynamic formation of virtual patient support groups.
10. Adapt PatientSupporter from the sports-injury use case to other medical domains such as weight control, food allergies, oral health, or (seasonal) flu.

The current implementation of PatientSupporter has focused on goals 1.-7. It models patient profiles in POSL [5] and N3 [2]. The profiles are interoperated through RuleML/XML as the intermediate format. Enquiring users are aided by an English-XML-bridging menu-based form.[10] Knowlege about patients and their injuries is organized using rules combined with light-weight ontologies in sorted (typed) Horn logic or N3. The `subPartOf` partonomy is mapped to `subClassOf` in RDFS.[11] Human-oriented syntaxes (of POSL and N3) have been used while modeling the patient profiles. The overall communication and coordination of the rule engines (e.g., OO jDREW, Prova, and Euler) has been organized through Mule, an open-source ESB. The use of an ESB allows architectural flexibility by decoupling the functional components of Rule Responder from the communication components [11].

Rule Responder's instantiation to PatientSupporter in the sports-injury domain allows to create virtual patient organizations consisting of virtual support groups that are defined through sports injuries structured by a partonomy of affected body parts (further explained in Section 3). Specifically, the OA becomes an assistant to the entire virtual patient organization. Each PA becomes an assistant to a group of patients having the same class of injuries from the partonomy,

[5] http://wiki.ruleml.org/index.php/Relax_NG
[6] http://www.jdrew.org/oojdrew/
[7] http://www.prova.ws/
[8] http://eulersharp.sourceforge.net/
[9] http://www.mulesoft.org/
[10] http://ruleml.org/PatientSupporter/RuleResponder.html
[11] http://ruleml.org/PatientSupporter/files/PS-Taxonomy.rdf

and helps them as *profile users* to get organized as a support group. The EA is utilized by patients as *enquiry users* to (register with its virtual organization and) query the profiles of the virtual organization's other patients.

Rule Responder employs the following sequence of steps: An enquiry user interacts with the EA to author and submit queries to the OA. The OA assigns (maps and delegates) each query topic to the PA most knowledgeable about it. Each PA poses the query to its local rulebase, and returns the derived answer(s) to the OA. The OA checks the answer(s) before giving them back to the EA, hence to the enquiry user.

By default, the OA does not reveal the identity of the nicknamed patient(s) behind the answering PA(s). Keeping the personal information hidden in this way, the OA acts as a mediator that helps protect the privacy of profiles of patients in the virtual organization. For participating in PatientSupporter-scheduled online discussions via Skype, MSN, etc., or via a (smart)phone, patients might also use dedicated (Skype, MSN, etc.) IDs or phones. However, if support group participants do not want to reveal even their voice, they have to resort to typing via chat or SMS. On the other hand, after a few discussions some within the same virtual support group may decide to reveal their everyday identities to selected or all participants.

Rule Responder's earlier instantiations include SymposiumPlanner [12] and WellnessRules [8], where WellnessRules extended Rule Responder with multiple participant profiles underneath each PA. PatientSupporter further extends the functionality of Rule Responder by making Social Semantic use of partonomies, mapped to taxonomies: Patient injuries are classified in the hierarchical part-whole manner of affected body parts. For example, injuries related to Foot are subordinated to those of Leg. Similarly, Heel and Toe injuries are subordinated to those of Foot. Employing partonomies as light-weight ontologies in this way allows the 'refinement' of a virtual support group (e.g., about Leg injuries) into subgroups (e.g., about Foot injuries and further about Heel and Toe injuries).

PatientSupporter in extension to WellnessRules allows users to be supported by PAs as follows: Each patient (as a profile user) publishes a profile employed by the responsible PAs to respond to queries (from enquiry users) about his or her preferences, constraints, etc. This dynamic profile association is implemented via the Profile Responsibility Matrix (PRM), and is not possible in SymposiumPlanner, where only chairs (as profile users) are supported by PAs.

The main agent types of PatientSupporter are described in the following subsections. Figure 1 depicts the interaction between the EA, OA, and PAs.

2.1 External Agent

The External Agent (EA) is the point-of-contact that allows a patient to query the Organizational Agent (OA) of a virtual patient organization. It is based on a Web interface that allows him or her as an enquiry user to compose queries employing a menu-based form, which uses JavaScript to generate both an English description and RuleML/XML, thus making it easy to query other patients' profiles. A sports-injury patient primarily selects the injury class from the parton-

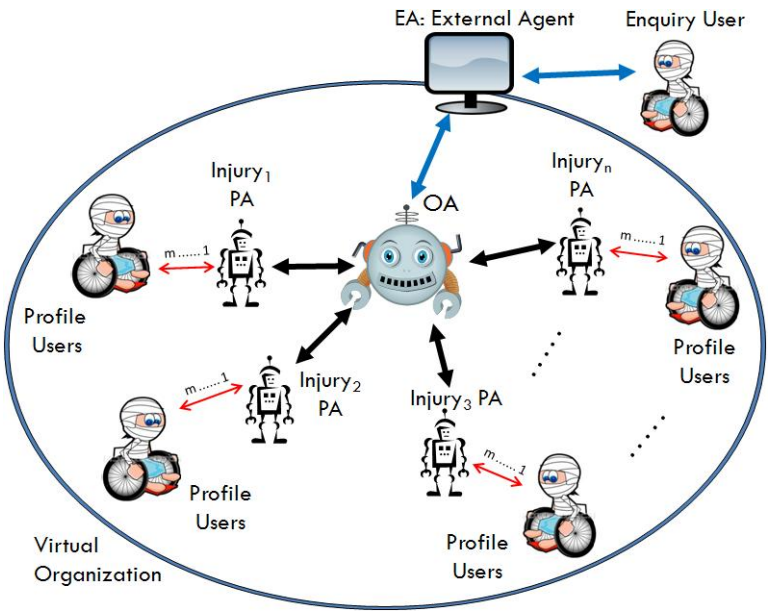


Fig. 1. Overall architecture of PatientSupporter

omy. He or she can then fill in property values about diseases and treatments as well as interaction constraints. The finished RuleML/XML query is submitted to the OA. Finally, the EA presents the OA's answer(s) to the patient.

2.2 Organizational Agent

The Organizational Agent (OA) is at the center of PatientSupporter, representing a virtual patient organization as a whole. The knowledge base of the OA is *global* across the virtual organization, and is written and run in the language and engine Prova. The OA also employs two matrices, on the basis of which incoming queries are mapped and delegated: The *Group Responsibility Matrix*, written as an OWL-Lite ontology, defines which group headed by a PA is best for which kind of query. The *Profile Responsibility Matrix*, written as an XML document, defines which patient profiles exist in a PA's group, and in which formats (here, POSL or N3).

2.3 Personal Agent

The Personal Agents (PAs) contain disease-oriented groups of patient profiles, where diseases are restricted to sports injuries. Each PA heads the group of patient profiles listed in the Profile Responsibility Matrix (cf. Section 2.2). The knowledge base of facts and rules of each profile under a PA is *local* to that profile, and is either written in POSL and run in OO jDREW or is written in N3 and run in Euler.

3 Global Knowledge Base for Virtual Organization

The ontologies and a subset of the rules are globally shared via the OA to benefit all the PAs. Another subset of rules is distributed amongst the PAs, where it is kept local (cf. Section 4). The shared ontology and the shared subset of rules are referred to as the *global knowledge base*, which is complemented by a shared *signature document*.

Global knowledge in PatientSupporter is modeled as a combination of ontologies and rules, where rule arguments are defined by signatures. The ontologies include a light-weight ontology realizing the Group Responsibility Matrix (cf. Section 2.2) and a body partonomy. The global rules include general constraints and preferences of the virtual organization. PatientSupporter makes use of the standard rule format RuleML/XML and the Rule Responder framework to transform to and from other rule languages.

The body partonomy was elicited as a commonsense ontology to reflect the patient-centric perspective of support groups. It is drawing, among others, on the Digital Anatomist Foundational Model of Anatomy (FMA) [18], the online Sports Injury Clinic [21], and the knowledge of a medically trained NRC-IIT colleague. It is referred to as a partonomy because it represents the logical hierarchy of body parts. However, it is realized as a taxonomy of injuries affecting the body parts. Thus, *A* `subPartOf` *B* implies *A Injury* `subClassOf` *B Injury*. Note that we do not express *what* (possibly still undiagnosed) injury it is, but only the (unlateralized etc.) body part *where* it is. This representation is proposed as an appropriate level of abstraction for finding patients with sports injuries, but could be refined for other purposes (cf. Section 7).

Under the root `Body`, PatientSupporter uses the partonomy classes `Head`, `Neck`, `Shoulder`, `Arm`, `Torso`, `Back`, and `Leg`. All of `Thigh`, `Lower Leg`, `Knee`, and `Foot` are regarded as direct parts of `Leg`. `Toe` and `Heel` are likewise part of `Foot`. The complete partonomy is shown in Figure 2. Its implementation as a `subClassOf` taxonomy in RDFS is available online.[11]

The rule component in PatientSupporter employs POSL with Horn logic plus Negation as failure (Naf) and N3 with scoped Naf. The use of Naf Hornlog POSL has been restricted to atoms with positional arguments,[12] leaving F-logic-like frames with property-value slots to N3. This demonstrates the range of our approach through complementary rule styles.[13]

The Naf Hornlog POSL sublanguage uses (positional) n-ary relations (or, predicates) as its central modeling paradigm. N3 instead uses (unordered) sets of binary slots (or, properties) centered around object identifiers (OIDs, called 'subjects' in RDF and N3).

[12] The POSL syntax thus corresponds to pure-Prolog syntax except that POSL variables are prefixed by a question mark while Prolog variables are upper-cased.

[13] To didactically exemplify the positional and slotted styles as well as POSL-N3 interoperation, the online PatientSupporter prototype redundantly keeps rulebases both as `.posl` and as `.n3` documents.

The following POSL example indicates the *positional signature* of the 16-ary predicate `myDiscussion`:

```
myDiscussion(?ProfileID,?Injury,?MinAge,?MaxAge,?MinRSVP,?MaxRSVP,?Category,?Treatment,
             ?HealingStage,?StartTime,?EndTime,?Duration,?Channel,?Contact,?Gender,?TimeZone).
```

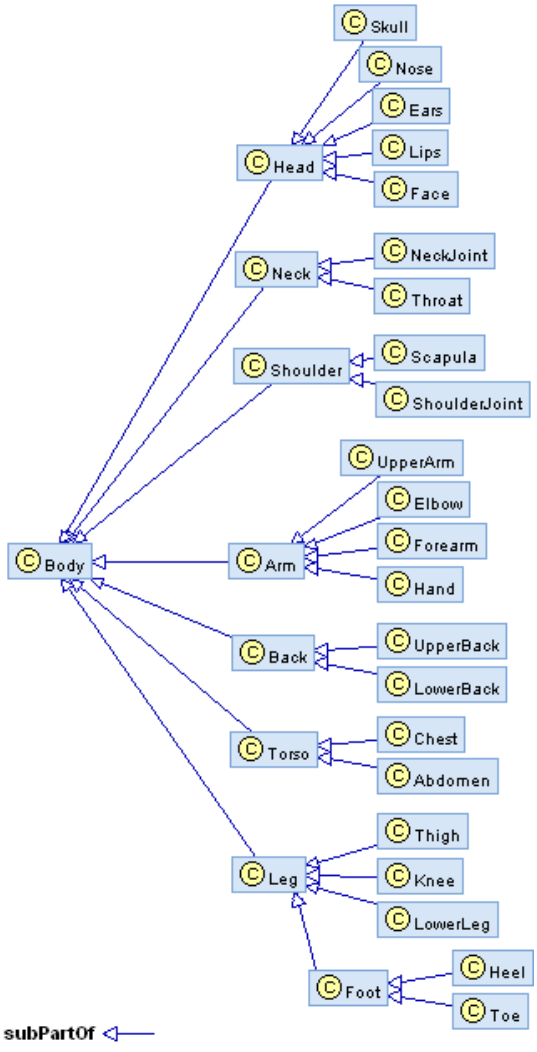


Fig. 2. Patient-centric body partonomy for localizing sports injuries

In N3 this becomes a *slotted signature* with subject `_:myDiscussion`, an `rdf:type` of `:MyDiscussion`, and the 16 arguments as the remaining slots:

```
_:myDiscussion
    rdf:type       :MyDiscussion;
    :profileID     ?ProfileID
    :injury        ?Injury;
    :minAge        ?MinAge;
    :maxAge        ?MaxAge;
    :minRSVP       ?MinRSVP;
    :maxRSVP       ?MaxRSVP;
    :category      ?Category;
    :treatment     ?Treatment;
    :healingStage  ?HealingStage;
    :startTime     ?StartTime;
    :endTime       ?EndTime;
    :duration      ?Duration;
    :channel       ?Channel;
    :contact       ?Contact;
    :gender        ?Gender;
    :timeZone      ?TimeZone.
```

The complete signatures are being maintained in a global document.[14]

While rules (including underlying facts) according to a positional signature are usually more concise, positional arguments must be specified in their fixed order (with missing or inapplicable arguments represented by 'null values'). Conversely, while a slotted signature usually makes rules more verbose, slotted arguments can be specified in any order (with missing or inapplicable arguments just becoming omitted). The Datalog special case of the positional paradigm (i.e., Hornlog without complex arguments) corresponds to the relational model in that facts with the same predicate correspond to relational tables, and rules to relational views. Conversely, the slotted paradigm is a special case of object-oriented models where objects ('subjects') are declaratively described by slots and can inherit slot values, but slot values are not procedurally updated. Depending on their previous experience with these paradigms (e.g., with relational databases or RDF metadata), whose characteristics transpire even in GUIs, virtual organizations or individual patients can take advantage of their favorite one. For a synthesis of the two paradigms see [6].

Shared rules defining PatientSupporter predicates have been collected for the rulebase of the OA. They, together with the signatures and ontologies, formalize the global knowledge of the PatientSupporter system.

An example of a POSL ('backward') rule defines `participation` in a virtual support group as follows:

```
participation(?ProfileID,?Injury,?MinRSVP,?MaxRSVP) :-
     groupSize(?ProfileID,?Injury,?Min,?Max),
     greaterThanOrEqual(?MinRSVP,?Min),
     lessThanOrEqual(?MaxRSVP,?Max).
```

The first argument of the conclusion predicate `participation` is the patient (`?ProfileID`) the rule is instantiated for, followed by a lesion (`?Injury`) argument, followed by the minimal (`?MinRSVP`) and maximal (`?MaxRSVP`) number of

[14] http://ruleml.org/PatientSupporter/Signatures/

participants the querier wants to have in a group for the lesion. The rule succeeds for its four positional arguments if `?ProfileID`'s desired group size (`groupSize`) is between `?Min` and `?Max`, `?MinRSVP` $\geq$ `?Min`, and `?MaxRSVP` $\leq$ `?Max`.

The corresponding N3 ('forward') rule for deriving `_:participation` facts is as follows, where the `?rsvpQuery` premise does not correspond to a premise of the POSL rule but is needed to bind the 'input' arguments of its conclusion:

```
{
 ?rsvpQuery
     rdf:type    :RSVPQuery;
     :profileID  ?ProfileID;
     :injury     ?Injury;
     :minRSVP    ?MinRSVP;
     :maxRSVP    ?MaxRSVP.

 ?groupSize
     rdf:type    :GroupSize;
     :profileID  ?ProfileID;
     :injury     ?Injury;
     :min        ?Min;
     :max        ?Max.

 ?MinRSVP math:notLessThan ?Min.

 ?MaxRSVP math:notGreaterThan ?Max.
}
=>
{
 _:participation
     rdf:type    :Participation;
     :profileID  ?ProfileID;
     :injury     ?Injury;
     :minRSVP    ?MinRSVP;
     :maxRSVP    ?MaxRSVP.
}.
```

The global OA knowledge base is being maintained in both language paradigms,[13] i.e. POSL[15] and N3[16].

4 Locally Distributed Knowledge Bases for Patients

Locally distributed knowledge bases are grouped as profiles underneath the PAs. Each PA group has its own kind of knowledge base, according to the medical subarea associated with the body partonomy (cf. Section 3). For example, the profiles created by patients with `Leg` injuries are kept with the `Leg` PA.

The local knowledge bases have information about patients to model their profiles using the following vocabulary of properties: A unique identifier of a profile, `ProfileID`; the kind of injury of the patient, `Injury`; the age of the patient, `Age`; the time zone of the the patient, `TimeZone`; the treatment required, `Treatment`; the stage of healing of the injury, `HealingStage`; and the category information, `Category`. The properties `Treatment`, `HealingStage`, and `Category` have the following allowed value ranges: The `Treatment` property currently has one of the values `Bandage`, `MajorOperation`, `MediumOperation`,

[15] http://ruleml.org/PatientSupporter/resources/OA/PS-Global.posl
[16] http://ruleml.org/PatientSupporter/resources/OA/PS-Global.n3

MinorOperation, MajorMedication, MediumMedication, MinorMedication, or ChangeOfLifeStyle; the HealingStage property has values Fresh, Medium, Convalescent, or Healed; and the Category property has values In or Out patient.

For example, this is a local myDiscussion fact about p0001 according to the positional signature of Section 3 (in POSL we use ?:Leg as an anonymous variable of type Leg, assuming p0001 has one leg injury):

```
myDiscussion(p0001,?:Leg,20:integer,50:integer,5:integer,10:integer,Out,Bandage,
             Medium,
             dateTime[2011:integer,6:integer,1:integer,10:integer,15:integer],
             dateTime[2011:integer,6:integer,1:integer,11:integer,20:integer],
             dateTime[0:integer,0:integer,0:integer,0:integer,30:integer],
             Skype,John27,Male,-400).
```

Similarly, given below is its counterpart according to the slotted signature (in N3 we use :Leg as a constant, again standing for one leg injury):

```
:myDiscussion_1
    rdf:type       :MyDiscussion;
    :profileID     :p0001;
    :injury        :Leg;
    :minAge        :20;
    :maxAge        :50;
    :minRSVP       :5;
    :maxRSVP       :10;
    :category      :Out;
    :treatment     :Bandage;
    :healingStage  :Medium;
    :startTime     [:year 2011; :month 6; :day 1; :hour 10; :minute 15];
    :endTime       [:year 2011; :month 6; :day 1; :hour 11; :minute 20];
    :duration      [:year 0; :month 0; :day 0; :hour 0; :minute 30];
    :channel       :skype,
    :contact       :John27,
    :gender        :Male,
    :timeZone      :-400.
```

Both express interest in a myDiscussion about Leg injuries, with Medium stage of healing, Bandage level for treatment, and with category of Out patient. It is proposed for June 1st, 2011, between 10:15 AM and 11:20 AM (GMT -4:00 Atlantic Time) for a duration of 30 minutes. It should have the form of a Skype call for 5 to 10 people. The Skype user name of the person advertising this time is John27.

This fact (in POSL and N3) can be generated by a local rule (again, in both paradigms) which uses another rule and facts to satisfy its premises. Given below is an example of a positional POSL rule from the PA knowledge base of patient p0001, defining the main predicate myDiscussion about Leg injuries, specifying his desired support-group discussion:

```
myDiscussion(p0001,?:Leg,?MinAge,?MaxAge,?MinRSVP,?MaxRSVP,?Category,?Treatment,?HealingStage,
                    dateTime[?StartYear,?StartMonth,?StartDay,?StartHour,?StartMinute],
                    dateTime[?EndYear,?EndMonth,?EndDay,?EndHour,?EndMinute],
                    dateTime[?DurYear,?DurMonth,?DurDay,?DurHour,?DurMinute],
                                                  ?Channel,?Contact,?Gender,?TimeZone) :-
  ageCheck(p0001,?MinAge,?MaxAge,?Age),
  participation(p0001,?:Leg,?MinRSVP,?MaxRSVP),
  communication(p0001,?Channel,?Contact),
  notEqual(?Channel,MSN),
```

```
  event(p0001,?:Leg,Possible,
    dateTime[?StartYear,?StartMonth,?StartDay,?StartHour,?StartMinute],
    dateTime[?EndYear,?EndMonth,?EndDay,?EndHour,?EndMinute]),
  duration(p0001,?:Leg,dateTime[?DurYear,?DurMonth,?DurDay,?DurHour,?DurMinute]),
  goodDuration(p0001,?:Leg,
    dateTime[?DurYear,?DurMonth,?DurDay,?DurHour,?DurMinute],
    dateTime[?StartYear,?StartMonth,?StartDay,?StartHour,?StartMinute],
    dateTime[?EndYear,?EndMonth,?EndDay,?EndHour,?EndMinute]),

  category(p0001,?:Leg,?Category),
  treatment(p0001,?:Leg,?Treatment),
  healingStage(p0001,?:Leg,?HealingStage),

  gender(p0001,?Gender),
  timeZone(p0001,?TimeZone).
```

The rule conclusion's `myDiscussion` predicate starts with the person's profile ID, `p0001`, followed by the kind of injury, `?:Leg`, the age limits `?MinAge` and `?MaxAge`, the group limits `?MinRSVP` and `?MaxRSVP`, the patient `?Category` (`In`/`Out`), the `?Treatment` and `?HealingStage` to be discussed, the start time, end time, and duration of the discussion, its `communication ?Channel`, as well as `p0001`'s `?Contact` name, `?Gender`, and `?TimeZone`.

The initial rule premises perform an `ageCheck`, test the `participation` constraints (cf. rule in Section 3), and filter on the `communication ?Channel`. The next premises query `p0001`'s `Possible ?:Leg`-injury `event`s and planned `duration` for this discussion, making sure it is a `goodDuration` within the event interval. The penultimate premises compute the `category`, `treatment` and `healingStage` constraints. The final premises concern the `gender` and `timeZone`.

The corresponding slotted N3 rule is given, abridged, below:

```
{
 ...

 ?event
  rdf:type       :Event;
  :profileID     :p0001;
  :injury        :Leg;
  :tense         :Possible;
  :startDateTime [:year ?StartYear; :month ?StartMonth; :day ?StartDay; :hour ?StartHour;
                                                                    :minute ?StartMinute];
  :endDateTime   [:year ?EndYear; :month ?EndMonth; :day ?EndDay; :hour ?EndHour;
                                                                      :minute ?EndMinute].

 ...
}
=>
{
 _:myDiscussion
  rdf:type       :MyDiscussion;
  :profileID     :p0001;
  :injury        :Leg;
  :minAge        ?MinAge;
  :maxAge        ?MaxAge;
  :minRSVP       ?MinRSVP;
  :maxRSVP       ?MaxRSVP;
  :category      :Out;
  :treatment     ?Treatment;
  :healingStage  ?Stage;
  :startDateTime [:year ?StartYear; :month ?StartMonth; :day ?StartDay; :hour ?StartHour;
                                                                    :minute ?StartMinute];
  :endDateTime   [:year ?EndYear; :month ?EndMonth; :day ?EndDay; :hour ?EndHour;
                                                                      :minute ?EndMinute];
```

```
  :duration      [:year ?DurYear; :month ?DurMonth; :day ?DurDay; :hour ?DurHour;
                                                                    :minute ?DurMinute];
  :channel       ?Channel;
  :contact       ?Contact;
  :gender        ?Gender;
  :timeZone      ?TimeZone.
}.
```

The POSL and N3 rules (and facts) for this and other fictitious patients are available, as templates, online.[17]

5 Interoperation between POSL and N3 Rules via RuleML/XML

The PatientSupporter use case includes a testbed for the interoperation (i.e., alignment and translation) of information in knowledge bases in the two main rule paradigms: Prolog-style (positional) relations and N3-style (slotted) frames. PatientSupporter inherits the interoperation mechanisms from Rule Responder. The interoperation methodology makes iterative use of alignment and translation: An initial alignment permits the translation of parts of a hybrid knowledge base. This then leads to more precise alignments, which in turn lead to better translations. Using this methodology, PatientSupporter can maintain relational (Pure Prolog) as well as frame (N3) versions of rules, both accessing the same, independently maintained, body partonomy.

The PAs of PatientSupporter can thus use either of these rule paradigms, while interoperation is carried out through the intermediate rule language RuleML/XML, which has sublanguages for both of them, so that the cross-paradigm translations can use the common XML syntax of RuleML. A pair of online converters[18] is used for rulebase conversion between the human-oriented POSL syntax and its XML serialization in RuleML.

For rulebase translation, the signatures of PatientSupporter relations and frames are aligned in a shared signature document,[14] discussed in Section 3, which specifies the argument positions of relations and slot names of frames. The alignment of sample relations and frames in Sections 3 and 4 then suggests the actual translations between the two rule paradigms.

Translations that are considered to be 'static' or 'at compile-time' take an entire rulebase as input and return its entire transformed version in RuleML/XML. Thus, an assumption of 'closed-arguments' of fixed signatures for relations and frames is made [8].

Positional-slotted translators for a version of RuleML are available online as an XSLT implementation.[19]

For example, POSL's `myDiscussion` relational fact of Section 4 is serialized in positional RuleML as follows, where `Ind`ividual constants are distinguished from `Data` literals:

[17] http://ruleml.org/PatientSupporter/resources/PA/
[18] http://ruleml.org/posl/converter.jnlp
[19] http://ruleml.org/ooruleml-xslt/oo2prml.html

```
<Atom>
  <Rel>myDiscussion</Rel>
  <Ind>p0001</Ind>
  <Var type="Leg"/>
  . . .
  <Data>Out</Data>
  <Ind>Bandage</Ind>
  <Data>Medium</Data>
  . . .
</Atom>
```

Extending the mappings in OO RuleML[20], N3's `myDiscussion` frame fact of Section 4 is serialized in slotted RuleML as follows, where RuleML's `Rel` represents N3's `rdf:type`:

```
<Atom>
  <oid><Ind iri=":myDiscussion_1"/></oid>
  <Rel iri=":MyDiscussion"/>
  <slot>
    <Ind iri=":profileID"/>
    <Ind>p0001</Ind>
  </slot>
  <slot>
    <Ind iri=":injury"/>
    <Ind>Leg</Ind>
  </slot>
  . . .
  <slot>
    <Ind iri=":category"/>
    <Data>Out</Data>
  </slot>
  <slot>
    <Ind iri=":treatment"/>
    <Ind>Bandage</Ind>
  </slot>
  <slot>
    <Ind iri=":healingStage"/>
    <Data>Medium</Data>
  </slot>
  . . .
</Atom>
```

While slotted-to-positional translation of atoms essentially fixes the argument order and omits the slot names, positional-to-slotted translation looks up the slot names in the shared signature document.[14] For the translation of a rule, the above translation of atoms is applied to the atom in the conclusion and to all the atoms in the premises. For a rulebase, the translation then applies to all of its rules. With the above-discussed human-oriented syntax translators, rulebases containing rules like the `myDiscussion` rule in Section 4 can thus be translated from Pure Prolog to POSL to RuleML (positional to slotted) and to N3, as well as vice versa. These translators permit rule, query, and answer interoperation, via RuleML/XML, for the Rule Responder infrastructure inherited by PatientSupporter.

The translators have been complemented by mappings between the Dlex subset of RuleML and of RIF [4].

[20] http://ruleml.org/indoo/n3ruleml.html

6 Distributed Rule Responder Querying

PatientSupporter inherits the distributed query mechanism from Rule Responder. For querying different rule engines, transformations between queries and answers from N3 and Pure Prolog through RuleML/XML are done as described for rules in Section 5. Both the global knowledge base, described in Section 3, and locally distributed knowledge bases, described in Section 4, are used in query answering.

Given below is an example of a POSL query for patient profiles, which is executed by Rule Responder's OO jDREW TD (Top-Down) engine:

```
myDiscussion(?ProfileID,?Injury:Leg,20:integer,50:integer,5:integer,10:integer,...)
```

It uses the rule from Section 4 to find any patient `(?ProfileID)` who has a `Leg` injury, age between `20:integer` and `50:integer`, and is interested in joining a discussion group with minimum `5:integer` to maximum `10:integer` people, where all the remaining arguments, indicated by '`...`', are left open as free variables.

This is the corresponding N3 query, to be executed by Rule Responder's EulerSharp EYE bottom-up engine:

```
@prefix : <patient_profiles#>.
@prefix rdf: <http://www.w3.org/1999/02/22-
            rdf-syntax-ns#>.

_:myDiscussion
  rdf:type        :MyDiscussion;
  :profileID      ?ProfileID;
  :injury         :Leg;
  :minAge         :20;
  :maxAge         :50;
  :minRSVP        :5;
  :maxRSVP        :10;
  ... .
```

After having declared two prefixes, it builds an existential ('_') node, `_:myDiscussion`, using slots for the fixed parameters and the fact-provided `?MinRSVP` (5) and `?MaxRSVP` (10) bindings to fill the variable slots again indicated by '`...`'.

Within our online test environment, the above sample query produces twelve solutions. These can be narrowed down to produce four solutions by descending the partonomy from `?Injury:Leg` to `?Injury:Foot`, and to two solutions when `?Injury:Foot` becomes `?Injury:Toe`. Using variations of such queries, patients-as-enquiry-users of PatientSupporter will be able to explore profiles of patients-as-profile-users to find or initiate a support group.

In our experiments, the overall processing times for the online-selectable[10] positional `myDiscussion` Example Queries 1-4 in Rule Responder instantiated to PatientSupporter on average were, respectively, 11s (for 12 answers), 7s (for 5 answers), 5s (for 2 answers), and 4s (for 1 answer), measured for Java JRE6 in Windows XP on an Intel Core 2 Duo 2.80GHz processor.

The implemented Rule Responder instantiation for PatientSupporter, with all source files and test queries, is available online.[10]

7 Conclusions and Future Work

PatientSupporter demonstrates the Social Semantic Subweb for patients who want to collaborate and share information with each other about sports injuries, on a global scale. It enables precise networking between patients by using ontologies combined with rules to specify and query patient profiles. Key features of PatientSupporter are: First, it permits interoperation of patient profiles between Pure Prolog (Naf Hornlog) and N3 through RuleML/XML. Second, it enables scalability of distributed knowledge on the Social Semantic Subweb via its PA modularization, starting with derivation rules and light-weight ontologies. Third, PatientSupporter uses the OO jDREW, Euler, and Prova engines, while its open Rule Responder architecture makes it easy to bring in new engines. Fourth, it makes use of a body partonomy for modeling sports injuries in a hierarchical manner (from the patients' commonsense point-of-view). Fifth, it makes use of ontologies and rules to precisely search for patient profiles, and allows enquiry users to narrow down their search in a step-wise fashion. Hence, by delivering the relevant profiles, PatientSupporter saves enquiry users from the hassle of browsing through a large set of patient profiles.

While the querying of patient rulebases by enquiry users is pretty well supported with a menu-based GUI[10], the editing of patient rulebases by profile users should be similarly supported. Especially for newcomers, the choice between positional, slotted, and combined language paradigms could be abstracted away as far as possible: A new profile user would visually select the features relevant for their profile and the underlying system would generate the profile in the language most appropriate for this selection. In future work, controlled or natural language interfaces could be developed for both querying and editing, following the ACE query interface of SymposiumPlanner-2011 [22], and information extraction methods could be explored as an alternative to 'from-scratch' profile editing.

In a separate effort, PatientSupporter's current vocabulary of properties could be refined – and rules over them could be written – to express multiple injuries to the same body part, injuries of multiple body parts, indirect symptoms, lab results, as well as specifics about diagnoses and therapies. Besides the treatment of injuries, their prevention could be represented.

An extension of PatientSupporter could include – along with the patients and their Social Semantic profiles – medical professionals and Personal Health Records (PHRs) [17]. This would assist in the formation of virtual support groups consisting of doctors and nurses as well as patients, based on the preferences and constraints of all three subgroups. For this, patients' commonsense knowledge and profiles should be mapped to medical expert knowledge and PHRs – and (partially) vice versa. For example, PatientSupporter's body partonomy for localizing global knowledge about sports injuries could (be mapped to a full-blown medical ontology such as SNOMED and) act as an index into medical knowledge about anatomy, physiology, etc. as it pertains to sports injuries. A medical professional could then provide injury-specific knowledge that is not patient-specific to the entire support group, rather than repeating it for each patient.

PatientSupporter and its use cases will thus provide new challenges and suggestions for improvements of RuleML, Rule Responder, and the involved engines. Since PatientSupporter rules are interoperated through RuleML/XML, they can also be ported from Rule Responder to other Social Semantic Web frameworks such as EMERALD [16] or be read into another Java-based system via JAXB.

8 Acknowledgements

The authors would like to thank colleagues at NRC-IIT Fredericton, especially Benjamin Craig and Julie Maitland, for helpful discussions. We also thank Zainab Almugbel, Usman Ali Chaudhry, and Sujan Saha, as well as the organizers, reviewers, and participants of CSWS2011 for valuable feedback. NSERC is thanked for its support through a Discovery Grant for Harold Boley. The work by Omair Shafiq, Derek Smith, and Taylor Osmun was done during their stays at NRC-IIT.

References

1. M. Ball, H. Boley, D. Hirtle, J. Mei, and B. Spencer. The OO jDREW Reference Implementation of RuleML. In A. Adi, S. Stoutenburg, and S. Tabet, editors, *Rules and Rule Markup Languages for the Semantic Web, First International Conference (RuleML 2005), Galway, Ireland, November 10-12, 2005, Proceedings*, volume 3791 of *Lecture Notes in Computer Science*, pages 218–223. Springer, 2005.
2. T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler. N3Logic: A Logical Framework For the World Wide Web. *Theory and Practice of Logic Programming (TPLP)*, 8(3), May 2008.
3. H. Boley. Are Your Rules Online? Four Web Rule Essentials. In A. Paschke and Y. Biletskiy, editors, *Proc. Advances in Rule Interchange and Applications, International Symposium (RuleML-2007), Orlando, Florida*, volume 4824 of *LNCS*, pages 7–24. Springer, 2007.
4. H. Boley. RIF RuleML Rosetta Ring: Round-Tripping the Dlex Subset of Datalog RuleML and RIF-Core. In G. Governatori, J. Hall, and A. Paschke, editors, *RuleML*, volume 5858 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2009.
5. H. Boley. Integrating Positional and Slotted Knowledge on the Semantic Web. *Journal of Emerging Technologies in Web Intelligence*, 4(2):343–353, Nov. 2010.
6. H. Boley. A RIF-Style Semantics for RuleML-Integrated Positional-Slotted, Object-Applicative Rules. In N. Bassiliades, G. Governatori, and A. Paschke, editors, *RuleML Europe*, volume 6826 of *Lecture Notes in Computer Science*, pages 194–211. Springer, 2011.
7. H. Boley and M. Kifer. A Guide to the Basic Logic Dialect for Rule Interchange on the Web. *IEEE Transactions on Knowledge and Data Engineering*, Forthcoming 2010.
8. H. Boley, T. M. Osmun, and B. L. Craig. Social Semantic Rule Sharing and Querying in Wellness Communities. In *4th Asian Semantic Web Conference, ASWC 2009, Shanghai, China*, 2009.

9. H. Boley and A. Paschke. Rule Responder Agents: Framework and Instantiations. In A. Elci, M. T. Kone, and M. A. Orgun, editors, *Semantic Agent Systems: Foundations and Applications*. Springer Studies in Computational Intelligence, Vol. 344, 2011.
10. H. Boley, A. Paschke, and O. Shafiq. RuleML 1.0: The Overarching Specification of Web Rules. In *Proc. 4th International Web Rule Symposium: Research Based and Industry Focused (RuleML-2010), Washington, DC, USA, October 2010*, Lecture Notes in Computer Science. Springer, 2010.
11. D. Chappell. Enterprise Service Bus. OReilly: June 2004, ISBN 0-596-00675-6.
12. B. L. Craig and H. Boley. Personal Agents in the Rule Responder Architecture. In N. Bassiliades, G. Governatori, and A. Paschke, editors, *RuleML*, volume 5321 of *Lecture Notes in Computer Science*, pages 150–165. Springer, 2008.
13. G. Eysenbach. Medicine 2.0: Social Networking, Collaboration, Participation, Apomediation, and Openness. Jul-Sep 2008.
14. B. Hughes, I. Joshi, and J. Wareham. Health 2.0 and Medicine 2.0: Tensions and Controversies in the Field. 2008.
15. A. Kozlenkov. Prova Rule Language Version 3.0 User's Guide. http://www.prova.ws/etc/Prova 3.0 User Guide.pdf, May 2010.
16. K. Kravari, T. M. Osmun, H. Boley, and N. Bassiliades. Cross-Community Interoperation between the EMERALD and Rule Responder Multi-Agent Systems. In N. Bassiliades, G. Governatori, and A. Paschke, editors, *RuleML Europe*, volume 6826 of *Lecture Notes in Computer Science*, pages 44–51. Springer, 2011.
17. L. S. Liu, P. C. Shih, and G. R. Hayes. Barriers to the Adoption and Use of Personal Health Record Systems. In *Proc. iConference, Seattle, WA, USA*, pages 363–370. ACM, February 2011.
18. J. L. V. Mejino, A. V. Agoncillo, K. L. Rickard, and C. Rosse. Representing Complexity in Part-Whole Relationships within the Foundational Model of Anatomy. http://sig.biostr.washington.edu/projects/fm/FME/.
19. A. Paschke, H. Boley, A. Kozlenkov, and B. Craig. Rule Responder: RuleML-Based Agents for Distributed Collaboration on the Pragmatic Web. In *2nd ACM Pragmatic Web Conference*. ACM, 2007.
20. L. Sterling and E. Y. Shapiro. *The Art of Prolog - Advanced Programming Techniques, 2nd Ed.* MIT Press, 1994.
21. M. Walden, H. Mills, and T. Dekkers. Sports Injury Clinic. http://www.sportsinjuryclinic.net/cybertherapist/injurylist.htm.
22. Z. Zhao, A. Paschke, C. U. Ali, and H. Boley. Principles of the SymposiumPlanner Instantiations of Rule Responder. In F. Olken, M. Palmirani, and D. Sottara, editors, *Rule-Based Modeling and Computing on the Semantic Web*. LNCS 7018, Springer, 2011.

Leveraging SADI Semantic Web Services to exploit fish ecotoxicology data

Matthew M. Hindle[1], Alexandre Riazanov[1], Edward S. Goudreau[2], Christopher J. Martyniuk[2], Christopher J. O. Baker[1]

[1] Department of Computer Science & Applied Statistics
[2] Canadian Rivers Institute and Department of Biology
University of New Brunswick, Saint John, New Brunswick, E2L 4L5, Canada
{hindlem, alexr, t969c, cmartyn, bakerc}@unb.ca

Abstract. In order to interpret experimental Omics-data, ecotoxicologists are faced with an array of disconnected bioinformatics databases and algorithms. These include tools for microarray analysis, gene annotation, functional gene set enrichment, and network analysis. Drawing together these Web tools and resources is a frequently labour-consuming technical exercise in identifying links across database records and the connecting input and output formats of tools. Interpreting experimental Omics-data in the context of the current available knowledge and methodologies from a single query platform with explicit semantics would be a valuable asset for toxicology in the analysis of DNA, transcriptomics, proteomic, and metabolomic experimental data.

Methods: We have created 30+ SADI semantic Web Services, resources and tools pertinent to the interpretation of Omics toxicological data. These services encompass a wide range of algorithms, domains and databases, including sequence alignment and protein domain finding tools (e.g. BLAST, HMMR3, and InterProScan), databases containing experimentally validated protein functions (e.g. ZFIN and MGI), and central repositories of sequence and microarray data (e.g. ArrayExpress and NCBI-RefSeq). All these services can be leveraged through SPARQL queries submitted to the SHARE query engine. This paradigm provides a single access-point on the Web for a toxicologist to submit semantically rich queries that are resolved using the relevant databases and tools. This frees the ecotoxicologist from learning unnecessary details concerning tool interfaces and the semantic idiosyncrasies of databases.

Results: We present a series of example queries for ecotoxicology, which facilitate the interpretation of transcriptomics data in the context of public knowledge and current tools. These queries include common tasks, specific to a user's experimental data set, such as gene ontology annotation of probes on a custom microarray experiment for an aquatic species of interest.

Keywords: SADI, SHARE, Semantic Web Services, Ecotoxicology, Fish, Toxicology

1 Introduction

Toxicology is increasingly a systems discipline, requiring the analysis of multi-scale Omics data [2, 17, 25]. This typically require tools and databases that can be leveraged for tasks such as microarray analysis, gene annotation, functional gene set enrichment, and network analysis. However, in order to meet these requirements toxicologists are faced with a bewildering array of disconnected bioinformatics resources. Drawing together and mastering these Web tools and resources is frequently an unnecessary and frustrating technical exercise in identifying common links across database records and the connecting input and output formats of bioinformatics tools. Interpreting experimental Omics-data in the context of the current available knowledge and methodologies from a single query platform with explicit semantics would be an invaluable asset to ecotoxicologists in the analysis of their DNA, transcriptomics, proteomic, and metabolomic experimental data. Working towards such a unified semantic framework in ecotoxicology, would free toxicologists from wasting time on technical and semantic idiosyncrasies, and enable the environmental toxicologist to synthesize Omics information to better predict risks associated with chemical exposures.

There are many existing approaches to integrating biological data sets. Project like Bio2RDF [4] and Linked Life Data [30] have used semantic technologies to build mash-ups of current biological information. However, many biological application cases also require the integration of bioinformatics tools and algorithms. Semantic Web Service architectures [9, 12, 16, 26] are an elegant solution to exposing both knowledge and algorithms in a semantically explicit framework. Services can be leveraged as components in complex bioinformatics analysis pipelines. This paper presents an initial framework of SADI Web Services for the ecotoxicology domain and example queries which demonstrate how such a framework can be leveraged to retrieve relevant information. Together with existing SADI use-cases [3, 7, 21], these example queries demonstrate the utility of SADI semantic Web Services in solving the problems of resource and tool fragmentation, and semantic heterogeneity in the life sciences.

1.1 What are SADI Web Services?

Most conventional Web Services produce an output without making an explicit semantic connection to the input data. Web Services built using the SADI framework [26] make the semantics of this relationship explicit. SADI is a set of conventions for creating Semantic Web Services, which as a consequence of their explicit semantics, can be automatically discovered and orchestrated. An RDF graph forms the service input and has some URI node designated as a central node. The whole input RDF graph is considered a description of this central node. Exactly the same node is always present in the output RDF graph and becomes the central output node. The sole function of a SADI service is therefore to decorate this central input node with new properties, which are asserted in the output RDF graph.

The classes and properties which a SADI Web Service accepts as input and computes as output must reference a defined input and output class in an OWL [31] ontology. The inputs and outputs of the Web Service are therefore always clearly defined and the behavior of the Web Service is formally specified. For example, Fig. 1 shows a SADI Web Service and an example of the RDF input and output accepted by the service. The input and output classes are defined in the service ontology.

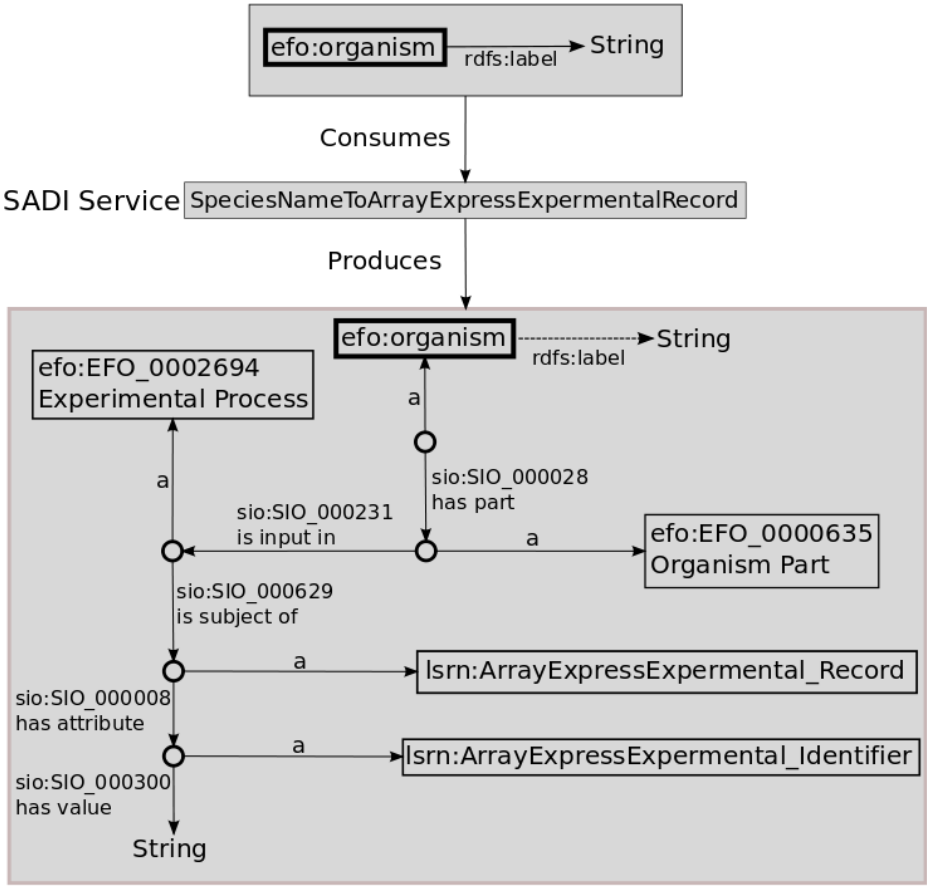


Fig. 1 A SADI service which uses input and output classes from the service ontology: http://unbsj.biordf.net/fishtox/arrayexpress-sadi-service-ontology.owl. The service consumes an OWL class, which must be a subclass of efo:organism and have an attach label. The modeling of the input is defined in ExperimentAnnotatedSpecies_Input. The service decorates the OWL organism class with the individuals of that species that are inputs for microarray experiments in ArrayExpress. The modeling of the output is defined by the ExperimentAnnotatedSpecies_Output class. A key to the prefixes used in this figure can be found in Table 1. The bold framed RDF class indicates the central node in the input and output graph.

The explicit nature of SADI service semantics means they can be automatically enacted by client software. In this paper we use the SHARE client [23], which computes SPARQL queries by picking and calling suitable SADI Web Services from a SADI service registry. Therefore, SPARQL queries can be written based on an understanding of the ontological primitives referred to in service semantic descriptions, available from the registry. However, browsing the services is often a useful exercise as it gives a good idea of what data is available.

1.2 The tools and databases wrapped as SADI Web Services

In order to improve predictive abilities, ecotoxicologists are becoming more interested in the pathways and associations regulated by a specific chemical (*e.g.* adverse pathways of toxicity). In order to provide a core bioinformatics toolbox for ecotoxicology, our initial SADI Web Services provide information pertinent to the analysis of exotoxicology microarrays. Specifically, we prioritize services which facilitate (1) comparison of an experimental dataset with other published transcriptomics data, and (2) sequence transcript information retrieval in the form of Pfam [11] protein domain, and Gene Ontology (GO) functional annotation. These knowledge domains provide the subject for the example queries, described in the results.

An important requirement for the analysis of a fish toxicological dataset is the ability to compare experimental results with existing published data. This has the potential to provide valuable insights into transcriptomics datasets by elucidating similarities and differences with transcriptomes that were subject to similar experimental conditions, such as the concentration of chemical or duration of exposure regime. ArrayExpress [20] is a database of functional genomics experiments which includes a large number of microarrays. It includes data on microarray platforms, as well as data recording individual experiments and their parameters. It provides Web Services that can be readily wrapped with SADI Web Services, which effectively provide a layer that adds explicit semantics.

Another requirement for our ecotoxicology use case is the annotation of microarray sequences with Pfam domains and GO functional annotation. In order to achieve this, SADI Web Services were required that exposed HMMR3 [8] and BLAST functionality. Microarray sequences are often derived from assembled EST sequences, and this is particularly true for the many custom arrays for fish. Consequently sequences may be incomplete and contain missing gene fragments, which introduce shifts in the reading frame. This makes the process of finding the correct open reading frame (ORF), which encodes the protein, challenging. It was therefore a priority to include a ORF prediction tool such as ORF-Predictor [19].

Sequence functional annotation with GO also requires the retrieval of experimentally derived annotations from model organism databases. We prioritized annotations for *Danio rerio* and *Mus musculus*, based on the evolutionary distance to fish and abundance of experimental annotation, respectively.

The Zebrafish Model Organism Database (ZFIN) [6], is the main data repository for the *Danio rerio* genome. *Danio rerio* is one of the most important model organisms for teleost fish and is used as a model for growth and development, pharmacology and toxicology studies [14, 22, 24]. ZFIN contains a repository of reference gene models, together with mappings to most of the sequence repositories. They also contribute a set of experimental and electronically inferred GO annotations for genes.

Mouse Genome Informatics (MGI) [5] is the main data repository for information concerning the *Mus musculus* genome. It contains a list of reference gene models, and external references to the main sequence repositories. It is the largest source of experimentally verified GO annotations for genes, which motivated us to included it as a data source for SADI Web Services.

2 Methods

This section describes the prior modeling and SADI Web Services, which were leveraged by the example queries in this use case. Defining an appropriately expressive model for the RDF that will be consumed and produced by services is crucial for enabling interoperability with other services, and flexible querying.

2.1 Reuse of existing upper and domain Ontologies

In order to improve the re-usability of our SADI Web Services, wherever possible we reference existing upper and domain ontologies. Table 1 lists the ontologies used by the SADI Web Services. The Semanticscience Integrated Ontology (SIO) provides a broad set of classes and properties, and is used extensively by other SADI Web Services. The Life Science Resource Name (LSRN) provides classes for defining database records and identifiers. It also uses the SIO ontology as an upper ontology. SIO and LSRN are our preferential upper ontologies for modeling services. The Experimental Factor Ontology (EFO), provides classes and properties for describing sample variables in experiments [15]. It has been used extensively for the Gene Expression Atlas [13], and in the Semantic Web Atlas Project [1]. We reuse EFO to encourage interoperability with the modeling provided by these projects. Our application ontologies mainly contain input and output class definitions. Where possible we have minimized the creation of any new classes or relations in these service ontologies.

Table. 1 Ontologies and Prefixes used in the SADI Web Services

Prefix	URL	Type
lsrn	http://purl.oclc.org/SADI/LSRN/	Upper
sio	http://semanticscience.org/resource/	Upper
efo	http://www.ebi.ac.uk/efo/	Domain
blastso	http://unbsj.biordf.net/fishtox/BLAST-sadi-service-ontology.owl#	Application
hmmrso	http://unbsj.biordf.net/fishtox/HMMR-sadi-service-ontology.owl#	Application
goaso	http://unbsj.biordf.net/fishtox/GOA-sadi-service-ontology.owl#	Application
microarrayso	http://unbsj.biordf.net/fishtox/arrayexpress-sadi-service-ontology.owl#	Application
tsso	http://unbsj.biordf.net/fishtox/record-translation-sadi-service-ontology.owl#	Application
stso	http://unbsj.biordf.net/fishtox/seq-tools-sadi-service-ontology.owl	Application

2.2 Modeling schematics

Fig 2. shows a schematic of the main classes and properties used to model RDF input and output for SADI Web Services. The schematic can be used to design SPARQL queries for the SHARE client. Some secondary classes and properties, such as BLAST and HMMR alignment scores, have been omitted. Also, the schematic does not show potential connections to classes and properties provided by other published SADI Web Services. The semantic richness of our modeling enables a greater expressiveness in writing SPARQL queries. It also reduces the need to re-model for new use cases when further SADI Web Services are added or become available.

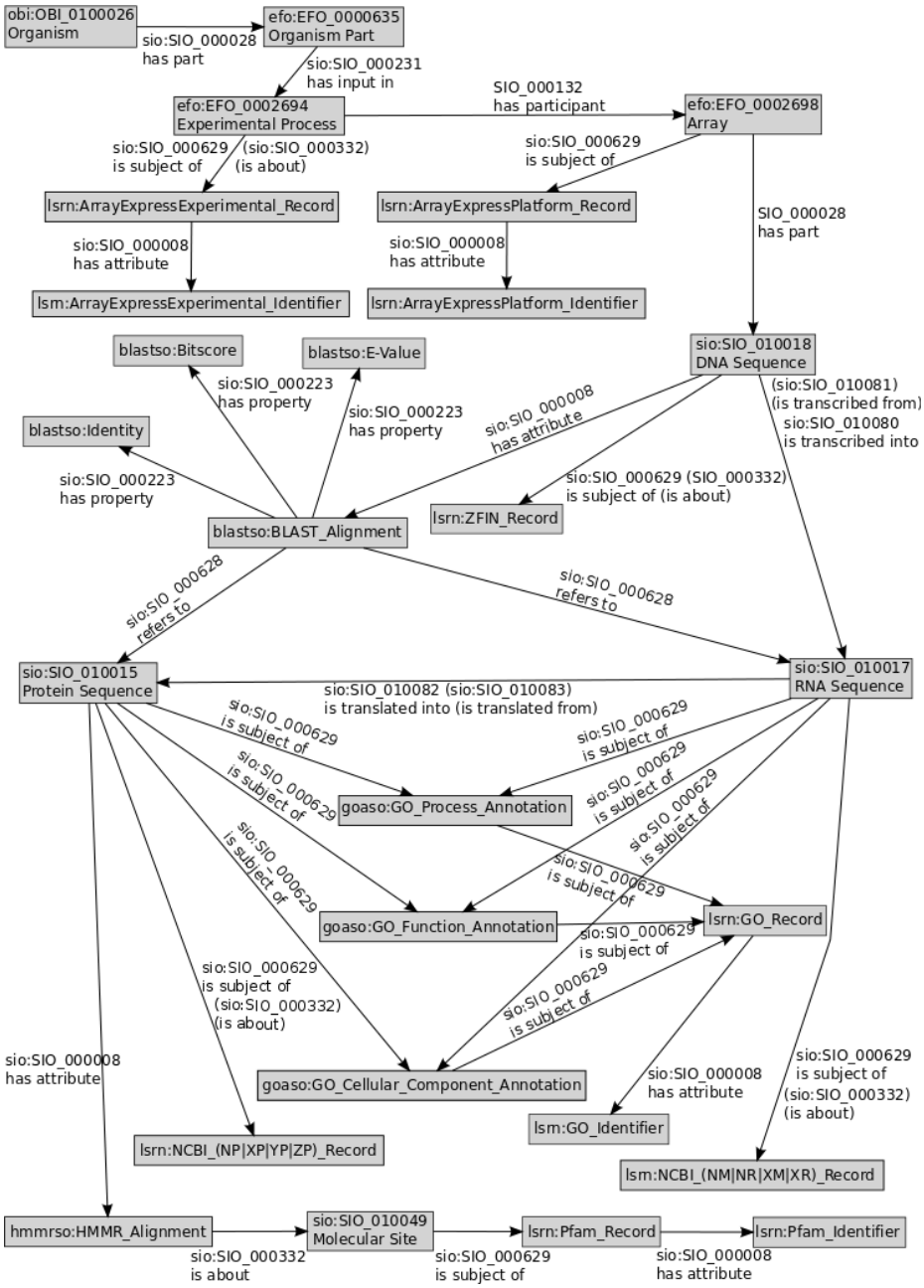


Fig 2. A schematic of the main OWL classes and properties used to expose data in the described SADI Web Services. A key to the prefixes used in this figure can be found in Table 1. The properties which are given in brackets are the inverse of the properties for the given direction, indicated by the connecting arrow.

2.3 SADI Web Services for fish research and aquatic ecotoxicology

In total we created 32 SADI Web Services which exposed information from five database: ArrayExpress, ZFIN, MGI, RefSeq, Pfam, and GO. We also exposed BLASTn, BLASTx, BLASTp, HMMR3 and ORF-Predictor tools. These services are too numerous to describe all but a selection in detail here, however a description of each is provided at http://unbsj.biordf.net/FISHTOX-SADIServices. A SHARE client has been made available to query these service at http://unbsj.biordf.net/cardioSHARE-fishtox.

Where possible we wrapped existing Web Services, provided by databases and tools, as SADI Web Services. This provides live data, which ensures results are current, and avoids the maintenance cost associated with data mirrors.

Four ArrayExpress SADI Web Services, one of which has been described already (Fig. 1), were created by wrapping the Web-Services provided by ArrayExpress. Information exposed was modeled using a combination of the existing EFO ontology (which the database supports natively) and SIO properties.

HMMR3 SADI Web Services were provided by wrapping the Web Services provided by janelia [10]. The input class of the service is a 'protein sequence' (sio:SIO_010015) and output class is defined in the hmmrso (Table 1) ontology as a class that:

```
'has attribute' min 1 (HMMR_Alignment that ('is about' min 1
('molecular site' that ('is subject of' min 1 Pfam_Record))))
```

Similarly SADI Web Services for BLAST were created by wrapping NCBI Web Services. The input to these services was either a 'protein sequence' (sio:SIO_010015) or a 'nucleic acid sequence' (sio:SIO_010016) depending on the variant of BLAST. The output is defined using an alignment class, in an approach similar to HMMR3 services. For example, the output for the BLASTx is defined in the blastso (Table 1) ontology as a class that:

```
'has attribute' some (BLAST_Alignment that ('refers to' min 1 ('protein
sequence' that ('is subject of' min 1 (NCBI_NP_Record or NCBI_AP_Record
or NCBI_XP_Record or NCBI_YP_Record or NCBI_ZP_Record)))))
```

Translation between any two database records is handled by modeling the relation between the sequences which they concern. For example, an lsrn:ZFIN_Record concerns some genomic sequence corresponding to a gene model. The SADI translation service consuming instances of this class as input, defines the relationship between the input and an NCBI protein record in RefSeq via the following output class tsso:RefSeq_Protein_Annotated_Record_Output:

```
'is about' min 1 ('deoxyribonucleic acid sequence' that ('is
transcribed into' min 1 ('ribonucleic acid sequence' that ('is
translated into' min 1 ('protein sequence' that ('is subject of' min 1
(NCBI_NP_Record or NCBI_AP_Record or NCBI_XP_Record or NCBI_YP_Record
or NCBI_ZP_Record)))))))
```

GO annotation SADI Web Services were created by directly RDFizing ZFIN and MGI annotations published on the GO website [29]. They annotate both

lsrn:ZFIN_Record and lsrn:MGI_Record classes. The definition of the output class is complex as the GO annotation can reference the function, process, or cellular compartment of the RNA or Protein product of the DNA which is the subject of ZFIN or MGI records.

3 Results

In this section we present three example queries, which address the types of questions pertinent to the analysis of gene expression data. However, the SADI Web Services we have built, and the modelling we employ, is not limited to these examples. Any number of combinations of these SADI Web Services, together with the growing number of public SADI Web Services, can be used to produce many useful queries. In this paper we focus on a few example queries based around the analysis of fish toxicology data, however these methodologies are widely applicable to gene expression analysis.

These queries are enacted by the SHARE client, which computes queries by picking and calling suitable SADI Web Services from a dedicated registry of fish toxicology-related services. The SHARE client Web interface reports results in tabular form and as a downloadable RDF graph.

3.1 Query I: Leveraging ORF finding algorithms to detect Pfam domains

After the gene sequences of interest have been identified, a common requirement is to classify these genes according to the protein domains, which they encode. This is often a non trivial task for microarray sequences, which are frequently derived from assembled EST sequences. Consequently sequences may be incomplete and contain missing gene fragments, which introduce shifts in the reading frame across the sequence. This makes the process of finding the correct open reading frame (ORF) which encodes the protein challenging. Combining the output of a ORF prediction tool, together with the HMMR3 algorithm, without scripting, would require a great deal of manual work for a biologist, which becomes insurmountable for anything but a trivial number of sequences. The following SPARQL query annotates Pfam domains for the ten most significantly regulated genes in *Micropterus salmoides*, relative to the control, under dieldrin-induced stress [18]. In order to compute the query SHARE calls three services. The first service decorates the DNA sequences on the chip with RNA. The RNA is then passed through the ORF prediction service to decorate a protein sequence, which is then passed to the HMMR3 service, which adds protein domains to the RDF model.

```
1. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3. PREFIX sio: <http://semanticscience.org/resource/>
4. SELECT ?DNA_chip_sequence ?pfam_name
5. FROM <http://unbsj.biordf.net/fishtox/TopTenLowestPvalue-DdResponsiveGenes.rdf>
6. WHERE {
```

```
7.  ?DNA_chip_sequence sio:SIO_010080 ?RNA_sequence .
8.  # (is transcribed into)
9.  ?RNA_sequence sio:SIO_010082 ?protein_sequence .
10. # (is translated into)
11. ?protein_sequence sio:SIO_000008 ?alignment .
12. # (has attribute)
13. ?alignment sio:SIO_000332 ?molecular_site .
14. # (is about)
15. ?molecular_site sio:SIO_000629 ?pfam_record .
16. # (is subject of)
17. ?pfam_record rdfs:label ?pfam_name
18. }
```

The first predicate in the query (line 7) causes SHARE to look for services indexed by the "is transcribed into" predicate. It finds a DNA2RNA SADI service which consumes a DNA sequence class and decorates this with an RNA sequence, which is attached by the "is transcribed into" property. The second predicate (line 9) is resolved by an ORF predictor service, which consumes RNA sequences, and uses sequence alignment to RefSeq proteins (BLASTx) to predict the most likely open reading frames that code for proteins. SHARE feeds the RNA sequences outputted by the DNA2RNA service into the ORF predictor SADI service, which decorates them with protein sequences attached by a "is translated into" property. The third predicate (line 11) can be resolved by the HMMR3 service, which consumes a protein sequence and produces HMMR alignments with attached Pfam protein domains. The forth and fifth predicates (line 15 and 17) are part of the output modeling of this HMMR3 service.

The SHARE client returned the answer that the domain Ribosomal_L7Ae was found on the gene UF_Msa_AF_100231. The low coverage on genes (10%) is not surprising given the species (*Largemouth Bass*), and the conservative default settings of the HMMR3 SADI service (e-value < Gathering threshold). The service is parameterized to allow these settings to be changed, but this functionality is not yet supported in SHARE. The RDF output from this query can be found at http://unbsj.biordf.net/fishtox/QueryIOutput.rdf.

3.2 Query II: Functional annotation of sequence data

One of the most powerful tools for microarray data is GO functional annotation. However, for a non-model organism like *Micropterus salmoides*, very little experimental evidence is recorded in public repositories for GO function. It is therefore necessary to infer function based on sequence similarity with known genes in model organisms. The following SPARQL query annotates the ten genes previously described in Section 3.2. To execute the query, SHARE calls the BLASTx service to find similar proteins in the RefSeq database, looks up the equivalent ZFIN and MGI Records, and then finally retrieves experimentally evidenced GO terms for these records using the corresponding SADI services from our set. The default e-value threshold for parameterized BLAST services is $1x10^{-4}$.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sio: <http://semanticscience.org/resource/>
SELECT ?DNA_chip_sequence ?zfin_or_msi_record ?go_id
FROM <http://unbsj.biordf.net/fishtox/TopTen-lowest-pvalue-DdResponsiveGenes.rdf>
```

```
WHERE {
  ?DNA_chip_sequence sio:SIO_000008 ?alignment .
  # (has attribute)
  ?alignment sio:SIO_000628 ?sequence_hit .
  # (refers to)
  ?sequence_hit sio:SIO_000629 ?refseq_record .
  # (is subject of)
  ?refseq_record sio:SIO_000332 ?RNA_Sequence .
  # (is_about)
  ?RNA_Sequence sio:SIO_010081 ?DNA_sequence .
  # (is_transcribed_from)
  ?DNA_sequence sio:SIO_000629 ?zfin_or_msi_record .
  # (is subject of)
  ?zfin_or_msi_record sio:SIO_000332 ?DNA_sequence .
  # (is about)
  ?DNA_sequence sio:SIO_010080 ?RNA_Sequence.
  # (is transcribed into)
  ?RNA_sequence sio:SIO_010082 ?protein_sequence .
  # (is translated into)
  ?protein_sequence sio:SIO_000629 ?GO_annotation .
  # (is subject of)
  ?GO_annotation  sio:SIO_000629 [rdfs:label ?go_id]
}
```

The results from SHARE indicate the presence of ribosomal processes and functions, that were experimentally evidenced in genes with sequences similar to the ten sequences being annotated. This accords well with the Pfam domains found by Query II. The query results also identified a number of additional gene functions, which include transcription factor, enzyme binding, steroid hormone receptor, cholesterol transporter, and phospholipid binding activities. The RDF output from this query can be found at http://unbsj.biordf.net/fishtox/QueryIIOutput.rdf.

3.3 Query III: Locating relevant microarray experiments

A frequent requirement of experimentalists involved in transcriptomics is the comparison of their own work with previous published experiments. Locating microarray experiments with related experimental variables is a prerequisite for further comparative analysis. In order to answer this question an experimentalist typically would use the Web-tools provided by ArrayExpress. One such example query might be “For the hypothalamus of *Micropterus salmoides*, what gene transcripts have been measured in existing experiments”. Using ArrayExpress Web based tools alone would require multiple searches and manual inspections of many experiments each of which may use different microarray platforms. The following declarative SPARQL query expresses this question formally. Note that the RDF file http://unbsj.biordf.net/fishtox/large-mouth-bass-27706.owl specified in the FROM clause contains the OWL class of the organism of interest (*Micropterus salmoides*) to instantiate the first query line. This was created from a subset of http://purl.org/obo/owl/NCBITaxon using OntoFox [28]

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX aeso:
        <http://unbsj.biordf.net/fishtox/arrayexpress-sadi-service-ontology.owl#>
SELECT ?experiment_accession ?tissue_name ?platform_accession ?gene_id
FROM <http://unbsj.biordf.net/fishtox/large-mouth-bass-27706.owl>
```

```
WHERE {
 ?organism_class aeso:has_instance ?organism_instance .
 ?organism_instance a ncbitaxon:NCBITaxon_27706 .
 ?organism_instance sio:SIO_000028 ?organism_part .
  # (has part)
  ?organism_part sio:SIO_000231 ?experimental_process .
  # (is input in)
 ?organism_part rdfs:label ?tissue_name .
  # (has value)
  ?experimental_process sio:SIO_000629 ?experimental_record .
  # (is subject_of)
  ?experimental_record sio:SIO_000008 [sio:SIO_000300 ?experiment_accession] .
  # (has attribute, has value)
  ?experimental_record sio:SIO_000332 ?experimental_process .
  # (is about)
  ?experimental_process sio:SIO_000132 ?array .
  #(has participant)
  ?array sio:SIO_000629 ?array_platform_record .
  # (is subject of)
  ?array_platform_record sio:SIO_000008 [sio:SIO_000300 ?platform_accession] .
  # (has attribute, has value)
  ?array_platform_record sio:SIO_000332 ?array .
  # (is about)
  ?array sio:SIO_000028 [rdfs:label ?gene_id] .
  # (has part)
  FILTER regex(?tissue_name, "hypothalamus", "i")
}
```

The query was submitted to the SHARE client which resolved the answer by leveraging 5 SADI Web Services which expose relevant ArrayExpress data. No understanding of the ArrayExpress semantic idiosyncrasies or data syntax was required to formulate the query. SHARE identifies two microarray experiments which meet the requirements of this query. The RDF output from this query can be found at http://unbsj.biordf.net/fishtox/QueryIIIOutput.rdf.

4 Conclusions and further work

The aim of this fish toxicology use-case was to demonstrate how a moderate number of SADI Web Services can enable diverse and powerful queries using the SHARE client. The services described exposed information from five databases and three analytical tools in a semantically rich and explicit way. They provided a single access-point for an ecotoxicologist to query data, and a unified and semantically consistent data representation. When using this framework, a ecotoxicologist would not require understanding of the semantics and technicalities of the underlying resources, in order to construct queries across databases and tools. We acknowledge that designing SPARQL queries may be beyond the reach of many biologists. However, the graphical workflow and query tools, Taverna [27] and Sentient Knowledge Explorer [32], have active SADI plug-ins under development, which may provide a solution to this interface deficiency.

In future work we will expand the SADI Web Services provided in this use-case to leverage experimental observations of gene expression. We will also provide services for common statistical methods, such as gene set enrichment analysis. In our specific

example with largemouth bass, this will enable queries such as “Which GO functions are significantly enriched in teleost fish in response to dieldrin treatment”. We also intend to develop queries which leverage some of the many public SADI Web Services, developed outside this project.

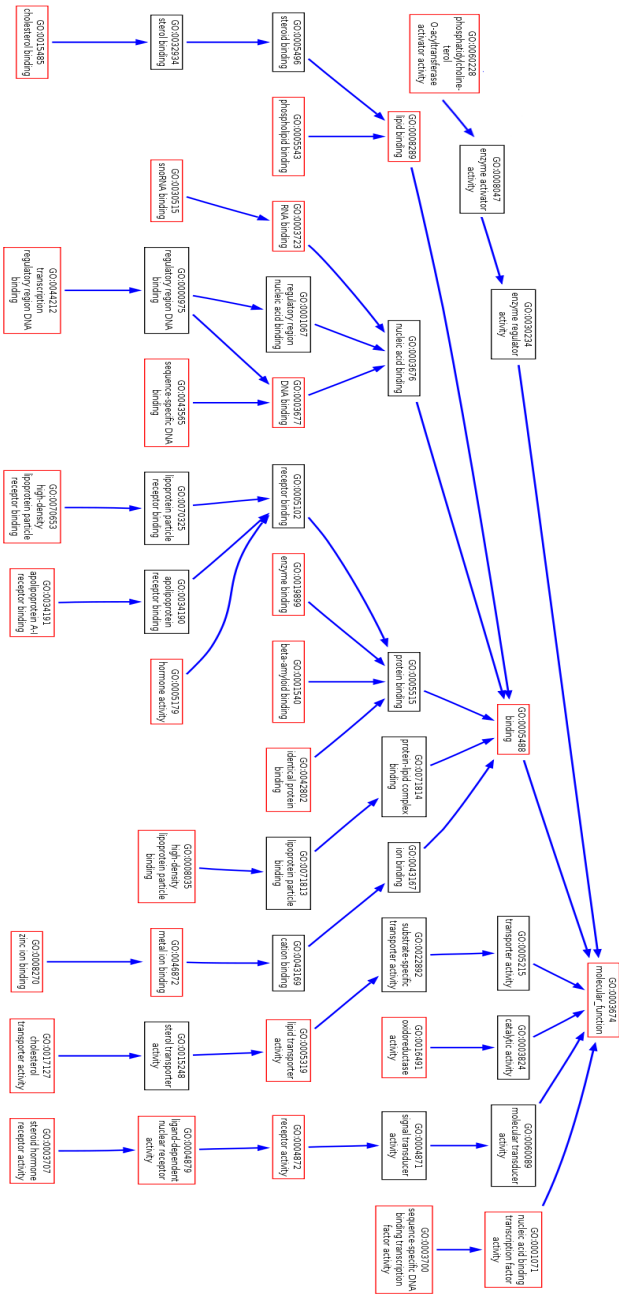
Acknowledgements

This research was funded by CANARIE NEP-2 Program (C-BRASS project). We thank Luke McCarthy and Ben Vandervalk for helping us with various SADI-related technical issues.

References

1. Adamusiak, T., Malone, J.: Semantic Web Atlas Project, http://www.ebi.ac.uk/efo/semanticweb/atlas.
2. Van Aggelen, G. et al.: Integrating Omic Technologies into Aquatic Ecological Risk Assessment and Environmental Monitoring: Hurdles, Achievements, and Future Outlook. Environ Health Perspect. (2009).
3. Baker, C.J.O.: Semantic Infrastructure for Automated Small Molecule Classification and Data Mining for Lipidomics. CSHALS. , Boston, MA (2011).
4. Belleau, F. et al.: Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. Journal of Biomedical Informatics. 41, 5, 706-716 (2008).
5. Blake, J.A. et al.: The Mouse Genome Database (MGD): premier model organism resource for mammalian genomics and genetics. Nucleic Acids Research. 39, Database, D842-D848 (2010).
6. Bradford, Y. et al.: ZFIN: enhancements and updates to the zebrafish model organism database. Nucleic Acids Research. (2010).
7. Chepelev, L.L., Dumontier, M.: Semantic Web integration of Cheminformatics resources with the SADI framework. Journal of Cheminformatics. 3, 16 (2011).
8. Eddy, S.R.: Accelerated profile HMM searches, ftp://selab.janelia.org/pub/publications/Eddy11/Eddy11-preprint.pdf.
9. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema, http://www.w3.org/TR/sawsdl/.
10. Finn, R.D. et al.: HMMER web server: interactive sequence similarity searching. Nucleic Acids Research. (2011).
11. Finn, R.D. et al.: The Pfam protein families database. Nucleic Acids Research. 38, Database, D211-D222 (2009).
12. Gessler, D. et al.: SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services. BMC Bioinformatics. 10, 1, 309 (2009).
13. Kapushesky, M. et al.: Gene Expression Atlas at the European Bioinformatics Institute. Nucleic Acids Res. 38, Database issue, D690-D698 (2010).

14. Löhr, H., Hammerschmidt, M.: Zebrafish in endocrine systems: recent advances and implications for human disease. Annu. Rev. Physiol. 73, 183-211 (2011).
15. Malone, J. et al.: Modeling Sample Variables with an Experimental Factor Ontology. Bioinformatics. (2010).
16. Martin, D. et al.: Bringing semantics to web services: The OWL-S approach. Semantic Web Services and Web Process Composition. 26–42 (2005).
17. Martyniuk, C.J. et al.: Omics in aquatic toxicology: Not just another microarray. Environmental Toxicology and Chemistry. 30, 2, 263-264 (2011).
18. Martyniuk, C.J. et al.: Genomic and Proteomic Responses to Environmentally Relevant Exposures to Dieldrin: Indicators of Neurodegeneration? Toxicological Sciences. 117, 1, 190 (2010).
19. Min, X.J. et al.: OrfPredictor: predicting protein-coding regions in EST-derived sequences. Nucleic acids research. 33, suppl 2, W677 (2005).
20. Parkinson, H. et al.: ArrayExpress update—an archive of microarray and high-throughput sequencing-based functional genomics experiments. Nucleic Acids Res. 39, Database issue, D1002-D1004 (2011).
21. Riazanov, A. et al.: Deploying the Mutation Impact mining pipeline with SADI: an exploratory case study.
22. Sukardi, H. et al.: Zebrafish for drug toxicity screening: bridging the in vitro cell-based models and in vivo mammalian models. Expert Opin Drug Metab Toxicol. 7, 5, 579-589 (2011).
23. Vandervalk, B. et al.: SHARE: A Semantic Web Query Engine for Bioinformatics. The Semantic Web. 367–369 (2009).
24. Vascotto, S.G. et al.: The zebrafish's swim to fame as an experimental model in biology. Biochem. Cell Biol. 75, 5, 479-485 (1997).
25. Villeneuve, D.L., Garcia-Reyero, N.: Vision & strategy: Predictive ecotoxicology in the 21st century. Environmental Toxicology and Chemistry. 30, 1, 1-8 (2011).
26. Wilkinson, M.D. et al.: SADI Semantic Web Services-‚cause you can't always GET what you want! Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific. pp. 13–18 (2009).
27. Withers, D. et al.: Semantically-guided workflow construction in Taverna: the SADI and BioMoby plug-ins. Leveraging Applications of Formal Methods, Verification, and Validation. 301–312 (2010).
28. Xiang, Z. et al.: OntoFox: web-based support for ontology reuse. BMC Research Notes. 3, 1, 175 (2010).
29. Current Annotations, http://www.geneontology.org/GO.downloads.annotations.shtml.
30. Linked Life Data, http://linkedlifedata.com/.
31. OWL 2 Web Ontology Language Document Overview, http://www.w3.org/TR/owl2-overview/.
32. Sentient Knowledge Explorer, http://www.io-informatics.com/products/sentient-KE.html.



Appendix 1. GO Molecular functions identified by SHARE for example query III. Red bordered concepts indicate annotations found for similar sequences.

Towards Evaluating the Impact of Semantic Support for Curating the Fungus Scientific Literature

Marie-Jean Meurs[1], Caitlin Murphy[2,3], Nona Naderi[1], Ingo Morgenstern[2,3], Carolina Cantu[2,3], Shary Semarjit[2,4], Greg Butler[1,2], Justin Powlowski[2,4], Adrian Tsang[2,3] and René Witte[1*]

[1] Department of Computer Science and Software Engineering
[2] Centre for Structural and Functional Genomics
[3] Department of Biology
[4] Department of Chemistry and Biochemistry
Concordia University, Montréal, QC, Canada
mjmeurs@encs.concordia.ca, cmurphy@gene.concordia.ca, n_nad@encs.concordia.ca, {imorgenstern,ccantut,sshary}@gene.concordia.ca, gregb@encs.concordia.ca, powlow@alcor.concordia.ca, tsang@gene.concordia.ca, rwitte@cse.concordia.ca

Abstract. We present our ongoing development of a semantic infrastructure supporting biofuel research. Part of this effort is the automatic curation of knowledge from the massive amount of information on fungal enzymes that is available in genomics. Working closely with biologists who manually curate the existing literature, we developed ontological NLP pipelines, integrated through Web-based interfaces, to help them in two main tasks: spending less time to mine the literature for facts, while also being provided with richer and semantically linked information. An ongoing challenge is to measure precisely how much the developed semantic technologies benefit the end users and what their overall impact on the quality of the curated data is. We present preliminary evaluation results that show a significant reduction in manual curation time.

1 Introduction

Producing sustainable liquid fuels with low environmental impact is one of the major technological challenges the world is facing today. Industrialized and developing countries consider *biofuels*, fuels produced from biomass, as a promising alternative to fossil based fuels. Extracting sugars from cellulose to produce biofuels requires to break down cellulose by using specific molecules called enzymes. Therefore, in the current race for replacing petroleum based fuels with renewable biofuels, discovering the most efficient enzymes for the cellulose degradation is a key challenge.

The largest knowledge source available to biofuel researchers is the PubMed bibliographic database, containing more than 19 million citations from over 21,000 life science journals. PubMed is linked to other databases, like *Entrez Genome*, which provides access to genomic sequences or *BRENDA, The Comprehensive Enzyme Information System* [9], which is the main collection of enzyme functional

*corresponding author

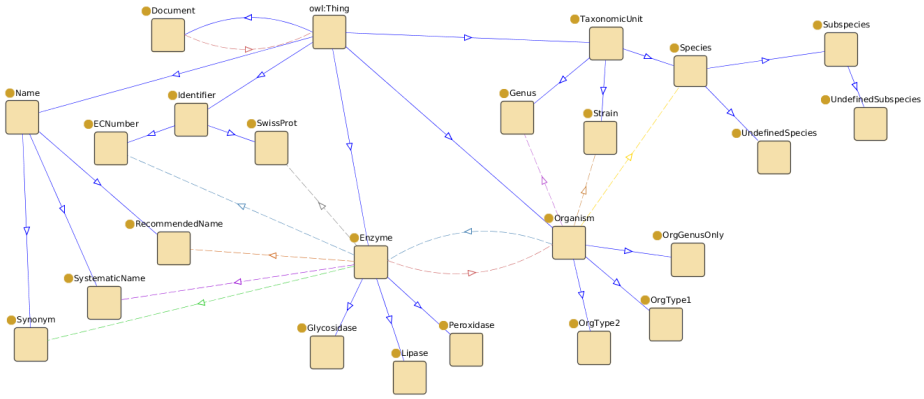


Fig. 1. Domain Ontology: Organism and Enzyme Entities

data available to the scientific community. A biology researcher querying PubMed using keywords collects an often long list of relevant papers. The way to analyze this collection is reading all the abstracts and sometimes the full text papers: this task is time consuming, difficult to handle and significant knowledge can be easily missed.

To address this problem, Natural Language Processing (NLP) and Semantic Web approaches are increasingly adopted in biomedical research [2, 10]. The work-in-progress we present in this paper focuses on the automatic extraction of knowledge from the massive amount of information on enzymes in fungi available from genome research. Text mining systems, like the one we developed here, are typically evaluated with *intrinsic* metrics, such as precision and recall. However, while these metrics can give insight into the accuracy of a system, they do not necessarily correspond to their *extrinsic* performance [1, 4]: How much does the system actually improve the tasks performed by users? Thus, in this work we are interested in also evaluating the impact of our semantic systems on the work performed by our biologists and the quality of the curated data.

2 Project Context and System Architecture

Before we describe our overall architecture and the text mining pipelines, we briefly introduce the user groups involved and the semantic entities we analyse.

User Groups. The identification and the development of effective fungal enzyme cocktails are key elements of the biorefinery industry. In this context, the manual curation of fungal genes provides the thorough knowledge required for guiding research and experiments. The biology researchers involved in this curation are filling the mycoCLAP database [8], which is a searchable database of fungal genes encoding lignocellulose-active proteins that have been biochemically characterized. The *curators* are therefore the first user group of our system. The *biology researchers* who make decision about the experiments to conduct and the *experimenters* executing them represent two further user groups. They are mainly interested in the ability of combining multiple semantic queries to the curated data, thereby integrating the various knowledge resources.

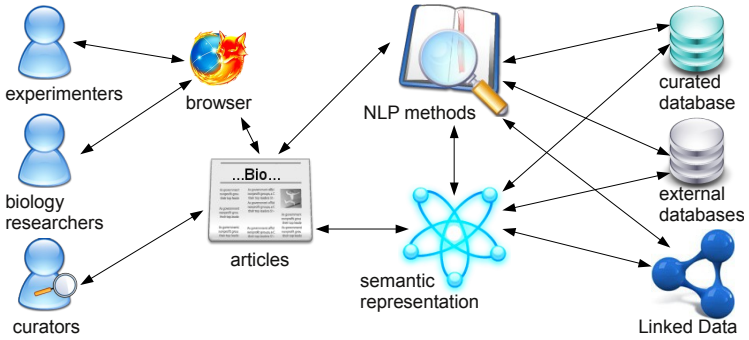


Fig. 2. Integrating Semantic Support in Curation, Analysis, and Retrieval

Semantic Entities. The system we are developing has to support the manual curation process; therefore, the semantic annotation types have been defined by the curators according to the information they need to store in the mycoCLAP database. Entities include information such as organisms, enzymes, assays, genes, kinetic properties, reactions, substrates, and environmental conditions. To facilitate semantic discovery, linking and querying these concepts across literature and databases, these entities are modeled in OWL ontologies, which are automatically populated from documents. As an example, Fig. 1 shows two main entities encoded in our ontology, *organisms* [13] and *enzymes*. The ontology is used both during the text mining process and for querying the extracted information.

Semantic Resources. In terms of knowledge sources, the system relies on external and internal processing resources and ontologies. The *Taxonomy database* [6] from NCBI is used for initializing the NLP resources supporting the organism recognition. BRENDA [9] provides the enzyme knowledge along with the UniProtKB/SwissProt [11]. References to the original sources are integrated into the curated data. This facilitates semantic connections through standard Linked Data techniques, e.g., from an organism mention in a research paper to its corresponding entry in the NCBI Taxonomy database.

System Architecture. With the large number of different user groups and their diverging requirements, as well as the existing and continuously updated project infrastructure, we needed to find solutions for incrementally adding semantic support without disrupting day-to-day work. Our solution deploys a loosely-coupled, service-oriented architecture that provides semantic services through existing and new clients. To connect these individual services and their results, we rely on standard semantic data formats, like OWL and RDF, which provide both loose coupling and semantic integration, as new data can be browsed and queried as soon as it is added to the framework (Fig. 2).

NLP services are provided by the Semantic Assistants architecture [12], which facilitates the publication of NLP pipelines through standard Web services with WSDL descriptions. Users can access these Semantic Assistants services from their desktop through client plug-ins for common tools, such as the Firefox Web browser or the OpenOffice word processor.

3 Text Mining Pipelines

Our text mining pipelines are based on the *General Architecture for Text Engineering* (GATE) [5]. All documents first undergo basic preprocessing steps using off-the-shelf components, such as tokenization, sentence splitting, and part-of-speech tagging. Custom pipelines then extract the semantic entities mentioned above and populate the OWL ontologies using the OwlExporter component. The same pipeline can be run for automatic (batch) ontology population, embedded in Teamware (described below) for manual annotation, or brokered to desktop clients through Web services for literature mining and curation.

Organism Recognition. The organism tagging and extraction relies on external resources that are automatically translated for reuse in our system, thereby providing users with the ability to update their installation when the NCBI Taxonomy database changes. Additionally, a custom built organism ontology, presented in Fig. 1, formally describes the linguistic structure of organism entities at different levels of the taxonomic hierarchy [13]. The GATE pipeline consists of modules for organism entity detection based on pattern matching to the NCBI reference taxonomy, providing scientific names and the NCBI Taxonomy Identifier. Strain mentions are extracted using a specific text tokenization and a machine learning based approach.

Enzyme Recognition. Despite the standards published by the Enzyme Commission [7], enzymes are often described by the authors under various formats. An enzyme-specific text tokenization, along with grammar rules written in the JAPE language, analyses tokens with the *-ase* enzyme suffix. Then, the enzyme entity recognition relies on automatically extracted knowledge from the BRENDA database. A pattern matching approach provides enzyme name identification. The detected enzyme mentions are associated with their *EC number*, their *Recommended Name*, their *Systematic Name* and their URL on the BRENDA website.

Temperature and pH Facts. Temperature and pH mentions are involved in several biological facts, like the temperature and pH dependence/stability or the description of the activity and kinetic assay conditions. Our GATE pipeline contains PRs based on JAPE rules and gazetteer lists of specific vocabulary that enable the detection of these key mentions at the sentence level.

4 Intrinsic and Extrinsic Evaluation

As explained above, text mining systems require an evaluation showing their efficiency and effectiveness, both intrinsically and from an end user's point of view. In this section, we first discuss the development of the gold standard corpus and present preliminary evaluation results of our system.

4.1 The Manual Annotation Process

For the intrinsic evaluation, we are building a gold standard corpus of freely accessible full-text articles by manually annotating them using GATE Teamware [3], a Web-based management platform for collaborative annotation and curation. The annotation team is composed of four biology researchers. The researcher in charge

of the curation task and an annotator having a strong background in fungus literature curation are considered as expert annotators. Their inter-annotator agreement is over 80%, hence their annotation sets are always defined as the most reliable sets during the adjudication process. The corpus is composed of ten papers related to a class of enzymes. Glycoside hydrolase papers and lipase papers each represent 40% of the articles, whereas 20% are related to peroxidases.

4.2 Intrinsic Evaluation: Precision and Recall

The correctness of our text mining pipelines is evaluated in terms of precision, recall and F-measure. The reference is provided by the manually annotated (gold standard) corpus. The preliminary results on the four most common entities (Enzyme, Organism, pH and Temperature) are shown in Table 1.

Table 1. Text Mining Pipelines: Precision, Recall and F-measure

	Strict (overlaps discarded)			Lenient (overlaps included)		
	Recall	**Precision**	**F-m**	**Recall**	**Precision**	**F-m**
Enzyme	0.64	0.55	0.59	0.78	0.67	0.72
Organism	0.84	0.81	0.82	0.88	0.83	0.85
pH	0.74	0.76	0.75	0.95	0.99	0.97
Temperature	0.64	0.67	0.65	0.90	0.93	0.91

4.3 Extrinsic Evaluation: Literature Mining and Annotation

The impact of the system on the *curation* and *annotation* tasks is evaluated in terms of required time (range and average) per paper and measured in minutes.

Paper selection. Since the beginning of the curation task, approximately 1000 papers have been examined. The time needed to examine an unannotated full paper and to make a decision about its selection for curation, without any semantic support, previously ranged from 2 to 3 minutes. With added support through the text mining services, the required time decreased to 1–2 minutes.

Paper curation. Among the 1000 examined papers, around 600 were already selected for curation. The time needed to curate an unannotated full paper, i.e., extracting salient facts for entry into the mycoCLAP database, ranged from 30 to 45 minutes for the fully manual workflow. With added semantic support through the text mining pipelines, the required time decreased to 20–30 minutes.

Paper annotation. For full paper annotation, we investigated the impact of different levels of semantic support on the time required to add annotations (Table 2). All sets have been manually annotated by four annotators. The 4 papers of the first set (SET 1) were annotated without any semantic support. The second set (SET 2) is composed of 3 papers, which have been pre-annotated by a degraded version of the system, using only generic tools, such as simple gazetteering list, resulting in lower precision and recall. The third set (SET 3) contains 3 papers, pre-annotated using the complete text mining pipelines, including the specialized tools and external resources as described above.

From the preliminary results, we can conclude that (1) there is a significant reduction of the average time required for paper selection, curation and annotation and (2) the level of support has a measurable impact as well.

Table 2. Average annotation time per paper with different levels of semantic support

set and level of semantic support		available tags	$\bar{t}$ (min)
SET 1	(no semantic support)	$\emptyset$	90
SET 2	(partial semantic support)	enzyme, organism, pH, temperature	65
SET 3	(full semantic support)	enzyme, organism, pH, temperature	56

5 Conclusions

We presented our ongoing development of a semantic infrastructure for enzyme data management. In the context of biofuel research, our system targets the automatic extraction of knowledge on fungal enzymes from genome research literature. Preliminary experiments show that semantic support allows for a significant decrease in manual curation time. However, future work is needed to evaluate the impact of such a system on the quality of the curated data.

Acknowledgments. Funding for this work was provided by Genome Canada and Génome Québec.

References

1. Alex, B., Grover, C., Haddow, B., Kabadjov, M., Klein, E., Wang, X.: Assisted curation: does text mining really help. In: Pacific Symposium on Biocomputing. vol. 13, pp. 556–567 (2008)
2. Ananiadou, S., McNaught, J.: Text Mining for Biology And Biomedicine. Artech House, Inc., Norwood, MA, USA (2005)
3. Bontcheva, K., Cunningham, H., Roberts, I., Tablan, V.: Web-based Collaborative Corpus Annotation: Requirements and a Framework Implementation. In: New Challenges for NLP Frameworks. pp. 20–27. ELRA, Valletta, Malta (May 22 2010)
4. Caporaso, J.G., Deshpande, N., Fink, J.L., Bourne, P.E., Cohen, K.B., Hunter, L.: Intrinsic evaluation of text mining tools may not predict performance on realistic tasks. In: Pacific Symposium on Biocomputing. vol. 13, pp. 640–651. World Scientific Publishing (2008)
5. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proc. 40th Anniversary Meeting of the ACL (2002)
6. Federhen, S.: The Taxonomy Project. In: McEntyre, J., Ostell, J. (eds.) The NCBI Handbook, chap. 4. National Library of Medicine (US), National Center for Biotechnology Information (2003)
7. International Union of Biochemistry and Molecular Biology: Enzyme Nomenclature 1992. Academic Press, San Diego, California (1992)
8. Murphy, C., Powlowski, J., Wu, M., Butler, G., Tsang, A.: Curation of characterized glycoside hydrolases of fungal origin. Database 2011 (2011)
9. Scheer, M., Grote, A., Chang, A., Schomburg, I., Munaretto, C., Rother, M., Söhngen, C., Stelzer, M., Thiele, J., Schomburg, D.: BRENDA, the enzyme information system in 2011. Nucleic Acids Res. 39, (Database issue):D670–676 (2011)
10. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE Intelligent Systems 21(3), 96–101 (2006)
11. The UniProt Consortium: The Universal Protein Resource (UniProt). Nucleic Acids Research 37(D), 169–174 (2009)
12. Witte, R., Gitzinger, T.: Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In: 3rd Asian Semantic Web Conference (ASWC 2008). LNCS, vol. 5367, pp. 360–374. Springer, Bangkok, Thailand (2009)
13. Witte, R., Kappler, T., Baker, C.J.O.: Ontology Design for Biomedical Text Mining. In: Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences, chap. 13, pp. 281–313. Springer (2007)

Ontology-Based Text Mining of Concept Definitions in Biomedical Literature

Saeed Hassanpour, Amar K. Das

Stanford Center for Biomedical Informatics Research,
Stanford, CA 94305, U.S.A.
{saeedhp, amar.das}@stanford.edu

Abstract. Many developers of biomedical knowledge bases typically validate and update formalized knowledge based on reviews of full-text scientific articles, but finding text relevant to domain concepts can be tedious and prone to errors. Prior methods have automated this process by matching term-based patterns within a single sentence. In our work developing a knowledge base of autism phenotypes, specified using Semantic Web standards, we are interested in finding multi-sentence sections of text that contains complex phenotype definitions. In this paper, we present a text-mining method that incorporates both ontology- and rule-based semantics to determine which section is relevant. We evaluated our method in undertaking text extraction for the set of full-text articles used to create the knowledge base. We show that our method has higher precision and recall than a term-based approach in identifying definitions that contain complex patterns and occur across sentence boundaries.

Keywords: Information Extraction, Text Analysis, Semantic Web, Ontology, Rules Base, OWL, SWRL

1 Introduction

Biomedical knowledge resources, such as terminologies and ontologies, are important for community-based annotation and sharing of data. Creating and maintaining these resources is challenging given the rapid growth of scientific knowledge. Generally, scientists, annotators and developers try to keep up by using search engines that find publications relevant to given concepts in the knowledge resource. However, users still need to review the publications and find sections within the documents that relate to the concept being searched. One solution to this challenge is to automatically identify the relevant parts of a full-text document. Prior methods, such as Textpresso [1], have focused on finding individual sentences that match the terms of biomedical concepts and of properties that connect concepts. Such approaches do not find sections of an article—including multiple sentences—that are semantically and implicitly relevant to the definition of a concept. In our work, we present a novel text mining method that retrieves the most semantically informative text in a document using definitions of concepts modeled as rules in a domain ontology, and we compare the precision and recall of our method against a term-based approach.

Our work is motivated by the needs of developers of an ontology of autism phenotypes [2, 3]. As part of these efforts, experts want to easily find text within a publication that relates to the definition of a phenotype concept, both to find new definitions of that concept and to annotate the document section as the relevant text to the concept. For example, in a paper on autism genetics, Hus et al. [4] define Savant-positive and Savant-negative phenotype concepts as:

> *The Savant Skills Factor was based on … current and ever scores of four ADI-R items: visuospatial ability, memory skill, musical ability, and computational ability. Item scores were summed and divided by total number of items to generate a score between 0 and 1. … Participants were then divided into two groups: Savant-positive and Savant-negative … .*

The autism ontology uses the Web Ontology Language (OWL) [5] to model concepts and hierarchical relationships and the Semantic Web Rule Language (SWRL) [6] to define phenotype concepts as value restrictions on data collected through standardized instruments, such as the Autism Diagnostic Instrument-Revised (ADI-R).

2 Related Work

Finding text relevant to a search term is undertaken by some web search engines, which provides a few lines of site description or snippet for a search result to indicate the relevance of a web page to the search query. Google, for example, uses the description provided by meta tags, references to the web pages, Open Directory [7], and the text around the query keywords on web pages to provide informative search result descriptions [8]. We argue that structured domain knowledge can be used to enhance the relevance of snippets to the queries as well and provide the most semantically relevant parts of web page contents in result snippets.

Another related work in this field is question-answering systems, which return a part of a text from a corpus as the answer to a specified question. These techniques rank the snippets from the relevant documents by criteria such as: containing expected types of named entities, the percentage of overlap with question terms, containing lexical patterns, and using information from lexicon dictionaries [9-11]. Other work has tried to retrieve descriptive phrases from free text by using pattern matching, word counting, and sentence location without using domain knowledge [12]. In our work, we address the broader problem of extracting text that is semantically relevant to domain concepts. Our approach leverages the structured and axiomatic forms of knowledge in ontologies and rules, which contain richer semantic relationships than lexical databases.

3 METHODS

In our work, we find the most relevant parts of science publications to domain concepts using existing OWL ontologies and SWRL rules. As noted, both provide formal definitions of domain concepts and their relationships to other concepts.

3.1 Semantic Concept Modeling

As the first step, we need a formal representation of domain concepts. In this work, we use vector space modeling, a common method in the web search engines for indexing web pages [13], and a structured knowledgebase as a basis of the concept modeling. The concepts in the knowledgebase may be formally defined in logical form of SWRL rules and saved as a part of an OWL ontology, as in the case of the autism ontology. We thus consider rules' components as relevant concepts and incorporate them in our modeling for better presentation of the main concept. Therefore, we have one dimension for each ontology class and property mentioned in the rule as relevant concepts.

Besides the classes and properties that are mentioned in the rule, we use ontology hierarchies to extract more related concepts and incorporate them in the concept presentation. We consider the parents and grandparents of the main concept and its related concepts extracted from the corresponding rule as potential related concepts that can strengthen our concept vector modeling. However, the relevance of these concepts from the ontology hierarchy decreases by their distance from the main concept in the hierarchy graph. Therefore, we weight these related terms in the vector presentation less than the main class and the related concepts explicitly mentioned in the rule that defines the concepts. As a heuristic choice to capture these differences, we count the frequencies of the parent classes or properties as half of the actual frequencies, and the frequencies of grandparent classes or properties as one-quarter of the actual frequencies.

3.2 Relevant Text Finding

After we model the concept, we go through a publication to find the most relevant parts of the text for a particular concept. As the first step, we look at the vector representation of the concept and found all the terms associated with that concept as the concept terms. Concept terms are the terms that have weights greater than zero in the concept vector presentation. We then go through the publication and mark all the occurrence of the concept terms in the text. We cover occurrences of different forms of a concept terms by applying, Porter stemming algorithm, a common stemming method for English terms [14], on both concept terms and publication terms.

Given the occurrences of concept terms in a publication, we treat them as indicators of relevant parts of the text and use single linkage hierarchal clustering to find the candidates for the most relevant parts of the publication. The average sentence length in our corpus is 20 words. In the single linkage clustering we use 30 words as a heuristic threshold and in every step we merge the closest clusters that are separated by less than 30 words. Thus, we ensure that a continuous section of text without any concept term is limited to a few sentences and the whole cluster is continuously correlated to the concept. We consider these clusters as the candidates for the most relevant parts of the text.

3.3 Text Modeling and Correlation Computation

In this work, our goal is to quantify the relevance between concepts and pieces of text. Therefore, we need a mathematical modeling of texts. We use vector space modeling again to provide a common basis for comparison. Vector space modeling for documents' text is based on term frequencies. To model a part of a text as a vector, we first remove the stop words, the most common English words that are not informative about the context. We use a common list of stop words in English [15]. Then we apply Porter stemming algorithm to replace different derivations of a word with their root. Then we build a vector with one dimension for each term in the text and assign the frequency of that term in the text as the value of that dimension in the vector.

After we present both text words and domain concepts as vectors, we need to compute the correlations between them in order to find the most relevant parts of a publication for a concept. To do that, we use cosine similarity as the measure of correlation between texts and concepts. The cosine similarity for two vectors is the cosine of the angle between them. Similarity values range from 0 for orthogonal vectors to 1 for parallel vectors.

3.4 Evaluation Strategy

In this work, we applied our method on the autism phenotype ontology and the papers used to derive those concepts as mentioned in Section 1. We examined only the top five most relevant parts of the publication for each concept and had an autism ontology expert review these text sections to determine the efficacy and accuracy of whether each section was related or not to the definition of the concept. To investigate the significance of using ontological hierarchies and rule bases, we compared our method to a baseline, which is a term-only method. The baseline method is a variation of our method that only uses the terms in the semantic concept-modeling step. That is, our baseline approach does not include concepts from the ontology or rules that are related to the term. To eliminate bias in the assessment of the performance of the two approaches, the expert was blind to which method produced the extracted text.

4 Results

The autism ontology contains 1726 classes and properties, and it includes 156 SWRL rules that correspond to 145 phenotype definitions. The ontology and rules were based on a review of 26 publications that had been undertaken by one of the authors (AKD) and other domain experts in autism [3]. For this study, we selected 49 domain concepts that had rules using multiple criteria to define a phenotype (such as the example concept of Savant positive given in Section 1). We excluded phenotype definitions where the concept directly corresponded to the value of a single item on a clinical assessment. We applied both our ontology-based text extraction method and the term only method on each of the 49 concepts, and we returned the top 5 most

relevant parts of the publication for review by the domain expert. Altogether 338 sections of text were reviewed and evaluated by the autism ontology expert as to whether they were relevant to the corresponding phenotype concept. Table 1 shows the precision of our ontology-based method and the concept-based method—that is, the percentage of returned sections that refer to the concepts.

Table 1. The precision of the term- and ontology-based methods in finding texts relevant to phenotype definitions

Method	Precision (%)
Term based	68 %
Ontology based	76 %

In our evaluation strategy, we knew that every concept had been defined in the corresponding publication. For further investigation of the relevance strength in our results, we asked the reviewer to identify which of the five most relevant parts of the publications for a concept contained a clear definition. We used this to calculate the recall for each method, which is the percentage of concepts that their definitions were found. Table 2 shows the recall of the concept- and ontology-based methods in finding the definitions of the concepts in the corresponding publication text.

Table 2. The recall of the term- and ontology-based methods in identifying phenotype definitions in the publication text

Method	Recall (%)
Term based	39 %
Ontology based	69 %

5 Discussion

In this paper, we present a novel method to find parts of text in scientific publications that relate to definitions of biomedical concepts. In comparison to methods that do term matching to find individual sentences that contain a single concept or pairwise sets of concepts, our ontology-based approach addresses the challenge of finding a concept definition that occurs across multiple sentences or that is semantically similar to predefined concepts. Our approach was particularly driven by the need to identify text related to complex domain concepts like autism phenotypes, in which use different terms and terminologies refer to similar concepts. Our evaluation shows that ontology hierarchies and rules have a large impact on identifying the relevant parts of the text. This is because of the informative nature of ontological hierarchies and the inter-relationship of concepts maintained in rule bases.

As future work, we are planning to improve upon our method by using the text's syntactic structures through constituent and dependency parsing methods. The syntactic and dependency information can be used in the text modeling to improve the concept relevance detection. Also, we will consider further addition of name entity

recognition methods, which can extract the information about the biomedical concepts outside of the ontologies in texts. We are planning to use this information to develop a richer presentation of text and find relationships between the publication text and the queried biomedical concept.

Acknowledgments. The authors would like to acknowledge Martin O'Connor and Siddharth Taduri for their comments on the approach. This research was supported in part by grant R01 MH87756 from the National Institutes of Health.

References

1. Muller, H.M., Kenny, E.E., Sternberg, P.W.: Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature. PLoS Biol. 2(11):e309. doi:10.1371/journal.pbio.0020309 (2004)
2. Young, L., Tu, S.W., Tennakoon, L., Vismer, D., Astakhov, V., Gupta, A., Grethe, J.S., Martone, M.E., Das, A.K., McAuliffe, M.J.: Ontology Driven Data Integration for Autism Research. 22nd IEEE International Symposium on Computer Based Medical Systems, pp. 1–7, Albuquerque, NM (2009)
3. Tu, S.W., Tennakoon, L., Das, A.K.: Using an Integrated Ontology and Information Model for Querying and Reasoning about Phenotypes: The Case of Autism. AMIA Annual Symposium, pp. 727–731, Washington, DC (2008)
4. Hus, V., Pickles, A., Cook, E.H., Risi, S., Lord, C.: Using the Autism Diagnostic Interview-Revised to Increase Phenotypic Homogeneity in Genetic Studies of Autism. Biol Psychiatry. 61(4), 438–448 (2007)
5. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation, <http://www.w3.org/TR/2004/REC-owl-features-20040210/> (2004)
6. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. <http://www.w3.org/Submission/SWRL/> (2004)
7. Open Directory Project, http://www.dmoz.org/
8. Google support on snippets, http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=35624
9. Cooper, W.S.: Fact Retrieval and Deductive Question-Answering Information Retrieval Systems. J. ACM. 11(2), 117–137 (1964)
10. Miliaraki, S., Androutsopoulos, I.: Learning to Identify Single-snippet Answers to Definition Questions. 20th International Conference on Computational Linguistics, Geneva, Switzerland (2004)
11. Radev, D.R., Prager, J., Samn, V.: Ranking Suspected Answers to Natural Language Questions Using Predictive Annotation. 6th Conference on Applied Natural Language Processing, pp.150–157, Seattle, WA (2000)
12. Joho, H., Sanderson, M., Retrieving Descriptive Phrases from Large Amounts of Free Text. 9th ACM Conference on Information and Knowledge Management, pp. 180--186, McLean, VA (2000)
13. Salton, G., Wong, A., Yang C.S.: A Vector Space Model for Automatic Indexing. Commun ACM. 18(11), 613–620 (1975)
14. Porter stemmer, http://tartarus.org/~martin/PorterStemmer
15. List of English stop words, http://members.unine.ch/jacques.savoy/clef

Social and Semantic Computing in Support of Citizen Science

Joel Sachs and Tim Finin

Computer Science and Electrical Engineering
University of Maryland, Baltimore County
{jsachs, finin}@cs.umbc.edu

Abstract. We describe our ongoing work on using social media as a platform for citizen science. Building on our previous work of facilitating citizen science observations, and using RDF to integrate them with existing biodiversity knowledge, we are currently building Facebook Apps that will enable the reporting of observations, as well as the browsing and tagging of existing observations. The tagging capability serves two main purposes. First, it permits (and, we hope, encourages) multi-stage crowdsourcing for image identification. Second, it serves as a driver of ontology evolution, and permits experiments on potential working relationships between expert-engineered ontologies, and tag-based folksonomies.

Keywords: Semantic Web, Social computing, Biodiversity informatics, Citizen science, Collaborative ontology development

1 Introduction

Species' geographic distributions and phenology (the timing of life cycle events) are changing rapidly in response to climate change, new pathways of migration, and other factors. Observations by amateurs are often crucial in understanding this response. Our previous work [1] investigated social computing mechanisms for publishing citizen science data on the semantic web, where it can be integrated with other sources of biodiversity and biocomplextity data (e.g., range maps, food webs, evolutionary and taxonomic trees; conservation and invasiveness status, etc.) already exposed as RDF. The system concept we envision is a "global human sensor net" – a data stream that can be mined for species of interest (e.g., invasive, threatened, etc.) and anomalies (e.g., species out of their known range.); and which supports drilling down on observations to see what relevant related data (e.g., genomic, behavioral, etc.) already exists in our knowledge base or on the Semantic Web.

We are currently building Facebook Apps that will enable the reporting of observations, as well as the browsing and tagging of existing observations. The tagging capability serves two main purposes. First, it permits (and, we hope, encourages) multi-stage crowdsourcing for image identification. Second, it serves as a driver of ontology evolution, and permits experiments on potential working relationships between expert-engineered ontologies, and tag-based folksonomies.

The motivation behind using Facebook as the platform is to expose observing and tagging activity in users' news feeds, thus facilitating conversation around observa-

tional events. Observations and their tags are stored in Google Fusion Tables, which, in turn, are used to drive RDF representations. There is some contention over what RDF representations of ecological observations, and the ontologies behind them, should look like, and one of our desired and expected contributions are RDF representations of biodiversity data demonstrated to satisfy typical citizen science use cases. Thus, although our Facebook app is itself small in conceptual scope, it serves as a microcosm for a number of design decisions facing the semantic web for biodiversity informatics.

2 Related Work

2.1 Ontologies for Biodiversity

Occurrence Data

The central unit of biodiversity informatics is the *occurrence*, the observed presence of an organism at a particular place and time. Chapman [2] provides an excellent overview of the uses of primary biodiversity (i.e. occurrence data), include building range maps, niche modeling, and gap analysis. The exchange standard for biodiversity occurrence data is Darwin Core, a collection of several hundred terms for describing properties of an occurrence. An important aspect of Darwin Core is that it does not distinguish between data and metadata, so *identifiedBy*, *scientificNameID*, *verbatimCoordinates*, and *eventTime* are all simply properties of the occurrence. There are no mandatory fields.

The TDWG 2010 Annual Meeting in Woods Hole sponsored a day-long bioblitz with the aims of demonstrating TDWG standards in action, and evaluating their potential for uptake and use in real world, citizen science events to serve as a testbed for experiments in social and semantic computing for citizen science. After the bioblitz, a number of long discussions broke out on the tdwg-content regarding the appropriate direction for Darwin Core and related standards [3]. These included questions of normalization, dealing with multiple identifications (both competing and reinforcing), dealing with introduced and cultivated species, GUIDs for taxon concepts, and the meaning of "occurrence". No consensus has been reached, and there are two fairly well developed approaches on the table that we are aware of: that of deVries [4]; and that of Baskauf/Webb [5]. In addition, there is our own representation, which we used to represent the bioblitz data [6]; this serves more to explore the limits of what is possible with a casual approach to knowledge representation, than it seeks to compete with the other two, more principled, approaches, as a possible standard.

Observational Data

We are often interested in knowing more than whether or not a species is present at a location. We may want to know quantitative measurements of physical charecteristics, or qualitative descriptions of phenophase, or descriptions of ecological interactions. Biologists' field note books are notoriously idiosyncratic, and there are a number of proposed models, and, more recently, ontologies, that have been proposed to accommodate the full diversity of observational practice. These include OBOE (the

Extensible Observation Ontology), Prometheus, Delta, and EQ (the Entity-Quaality Model), all of which decompose the observational process in slightly different ways.

2.3 Collaborative Ontology Engineering

Web 2.0 was interpreted in a number of ways, in regards its relationship to the semantic web. For much of 2005 and 2006, it was in vogue to refer to Web 2.0 as the *lowercase semantic web.* This term conflated a number of things: the success of free-tagging to attach keywords to non-text objects; the folksonomies that resulted from said tagging; the embedding of semantics within HTML; and the notion that semantics is best built from the bottom up, rather from the top down.

Almost immediately, upper case Semantic Web researchers sought ways to harness the obvious power of socially created semantics to drive the "real" semantic web. Conceptually, we can divide the resulting collaborative knowledge engineering efforts into two categories: those in which the participants know that they are collaboratively building ontologies, and those in which the the ontologies (or other KR artifacts) emerge from the behaviour of users seemingly engaged in other, non-KR, activities. A large literature exists in both areas, and Angeletou et al. [7] provide a useful guide. Here, we describe only the work most relevant to our own.

Deliberate KR

Siorpaes and Hepp [8] describe a wiki-based approach to marrying ontology engineering and collective intelligence. They contrast *engineering-oriented* ontology design (by far the dominant paradigm) with a *community-oriented* approach, and motivate the need for the latter by listing three main advantages: inefficiency of the engineering oriented approach at keeping up with changing conceptual dynamics; distribution of the KR burden; and higher likelihood of community buy in.

KR as an artifact of user behaviour

Passant describes a system [9], in which tags are associated to concepts in an ontology. If a tag can't be mapped into the ontology, the knowledge engineer takes this as a clue that the ontology needs revision. Thus the traditional domain expert/knowledge engineer partnership is preserved, but with the domain expert role being replaced by the collective wisdom of the community. Passant's focus was information retrieval, where the only reasoning is using subsumption hierarchies to expand the scope of a query, but the principle should apply to other reasoning tasks as well.

Pitts [10] noted that tagging appears to have hit an innovation plateau because it is difficult for users to add more than shallow, impressionistic meaning to a subject, and worked on two projects, Memecat and Listgasm, to encourage meaningful tagging. In order to add the third "predicate" dimension to the tagging of a subject, he provided cues as to what the tagging context is when a user enters tags.

3 Facebook as a Platform for Citizen Science

Facebook may be an excellent platform for citizen science. Incorporating observational events in a user's news feed serves to expose the event to many potentially interested parties, fosters discussion around the event, and promotes discovery of the reporting tool, thereby resulting in more observations. We describe two apps that we are currently developing. One, iDentify, enables multi-stage crowdsourcing of images. The other, iPhenology, enables the reporting of phonological observations. Both apps result in data being published in RDF, and provide us the opportunity to experiment with RDF design patterns for representing biodiversity information. Using tag clouds to annotate the images also enables us to experiment with relating folksonomies and ontologies.

3.1 iDentify

After the bioblitz held at TDWG 2010, we had several hundred unidentified photos. To address this, a webpage (the Taxonomizer) was set up which presented users with an unidentified image, and requested classification. But this resulted in very few new identifications. Two issues with Taxonomizer were i) no one knew about it; and ii) potential users had to sit through many images that they did not recognize before coming to images that they did. iDentify addresses this by allowing identification to occur in stages. If an image is tagged "butterfly", for example, the butterfly experts can look at it to classify it further. Experts can learn about the image either by seeing a post on their wall that says "Your friend has just tagged XYZ 'butterfly'", or by adjusting their settings to show only pictures tagged butterfly.

3.2 iPhenology

Phenology is the study of the timing of life cycle events. For plants, these include first flower, first leaf, leaf senescence, etc. For animals, these include nest building, mating, migration, food gathering, etc. Two major citizen science initiatives in the U.S. capture phonological data: the National Phenology Network, and Project BudBurst. They each provide a controlled vocabulary for describing phenological events. A few things worth mentioning are: i) these two vocabularies use identical terms to mean slightly different things; ii) each vocabulary uses terms not in the other; iii) the NPN vocabulary was revised in the Spring of 2011, illustrating that it is still in flux. In addition to the evolving "standard" phenophase vocabularies, there is rich scope for unexpected, unconventional phenophase description. For example, there is growing interest in tapping into aboriginal knowledge to understand the Boreal Forest's response to climate change, and aboriginal terminology is likely to differ considerably from the terms already defined. Thus the iPhenology app we are developing seeds a tag cloud with terms from these vocabularies, prompts users to select terms from the cloud, and also to free tag where appropriate.

4 Representing the Data in RDF

Darwin Core

One hypothesis is that ontologies for the artifacts of human behaviour should be less constrained than ontologies for the natural world. So, in representing Darwin Core in RDF, we are not concerned with relating the concepts of occurrence, event, location, specimen, etc., through the use of intricate collections of *is_a*, *has_a*, and *part_of,* relations; and heavy use of domain and range constraints, and functional and inverse functional properties. Rather, we see the appropriate place for such ontologies as being the controlled vocabularies that are used as the objects of Darwin Core (DwC) predicates, (rather than for relating DwC predicates themselves). In other words, we see more value in using ontologies to model biodiversity ("tree has_part fruit", "green is_a colour", "human is_a ape", etc.), than in using them to model biodiversity informatics "observation has_part individual", "individual has_part taxon concept", etc.). Therefore, rather than defining an occurrence semantically - for example as the intersection of an event, an individual organism, and an observer - we consider it purely syntactically, as a tuple of time, location, and individual, together with some optional properties.

Flat vs. Hierarchical Ontologies

The notion persists that anything flat is not a "real" ontology, or somehow not semantic. But semantics accrue via human agreement, and do not depend on the topology of the representation. Consider, for example, the following two representations of an occurrence. In the first, *scientificName* is a property of *Occurrence*, while in the second it is a property of *Identification*, which is itself a property of *Individual*, with *Individual* being a property of *Occurrence*.

```
<Occurrence>
  <scientificName>mus musculus</scientificName>
  <individualID>145</individualID>
<Occurrence>


  vs.
<Occurrence>
  <hasIndividual rdf:resource="http://myMuseum.org/specimens?id-145"> /
</Occurrence>

<Individual rdf:about="http://myMuseum.org/specimens?id-145">
  <hasIdentification
rdf:about="http://myMuseum.org/identifications?id=CD/>
</Identification>

<Identification rdf:about="http://myMuseum.org/identifications?id=CD">
  <scientificName>mus musculus</scientificName>
</Identification>
```

The semantics of the above are the same, namely: "There's a thing in the museum that someone thinks is a mouse." We know that, in a sense, semantics transcends worldviews; otherwise people would never understand each other. Often, with no loss

of semantics, the model can be left out of the representation; data can be represented simply as a series of key-value pairs, and then the consumers can ingest the data into their own models.

For representing phenophases, we forgo (for now) the observational ontologies mentioned in Section 2, and instead make use of two terms from the Darwin Core measurement class: *measurementType*, and *measurementValue*. This allows us to embed the phenophase observation within a Darwin Core occurrence record as, e.g.

```
DwC:measurementType phenophase
DwC:meaurementValue first_flower
```

To the extent possible, we represent competency questions as sparql queries (see, e.g., 11), and use these to evaluate our approach.

5 Conclusions

Our current development effort is aimed at answering three questions: Can appropriate tag-cloud interfaces serve as feedback mechanisms for ontologies, and be used to propose new terms?; Can simple RDF representations of biodiversity data support citizen science use cases?; and Is Facebook a good platform for citizen science? We invite comments on our approach, suggestions for further use cases.

6 References

1. Andriy Parafiynyk, Cynthia Parr, Joel Sachs and Tim Finin, Adding Semantics to Social Websites for Citizen Science, Proceedings of the Workshop on Semantic e-Science, AAAI Press, June, 2007.http://ebiquity.umbc.edu/p/365
2. "*Uses of Primary Species-Occurrence Data*". Report for the Global Biodiversity Information Facility 2005. 111pp. (2005) Copenhagen: GBIF.
3. http://lists.tdwg.org/pipermail/tdwg-content/2010-October/thread.html
4. http://www.taxonconcept.org/
5. http://code.google.com/p/darwin-sw/
6. http://www.cs.umbc.edu/~jsachs/occurrences/TechnoBioblitzOccurrences.rdf
7. Angeletou, S., Sabou, M., Specia, L., Motta, E., (2007) Bridging the Gap Between Folkso- nomies and the Semantic Web: An Experience Report. Workshop: Bridging the Gap be- tween Semantic Web and Web 2.0, European Semantic Web Conference.
8. Katharina Siorpaes and Martin Hepp: myOntology: The Marriage of Collective Intelligence and Ontology Engineering, in Proceedings of the Workshop Bridging the Gap between Semantic Web and Web 2.0 at the ESWC 2007, Innsbruck, Austria, June 7, 2007.
9. Alexandre Passant, Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs: Theoretical background and corporate use-case, Proceedings of the International Conference on Weblogs and Social Media, AAAI Press, March, 2007.
10. Blog post: http://www.semanticwave.com/blog/archives/2008_01.tt
11. http://www.cs.umbc.edu/~jsachs/occurrences/queries/sample.txt

Unresolved Issues in Ontology Learning
- Position Paper -

Amal Zouaq[1, 2], Dragan Gasevic[2] and Marek Hatala[3],

[1] Royal Military College of Canada, [2]Athabasca University, [3] Simon Fraser University
amal.zouaq@rmc.ca, dgasevic@sfu.ca, mhatala@sfu.ca

Abstract. Despite a number of approaches to ontology learning in the last decade, there are still a number of challenges that need to be tackled by the research community. This paper describes some of these challenges and sketches some ideas that might be beneficial for solving them.

Keywords: ontology learning, Semantic Web, challenges

1 Introduction

Ontologies are a fundamental knowledge representation structure in modern Artificial Intelligence. They are also an essential component of the Semantic Web, which uses domain ontologies to conceptualize a domain through the definition of concepts, relationships, axioms and rules. However, this heavy reliance of the Semantic Web on domain ontologies also hinders its development, as building and maintaining domain ontologies is a highly error-prone and time-consuming process. Not only does the Semantic Web require domain ontologies, but it also requires semantic markup of Web content once domain ontologies are available, which is again a tedious and non-scalable task if it is done manually. To alleviate this bottleneck, the Semantic Web community has been investigating for more than a decade how to automatize ontology building and maintenance through ***ontology learning***. Various ontology learning systems like Text-To-Onto [4], Text2Onto [7], Ontolearn [3], OntoGen [5], Abraxas [8], Texcomon [2] and OntoCmaps [1] have been proposed. In general, these tools extract ontological structures from text corpora.

This paper aims at identifying the challenges facing the ontology learning tools, and opens some questions on the way these challenges might be solved.

2 What is ontology learning?

It is now widely accepted in the community that ontology learning refers to learning its constitutive components in OWL[1]: concepts (classes), taxonomy, conceptual relationships (OWL Object Property), attributes (OWL Data Type Property), axioms (De-

[1] Web Ontology Language

fined classes) and axiom schemata (disjointness, functional properties, transitive properties, etc.). However, one can notice that the majority of the approaches focus on concept and taxonomy learning [4, 7], with very few attempts to develop the other levels [1, 17]. Through our research, exploration of literature and interaction with different end-users, we have identified a number of issues that are, to our opinion, not satisfactorily resolved or dealt with in the research community.

3 Text Understanding

The issue of text understanding refers to the ambiguity and complexity of natural language and raises the question of the availability of NLP tools able to deal with this complexity. In fact, there has been considerable progress these last few years in computational syntax and semantics with the development of robust statistical syntactic parsers and wide-coverage semantic parsers [13]. These advances will certainly facilitate the understanding of texts but it still remains true that current knowledge extraction techniques are fragmentary and generally work at the sentence-level. Building wide-coverage semantic parsers would mean a broader perspective at the discourse level with the incorporation of techniques such as anaphora resolution and discourse representation structures [15]. However, to the best of our knowledge, there is no ontology learning tool which currently adopts this approach due to the complexity of the task. In fact, ontology learning tools generally rely on shallow NLP techniques and statistical methods [7]. Moreover, even with the progress in NLP-based tools for syntactic and semantic analysis, one should expect that extending the coverage of the extraction would also result in more noisy results. Dealing with this noise is another issue that we address in Section 6. Finally, semantic analysis, as practiced by the computational semantic community, adopts formal representations that can take the form of very detailed logical expressions. However, as stated by [12], purely logical approaches produce representations that are not yet robust enough to handle real text corpora. From another perspective, since current works on ontology learning rely mainly on shallower NLP or statistical methods, they fail to handle semantic phenomena such as negation and quantification and thus are unable to produce rich conceptual relations and axioms. To overcome these shortcomings, we advocate an approach to semantic analysis which takes a middle stance between such formal approaches and shallower approaches.

4 Knowledge Extraction

As previously said, the field of ontology learning theoretically covers the extraction of a number of ontological layers in increasing order of complexity. In reality, due to their reliance on shallow NLP methods, the majority of the approaches only covers the extraction of concepts and taxonomies, and generally fails to address the more complex-levels. Thus, the implementation of deeper NLP methods is a must [16]. In particular, conceptual relationships and axiom extraction seem to be lacking in the state-of-the-art, with the exception of very few works [1, 17]. In the best case, most of

the available NLP approaches to ontology learning are based on regular expressions. One disadvantage of regular expressions is that they might not discover long-distance dependencies, or they might fail to appropriately extract the right knowledge from complex structures. In our previous work [1, 2], we have proposed patterns based on dependency grammars with a syntactic-semantic interface that transforms a syntactic representation into a "semantic" one. However, similarly to the majority of ontology learning approaches which rely on a fixed number of regular expressions, our pattern knowledge base was created manually, which limits its coverage. Implementing *automatic methods for pattern learning* is one challenge that should be tackled by the ontology learning community, with pattern weighting schemes that indicate the confidence or reliability of each discovered pattern. Moreover, such a learning method would provide also a way to learn *domain-dependent patterns* as well. In fact, this research is important in order to evaluate how far we can go with the domain independence paradigm, but we are also fully aware that we might hit a limit at some point. Defining this limit would be of interest to the research community and would define a clear-cut architecture with some domain-independent and domain-dependent layers.

Besides pure knowledge extraction issues, it is also of tremendous importance to start considering how ontology learning can effectively help domain experts in their work (e.g., biological data curators) [19]. In fact, current prototypes do not really allow for much interaction with the expert. Given that ontologies are a way to formalize expert knowledge, and that some fields rely heavily on very large ontologies (e.g., biomedicine), there is a need to develop an ontology learning platform which would suggest not only new concepts and relationships to the expert, but would suggest also appropriate resources (definitions, web pages, and papers) related to a given ontological item, and would exhibit active learning capabilities by considering expert input.[2]

5 Ontological Structures Labeling

As ontological structures are learned from texts, ontology learning often takes the form of learning linguistic or lexical items. This approach is motivated by the fact that domain ontologies often represent *an interface between human and machines* rather than purely logical machine-readable metadata. However, this lexical-based approach might also lead to some problems. Firstly, some domains such as the biomedical field have evolving terminologies (e.g. known genes can be renamed) [18]. Maintaining lexical ontologies in this case seems to be a huge hurdle for the domain expert. Secondly, this creates the problem of the effective label to be associated to the ontological item (e.g. stem, lemma). In the case of relationships, this problem is even harder to solve: which lemma can we assign for example to the relationship *X can be described with Y?* If we choose *"describe",* then what is the conceptual difference with the relationship *X describes Y?*

In general, an ontological element (class, relationship) is conceptually separate from its labels, which can take various forms from a language to another (Semantic Web (en), Web sémantique (fr)) and even from a domain to another. However, to

[2] Many thanks to Prof. Melissa Haendle and Prof. Carlo Torniai of Oregon Health Sciences University for fruitful discussions on the needs of the biomedical community.

keep this notion of interface between human and machines and facilitate the ontology reading for a domain expert, there is a need to identify naming conventions and standard annotations for ontological items to increase their recognition-velocity, i.e. the ability to quickly grasp the meaning of a term via its name, for domain experts [14] but also for machines. In [14], a set of annotations associated to each ontological element is proposed such as “Display name” (the name appearing in the ontology structure) or “lexical variant”. A similar standard nomenclature would allow a certain consistency in the output of ontology learning tools. In our opinion, the “Display name” should not be related to the label contrary to what is being currently done by all ontology learning tools but should be a semantic free identifier with a set of semantic annotations. This would help the management and evolution of ontologies.

6 Ontological Structures Filtering

As we already mentioned, ontology learning extracts lexical items from texts. The question is how to identify important lexical items that should be promoted as ontological structures in the domain ontology. This also raises the issue of the nature of a concept, which is here considered as a relevant/important term. For example, while building an ontology about SCORM, an eLearning standard, the term “SCORM” is certainly relevant. However, it does not admit an instance as there is no object that could be of type “SCORM”. Nevertheless, this term will be a candidate class in the majority of ontology learning systems, and this is also the approach adopted in our own work [1, 2]. Generally, a concept is considered as a nominal expression (including multi-word expressions) that is relevant to a domain. However this widely adopted definition also raises questions. For example, given the following expressions, one can wonder if they are acceptable in a domain ontology: XML representation of content organization (yes), Aggregation of content object (may be?) and Educational use of SCORM content model component (may be?). As it can be seen, it is not always easy to differentiate what is a relevant expression (concept) and what is not.

Besides this question on the nature of concepts, there is also the notion of the statistical ranking or importance of knowledge items. In general, some ontology learning tools such as OntoGen [5] do not assign any explicit score to the extracted knowledge items while others, such as Text2Onto [7], allocate some score to the extracted knowledge using traditional metrics from information retrieval such as Relative Term Frequency (RTF), TF-IDF, or Entropy. This score is used to determine the relevance of a given item but is not used to automatically filter out the extraction. However, by looking at the precision/recall results of such systems (see for example [9, 10]), which are very low, it is obvious that there is much room for improvement both at the extraction level and at the filtering level.

Another popular weighting scheme is the use of the number of hits of a search engine to calculate the probability of a given item. However, using search engines comes at the cost of a number of issues [11] generally ignored by the ontology community. For example, search engines do not stem or lemmatise the terms. Thus, all combinations of a given term should be submitted to the search engine to obtain an appropriate (if not entirely correct) web frequency. Moreover, the number of hits refers to the number of pages containing the term rather than the frequency of the term

itself. For all these reasons, relying on NLP-specific resources such as Google N-gram Corpus[3] might be an interesting avenue to explore by the ontology learning community.

Finally, graph-based metrics (Betweenness, Degree, Hits, and PageRank) were also proposed to identify relevant ontological structures in our work [1]. To our knowledge, this is the sole initiative that uses these types of metrics for ontology learning. Surprisingly, these graph-based metrics outperformed standard term relevance schemes such as TF-IDF or frequency of co-occurrence in our experiments. However, these results need to be replicated on several domains and further research need to be devoted to that aspect.

7 Ontology Evaluation

One of the last but not least issues of the ontology learning community is how to handle the appropriate evaluation of the extracted ontologies due to the lack of gold standards and resources. This hinders the development of the ontology learning field and does not enable the proper evaluation of the developed tools. While we notice a number of competitions in information retrieval (e.g. TREC[4]) or information extraction (e.g. ACE[5]), such resources do not exist for ontology learning. The experience also shows that a field starts to be more mature when resources and tools can be shared and compared. Therefore, the ontology learning community would need corpora coupled with gold standards (incorporating all the constituent knowledge items of an ontology and not only glossaries and taxonomies) mimicking the content of corpora in various domains to effectively evaluate the tools. In fact, it does not seem fair for an automatic tool to compare its output to an ontology built manually by domain experts for a number of reasons:

- The ontology learning tool does not have access to the background knowledge of experts, which is one of the oldest problems in AI. An extracted ontology can only mimic or represent the content of the knowledge source. Thus comparing such an ontology with an extensive ontology built by domain experts is not satisfactory, as it does not evaluate the possibilities of the tool but rather the lack of background knowledge of the tool.
- Another challenge is related to the domain coverage of texts. Generally, even the most extensive collection of texts will not cover sufficiently a domain. Some researchers have advocated using the Web to resolve this issue (e.g. [17]), but this may also introduce more noise, hence urging the need for efficient filtering mechanisms as explained in section 6.

As a conclusion, we believe that the first challenge of an ontology learning tool should be to adequately extract meaningful information from text (with the least possible omissions of important knowledge). Thus the need of corpora and ontological gold standards is one of the most acute issues of the field.

[3] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13
[4] http://trec.nist.gov/
[5] http://projects.ldc.upenn.edu/ace/

8 Conclusion

Ontology learning is a complex process that, besides integrating deeper NLP techniques than what is currently being done in the field, is of an acute need for appropriate evaluation resources. This paper summarizes some of the current issues and open questions of the field.

Acknowledgments. This research was partly funded by the NSERC Discovery Grant Program and by the Burroughs Wellcome Fund.

References

1. Zouaq, A., Gasevic, D. and Hatala, M. (2011). Towards open ontology learning and filtering, Information Systems, Volume 36, Issue 7, Pages 1064-1081.
2. Zouaq, A. (2008). An Ontological Engineering Approach for the Acquisition and Exploitation of Knowledge in Texts, PhD Thesis, University of Montreal (in French).
3. Navigli, R. and Velardi, R.: Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. Computational Linguistics 30(2): 151-179 (2004)
4. Maedche, A. and Volz, R. (2001). The ontology extraction maintenance framework Text-To-Onto, in Proc. of the Wshp on Integrating Data Mining and Knowledge Management.
5. Fortuna, B., Grobelnik, M., and Mladenic, D. (2006). Semi-automatic Data-driven Ontology Construction System. Proc. Of the 9th Int. Multi-conf. on IS, pp. 309-318, Springer.
6. Frantzi, K.T. and Ananiadou, S. (1999). The C/NC value domain independent method for multi-word term extraction, Journal of NLP 3(6): 145-180.
7. Cimiano, P. and Völker, J. (2005). Text2Onto. NLDB 2005, pp. 227-238, Springer.
8. Brewster, C.A. (2008). Mind the gap: bridging from text to ontological knowledge, Ph.D. Thesis, University of Sheffield.
9. Brewster, C., Jupp, S., Luciano, J., Shotton D., Stevens R. and Zhang Z. (2009). Issues in learning an ontology from text. BMC Bioinformatics 10, S1.
10. Cimiano, P. Hotho, A. and Staab S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. J. Artif. Int. Res. 24, 1, 305-339.
11. Adam Kilgarriff. 2007. Googleology is Bad Science. Comput. Linguist. 33, 1 (March 2007), 147-151.
12. MacCartney, B. (2009). Natural language inference. Ph.D. dissertation, Stanford Un.
13. Bos, J. (2008). Introduction to the shared task on comparing semantic representations. In Proc. of the 2008 Conf. on Semantics in Text Processing, pp. 257-261, ACL.
14. http://ontogenesis.knowledgeblog.org/948
15. Kamp, Hans and Reyle, U. 1993. From Discourse to Logic. Kluwer, Dordrecht.
16. Zouaq, A. (2010). Shallow and Deep Natural Language Processing for Ontology Learning: a Quick Overview, In Ontology Learning and Knowledge Discovery Using the Web.
17. Sanchez, D. and Moreno, A. 2008. Learning non-taxonomic relationships from web documents for domain ontology construction. Data Knowl. Eng. 64, 3, 600-623.
18. Tuason, O., L. Chen, H. Liu, J.A. Blake, and C. Friedman. Biological Nomenclature: A Source of Lexical Knowledge and Ambiguity. In: Proc. of Pac Symp Biocomput. 2004. p. 238-49.
19. Winnenburg, R, Wächter, T, Plake, C, Doms, and A, Schroeder, M. Facts from text: can text mining help to scale-up high-quality manual curation of gene products with ontologies? Brief. Bioinform. 2008;9:466–478

Towards Integration of Semantically Enabled Service Families in the Cloud

Marko Bošković[1,2], Ebrahim Bagheri[1,3], Georg Grossmann[4], Dragan Gašević[1,2], Markus Stumptner[4]

[1]Athabasca University, Canada
{firstname.lastname}@athabascau.ca
[2]Simon Fraser University Surrey, Canada
[3]University of British Columbia, Vancouver, Canada
[4]University of South Australia, Australia
{georg|mst}@cs.unisa.edu.au

Abstract. Success of a Software Product Line (SPL) typically induces increase of requirements that expand over the expertise of its initial company. In the context of cloud computing, where SPLs are deployed in the form of business process families that are offered over the Internet, this expansion requires partnering with other available families. With the increasing number of companies that offer their solutions in the cloud, there is a need for tools and methods for integration of configurable business processes. In this position paper, we propose a methodology for integration that employs ontologies and Semantic Web technology, and propose a tool support that supports the proposed methodology.

1 Introduction

Motivated by the fact that different stakeholders have similar requirements, Software Product Line Engineering [1] (SPLE) argues the development of similar software systems as a whole, herewith sharing many assets and increasing reuse ability. An SPL is customized for every customer by selecting the set of most desirable features. Beside SPLE, Service-oriented Computing is another computing paradigm that promotes reuse where services enable rapid and easy composition of loosely coupled distributed software applications, and provide general computational elements that can be reused across different domains [2]. At this moment, there is a significant research for integrating these two software engineering paradigms, e.g. [3,4,5,6,7,8]. Recently, benefits of this synergy have been seen in the context of cloud computing [9], where synergistic solutions for service-oriented applications and SPLs are delivered over the Internet in the form of Business Process Families (BPFs) that are being configured for each user independently, while keeping BPFs, supporting systems software, hardware and maintenance, away from her [10].

The success of SPLs usually leads to their expansion that reaches a level that exceeds the innovation capabilities of one organization [11]. In such an expansion, companies converge different domains, often those that were not their primary business. In the context of BPFs in the cloud, this requires partnering of already existing BPFs. Therefore, there is a need for methods and tools for integration of BPFs.

Contemporary methods for integration of SPLs are mostly formal, and assume only feature equivalents across different families, e.g. [12,13,14,15]. However, in practice, because features are typically not equivalents, we consider integration of families as

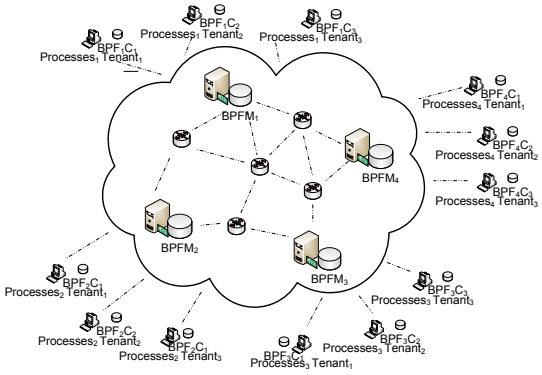


Fig. 1. Families of business processes in cloud computing (inspired by van der Aalst [10]). (BPFM=business process family model, BPFC=configuration)

an engineering task that cannot be fully automated. Therefore, we provide a method and propose a tool support that heavily uses ontologies [16] and Semantic Web technologies [17] for semantic annotation of BPFs, that can be used to automatically derive interdependencies and allows for semi-automated integration.

2 The Proposed Method

Cloud computing is an emerging computing paradigm that promotes delivery of applications to users as services over the Internet while keeping the hardware, systems software and system maintenance away from her [9]. Therefore, each BPF in the cloud is distributed and independently deployed [10], as illustrated in (Figure 1).

Each BPF is specified with Business Process Family Models (BPFMs) consisting of artifacts specifying the problem space, the solution space, and the mappings between problem and solution spaces [18]. The solution space is typically a Business Process Model Template (BPMTs) [19], i.e., superimposition of all business process variants. The problem space, on the other hand, represents all possible features of family members and typically is captured with feature models, a tree-like structure [20]. A BPF is configured for each user by selecting the desired features of the family. A feature selection, with the help of mappings, forms the final business process for a particular user. In the context of BPFs in the cloud, problem, solution, and mapping models are deployed to an external location on the Internet, while each tenant has his own customized configuration of the family, as shown in Figure 1.

SPLE generally consists of two life-cycles: Domain Engineering (DE) and Application Engineering (AE) [21][1]. In short, DE aims at the development of common assets (e.g. models, components, documentation) and configuration knowledge (typically feature models and mappings). AE is dedicated to the selection of appropriate features.

Our integration methodology considers integration of BPFs as a form of the DE. It builds upon the framework proposed by Linden et al. [21] and is depicted in Figure 2.

DE consists of *requirements engineering*, *domain design*, *domain realization* and *domain testing*. In our methodology, *requirements engineering* results in a fully inte-

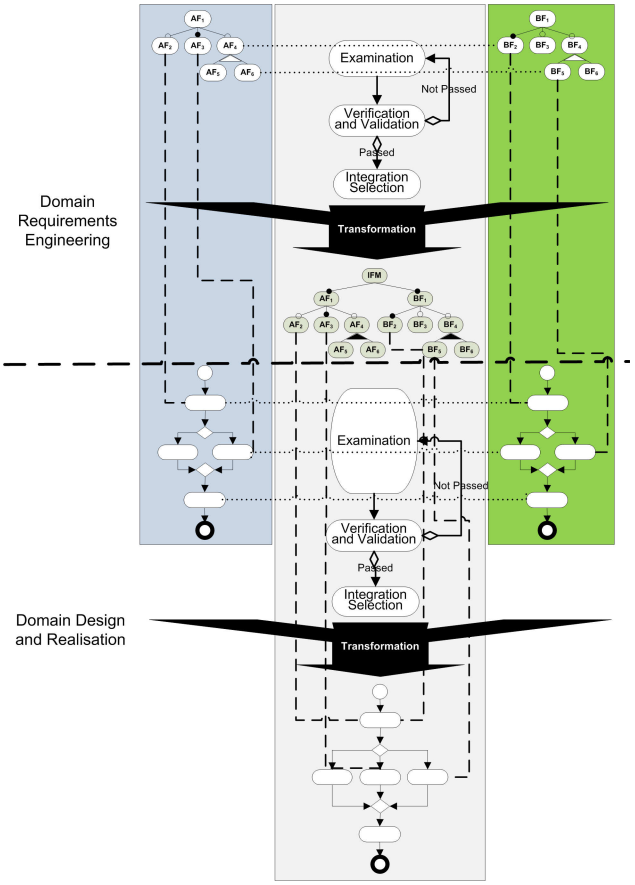


Fig. 2. Integration of families of service-oriented systems

grated feature model, while the *domain design and domain implementation* are one phase that results in integrated BPF. *Domain testing* is out of the scope of this paper.

In the *requirements engineering* phase we propose the following activities:

1. Examination of relationships between features of independent families. For example, in the integrated feature model, we can have features of different families that represent identical business processes by intention, but their actual realizations (extensions) are different. Some other examples of relationships are that can be found is that features represent business processes of different families with the same intension and extension (they use the same service), or that they are history related, meaning that one business process must be executed before the other one. We base our relationships on the ones identified by Grossmann et al. [22,23] (More on the relationships and their integration options can be found at: https://files.semtech.athabascau.ca/public/TRs/TR-SemTech-03052011.pdf). To automate this recognition we employ ontologies and Semantic Web technologie;

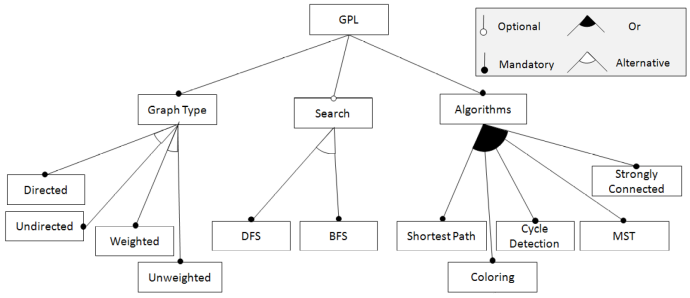


Fig. 3. A feature model of Graph Product Line

2. Verification and Validation of relationships. In the process of defining integrated feature models, there is a need for the validation of relationships between features with target customers and developers of different families, and verification whether the relationships are well specified, e.g., to recognize whether there are inconsistencies in the integrated feature model;
3. Integration selection is an activity where an integration engineer selects the appropriate choices for integration. Every relationship between features does not uniquely specify the configuration relationship, but rather provides a set of possible choices. For example, the integration engineer might choose to have in the integrated feature model, two features that are identical by intension but different by realization (extension). In such a case, the integration engineer might also choose to allow for mutually exclusive configuration, or that both can appear in the final application. The integration engineer selects this relationship from the set of available configuration relationships [0];
4. Transformation is the final activity, where selected integration patterns and initial feature models are inputs, and output is a feature model of the integrated families.

In the context of integration, *domain design* and *domain implementation* are the same phase, because the outcome is a business process template of already implemented families. We propose the same activities as in the *requirements engineering* model, namely: 1)**Examination** of relationships between business processes in business process models as proposed by Grossmann et al [22,23]; 3) **Verification and Validation** of relationships for semantic and well-formedness; 3) **Integration selection**, i.e., the selection of predefined integration options(e.g., the services with the same intention but different extension can be integrated in a way that at runtime their results are accumulated or that exactly one can be executed); 4) **Transformation** from input BPMTs to the integrated BPMT.

3 Foundations

3.1 Feature Modeling

A feature model is a means for representing the possible configuration space of all the products of a system product line (system family) in terms of its features. Typically, feature models are represented with feature diagrams in the form of a tree whose root node

represents a domain concept, e.g., a domain application, and the other nodes concept property, e.g., domain application functionality, modeled in a way to capture commonalities and variability among product family variants. The rest of features are classified as:

- **Mandatory** feature: the feature must be included in a product if its parent feature is selected.
- **Optional** feature: the feature may or may not be included if its parent is selected.
- **Or feature group**: from a set of Or feature group, any non-empty subset of features can be included if their parent feature is selected.
- **Alternative feature group**: from a set of alternative features, only one feature can be included if their parent feature is selected.

Additional constraints are defined on the feature models, named integrity constraints. Two main constraints are: **includes** – selection of a given feature requires the inclusion of another feature; and **excludes** – that specifies mutual exclusion of two features. An example of a feature model of Graph Product Line is given in Figure 3.

3.2 Semantically-enhanced Business Process Model Templates

As previously stated, a Business Process Model Templates is a superimposition of all members of a BPF [19]. Web services are seen as main means for operationalization of business processes and accordingly, BPMTs [2,24].

The main characteristic of Web services is that they can be deployed over large scale networks such as the Web; hence, they need to and indeed carry machine processable descriptions that properly inform other programs of their operations and how they can be properly invoked. One of the limitations of contemporary Web services is that their description lacks meaningful explanations or in other words semantic descriptions. Semantic Web services add capability of describing structural and behavioral semantics to Web services by providing the means to expressively annotate Web services with shared conceptualizations in the form of ontological concepts [25]. Ontologies provide agreed upon and formal domain specifications [16] based on Semantic Web markup languages such as OWL and DAML and are shared by different software systems and applications. Not only does this sharing of knowledge allow software systems to search for suitable Web services based on syntactical matches, but to also consider semantic relevance within the matchmaking process. BPMTs that use Semantic Web Services as operationalizations, are called Semantically-enhanced Business Process Model Templates.

4 The proposed tool support

4.1 Feature Model Representations

Several different formats for storing and manipulating feature models have been proposed in the software product line community including XToF, SXFM and TVL [26]. Although the representations of these serializations are different, the semantics of the languages are quite similar and they can be easily transformed to one another. Within our framework, Figure 4, we model feature models with *Semantic Annotations for Feature Model Description Language* (SAFMDL), and serialize them as profile for feature

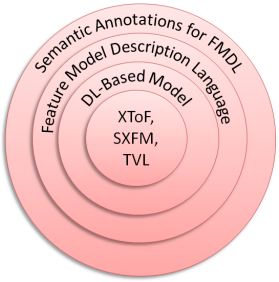


Fig. 4. The Onion Architecture for SAFMDL

modeling based on Web Ontology Language (OWL). However, AUFM Suite that supports our framework, provides mechanisms to convert to and from other serialization into SAFMDL.

As shown in Figure 4, the core of SAFMDL is a Description Logic based model specified with Feature Model Description Language (FMDL). FMDL is a feature modeling profile that provides the standard concepts for developing a feature model. It is modeled based on OWL and can essentially be seen as an ontology for feature modeling. The structure of FMDL consists of the required concept and property definitions for instantiating a feature model, which corresponds to the feature modeling meta-model. The instantiation of the feature modeling meta-model is performed by providing ontology individuals (concept instances) for the FMDL concept definitions. Figure 5 depicts the details of FMDL in the Protégé ontology editor. FMDL feature models can also be developed within our AUFM Suite, which is a graphical Eclipse plugin for feature modeling.

As shown in Figure 5, the structure of a feature model is based on the two main concepts of Root and Feature, and two of its sub-concepts Mandatory and Optional. These concepts are shown on the Class Hierarchy panel (Box 1). A new feature model can be instantiated by providing individuals for each of these concepts. For instance, the Algorithm and GraphType features have been added to this feature model as mandatory features (Box 2). The relationships between the features are modeled through properties. The list of possible relationships between the features of a fetture model is shown in Figure 3 (Box 4). For instance, it can be seen that *Algorithm* has a *siblingRelationship* with both *GraphType* and *Search* features. It can also be seen that the *Root* of the feature model is named *GPL* (Graph Product Line) and that *Algorithm* is one of the direct children of the *Root* (Box 3).

The benefit of using DL-based feature models is that standard DL reasoning mechanisms can be used to derive and validate feature model configurations and also extended DL algorithms can even be used to detect and resolve inconsistencies within feature models. Besides the exact syntax and semantics of FMDL, it provides an additional advantage of providing grounds for being extended with additional capabilities without requiring structural changes. Since, FMDL is based on OWL, additional information or data can be added to it through the introduction of new *Class* or *Property* definitions. This has been exploited to further extend FMDL to support the semantic annotation of its elements, referred to as SAFMDL.

SAFMDL profile introduces three new properties that reference concepts within external shared ontologies. These additional properties, *selfModelReference*, *preCon-*

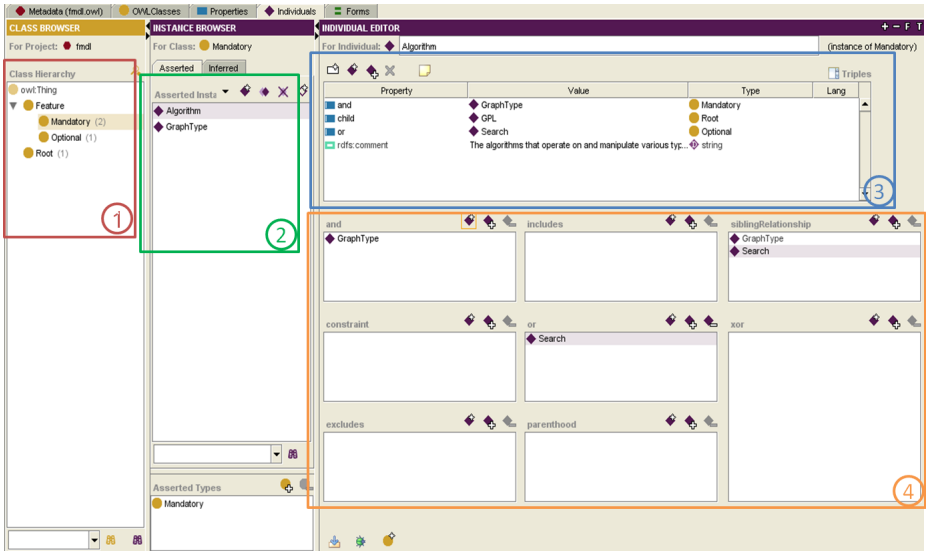


Fig. 5. The Structure of FMDL in Protégé

```
<fmdl:Mandatory rdf:ID="Algorithm">
  <safmdl:selfModelReference rdf:resource="http://monet.nag.co.uk/algorithm#Algorithm"/>
  <fmdl:or rdf:resource="#Search"/>
  <fmdl:and rdf:resource="#GraphType"/>
</fmdl:Mandatory>
```

Fig. 6. Semantically annotating feature model elements

ditionModelReference, and *postConditionModelReference* allow each feature to be further described by presenting what concept or notion the feature represents in the domain of discourse, what other notions it relies on for being realized, and which other concepts will be impacted by this feature, respectively. Given this capability, SAFMDL allows designers to qualify their design with meaning and hence avoid ambiguity and enhance communication, model sharing, better model realization, and finally integration. Figure 6 depicts an example where the Algorithm feature is grounded/annotated using the *Algorithm* concept within the MONET ontology. MONET is an ontology for describing and provisioning web-based mathematical services. With this annotation, the Algorithm feature is now confined with the semantic meaning attached to Algorithm in MONET and its scope is restricted by what defined clearly in that ontology.

4.2 Connecting Problem and Solution Spaces

As discussed earlier, we will need to move from the problem space (i.e., the feature model) into the suitable solution space (i.e., BPMTs). The main challenge towards the operationalization is to find the right Web services that both syntactically and semantically implement features that are available in a feature model.

Given that SAFMDL provides the means for describing feature models with semantic descriptions, it is possible to create a correspondence between the problem space

feature models and solution space semantic Web services. The semantic descriptions shared between both spaces can be seen as glue that can enhance the discovery of the most appropriate services for realizing the abstract software applications. In order to operationalize abstract product representations of the problem space, here are three sources of information that need to be completely integrated, namely 1) semantically annotated feature models; 2) semantically annotated Web services; 3) the sources of the semantic information,

These three sources of information are either expressed in a valid XML format or through some extensions of the RDF triple format; therefore, appropriate XSPARQL [27] queries can consolidate these sources of information and provide for the realization of problem space models using Semantic Web services. If we return to the example from Figure 6, and assume that a set of Web services is available to us that are annotated using SAWSDL [28] with concepts from subsets of the MONET ontology. In the example in Figure 7, we show that the Search feature from GPL can be operationalized using XSPARQL. As seen in the process shown in Figure 7, the first step is to extract the semantic annotation that describes the feature of interest (♣). This will provide the basis to search for Web services that are also annotated similarly. The valuable aspect of ontological semantic descriptions is that they provide meaningful hierarchical relationships; therefore, even if two concepts are not identical, they can still be related lower down or higher up the subsumption hierarchy. Concepts below another concept in the hierarchy can be seen as further specializations of that concept and can hence be relevant in the matchmaking process. For this reason, it is reasonable to look for Web services that are either directly annotated with the semantic annotation of the feature of interest or other concepts that are below it in the hierarchy (♥). The last step is to explore the set of available Web services that are annotated with acceptable ontological concepts (♠) using a suitable query. The outcome of this query is a list of Web services that have appropriate matching semantic descriptions to the feature of interest (Search).An expert designer or software developer would then need to review the matches and select the best one to operationalize that feature. In Figure 7, we have only checked for matches based on *sfmdl:selfModelReference* to save space but in reality checks also should be put in place for pre and postconditions as well.

4.3 Recognizing Relationships

As previously mentioned, by employing ontologies and Semantic Web technologies, all of our artifacts (feature models and business process templates) are annotated with semantic descriptions. These semantic descriptions can also be used to automatically derive the integration relationships between different features. For example, a similar query to the one presented in Figure 6, can be used to search for the features representing the identical business process. The only change in the query is that it should search for the exact match of the *Algorithm* concept.

In order to provide automatic recognition of interrelationships between feature models, we intend to provide a library of XSPARQL Queries, that automatically recognize relationships between features in feature models and business processes in business process templates. These queries are intended to be triggered in the *Elicitation* phase of the *Domain Requirements Engineering* for identifying of relationships between features in different feature models and in *Domain Design and Implementation* for finding of relationships between business processes in business process templates. Furthermore, more soffisticated ontology based techniques for automatic recognition can be employed, like

the ones used for service matchmaking [29,30] and business process matchmaking [31] based on similarity metrics proposed by Dijkman et al. [32].

4.4 Implementation Aspects

To support software developers for working with our proposed framework, we have started to implement the AUFM Suite - a chain of Eclipse based tools for development of Semantically-enhanced Families of Business Processes. So far, we have implemented the following tools:

- SAFDML Editor: This tool provides ontology representation of feature models, as described in Section 4.1
- rBMPN tool: for modeling the composition of features (represented as activities) using Business Process Modeling Notation 2.0 (BPMN2). Additionally, the tool provides facilities for modeling business rules over BPMTs.
- S-AHP tool: This tool goes beyond the work presented in this paper, and is used in AE phase of integrated BPF. The tool captures stakeholders' preferences in the terms of relative importance, and ranks features according using the implementation of our S-AHP algorithm [33].
  For the next stage of our development, we are working to integrate the XSPARQL language with our tooling support for formulating and executing queries on the repository of semantic Web services.

5 Related Work

Up to this day, several formal approaches exist for composition of feature models and solution space models. Feature model composition has been a topic of Acher et al. [15] and Segura et al. [14]. Acher et al. [15] introduce a domain-specific language for integration of feature models with operators for merging and inserting. Segura et al. [14], introduce an approach for automated merging of feature models, using graph transformations. In their work they provide a set of rules, and with the means of graph transformation, they perform the merge. Beside the fact that both of these approaches are focused only on feature models, they are formal and assume that there is a semantic equality between features that are merged. Our work takes an engineering perspective and assumes that there can be also different levels and semantics of equivalence. Due to this fact, our approach is semi-automated, and does not take the developers out of the process of integration. Rather it is an interactive process, where developers specify the semantic interrelationships and choose between different integration options.

Similar to formal composition of feature models, there are several approaches for formal composition of features in solution space models. Batory et al. [34] has introduced also an algebraic framework for specification of composition of features. Similarly, Erwig et al [12] also introduces a formal calculus for composition of different features in solution space models. However, our work, goes beyond these approaches and provides a semantics-based composition. Furthermore, Batory et al. and Erwig et al. focus on composing features of a single SPL, while in our work we focus on integration of SPLs. Finally, Apel et al. [13][35] introduces a feature algebra for language independent feature compositions. A feature is represented as a feature structure tree (FST), a language independent representation of a subset of the abstract syntax trees.

```
declare namespace xs    = "http://www.w3.org/2001/XMLSchema";
declare namespace rdfs = "http://www.w3.org/2000/01/rdf-schema#";
declare namespace fmdl = "http://ebagheri.athabascau.ca/spl/fmdl.owl#";
declare namespace safmdl = "http://ebagheri.athabascau.ca/spl/safmdl.owl#";
declare namespace rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#";
declare namespace gpl = "http://ebagheri.athabascau.ca/temp/gpl.owl#";
declare namespace wsdl = "http://www.w3.org/ns/wsdl";
declare namespace sawsdl = "http://www.w3.org/ns/sawsdl";


for $y from <http://ebagheri.athabascau.ca/temp/gpl.owl>
    where (
            gpl:search safmdl:selfModelReference $y. ♣
          )

    construct
            (
                {
                for $x from <http://ebagheri.athabascau.ca/temp/algorithm.owl>
                        where (
                                $x rdfs:subClassOf $y. ♥
                               )

                construct
                                (
                                    {
                                      let $doc := doc("http://ebagheri.athabascau.ca/temp/algorithm.sawsdl")
                                      let $operations := $doc//wsdl:operation

                                      for $o in $operations
                                      let $el := if ($o/@sawsdl:modelReference=$x) then $o/@name else "Not Suitable" ♠

                                      construct
                                        {
                                          [ :suitableService $el; ].
                                        }

                                    }
                                )
                }
            )
```

Fig. 7. A sample of XSPARQL Query for mapping problem and solution spaces

With this algebra, when two features are composed, they are merged only in the case when they have the same name and type. Our integration, goes beyond just a name and type based integration and facilitates semantics based integration.

To our knowledge, van Ommering was the first to observe composition (integration) of SPLs from the engineering perspective. In his work [36][37][38], he has introduced a notion of product populations, a set of SPLs whose members share many commonalities. In such context, (semi-) independent SPLs are developed by separated intra-organizational teams and later integrated into one variant rich product population. To support development of product population, van Ommerling introduces a lock-step process and a component model. The component model supports integration with the means of glue code. Our specification of interrelationships goes beyond glue code, and enables semantic based specification of interrelationships and semi-automated integration based on these semantic correspondences.

Recently, Bosch et al. [39] have proposed different process models for development and integration of SPLs in various global software engineering contexts. Our work focuses on the technical level of integration and can be applied in all engineering processes proposed by Bosch et al.

6 Conclusions and Future Work

In this paper, we have described a semantically enabled approach to the integration of Service Families in the Cloud. This task is a challenge specific to a leading edge en-

vironment where software engineering techniques are currently breaking new grounds along multiple dimensions: business processes evolve into service processes dynamically deployed in the Cloud; software product lines evolve into service families, with feature models being used to describe a more dynamic and flexible architectural style; integration technologies developed for business processes need to be extended to fit the service environment and so provide high level tool support in situations where traditional methods could not keep up.

We have described how a business process integration technology based on the semantic classification of correspondences and selection of integration patterns can be adapted to service families by using a process fragment classification approach for the extended feature models describing the services. Furthermore, we demonstrate how ontologies and Semantic Web technologies can be employed to automatically identify correspondences between business processes and features. We have given an example and described the tool support that can be employed for these tasks.

In the future, we are going to focus on completing the tool support and evaluation of the approach by applying it on realistic case studies.

Acknowledgments. This research was in part supported by Alberta Innovates – Technology Futures through the New Faculty Award program,

References

1. Pohl, K., Böckle, G., van der Linden, F.J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer (2005)
2. Papazoglou, M.: Whats in a service? In Oquendo, F., ed.: Software Architecture. Volume 4758 of LNCS. Springer (2007) 11–28
3. Lee, J., Muthig, D., Naab, M.: A feature-oriented approach for developing reusable product line assets of service-based systems. J. of Systems and Software **83**(7) (2010) 1123–1136
4. Medeiros, F.M., de Almeida, E.S., Meira, S.R.L.: SOPLE-DE: An Approach to Design Service-oriented Product Line Architectures. In: Proc. of the 14th Int. Conf on SPLsd. SPLC'10, Springer (2010) 456–460
5. Koning, M., Sun, C.a., Sinnema, M., Avgeriou, P.: Vxbpel: Supporting variability for web services in bpel. Inf. Softw. Technol. **51** (2009) 258–269
6. van der Aalst, W.M.P., Dreiling, A., Gottschalk, F., Rosemann, M., Jansen-Vullers, M.H.: Configurable process models as a basis for reference modeling. In: Proc. of BPM 2005 Workshops. Volume 3812 of LNCS. (2005) 512–518
7. Boffoli, N., Cimitile, M., Maggi, F.M.: Managing business process flexibility and reuse through business process lines. In: ICSOFT 2009 - Proc. of the 4th Int. Conf. on Software and Data Technologies, Vol. 2, Springer (2009) 61–68
8. Schnieders, A., Puhlmann, F.: Variability mechanisms in e-business process families. In: Proc. of the 9th Int. Conf. on BIS. Volume 85 of LNI., GI (2006) 583–601
9. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A View of Cloud Computing. Communications of ACM **53**(4) (April 2010) 50–58
10. van der Aalst, W.: Configurable services in the cloud: Supporting variability while enabling cross-organizational process mining. In: On the Move to Meaningful Internet Systems: OTM 2010. Volume 6426 of LNCS. Springer (2010) 8–25
11. Bosch, J.: The challenges of broadening the scope of software product families. Communications of ACM **49** (December 2006) 41–44
12. Erwig, M., Walkingshaw, E.: The choice calculus: A representation for software variation. ACM Trans. on SE and Methodology (to appear)

13. Apel, S., Lengauer, C., Mller, B., Kstner, C.: An algebra for features and feature composition. In: Alg. Meth. and Softw. Technology. Volume 5140 of LNCS. Springer (2008) 36–50
14. Segura, S., Benavides, D., Ruiz-Cortés, A., Trinidad, P.: Automated merging of feature models using graph transformations. In Lämmel, R., Visser, J., Saraiva, J.a., eds.: Gen. and Transf. Techniques in SE, Springer (2008) 489–505
15. Acher, M., Collet, P., Lahire, P., France, R.: Composing feature models. In: 2nd Int. Conf. on SLE (SLE 2009). Volume 5969 of LNCS., Springer (2010) 62–81
16. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. on Human-Computer Studies **43** (December 1995) 907–928
17. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (May 2001)
18. Czarnecki, K., Eisenecker, U.W.: Generative programming: methods, tools, and applications. ACM Press/Addison-Wesley Pub. Co. (2000)
19. Czarnecki, K., Antkiewicz, M.: Mapping features to models: A template approach based on superimposed variants. In: Proc. of the Int. Conf. on GPCE. LNCS, Springer (2005) 422–437
20. Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, A.S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute (1990)
21. Linden, F.J.v.d., Schmid, K., Rommes, E.: Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer (2007)
22. Grossmann, G., Ren, Y., Schrefl, M., Stumptner, M.: Behavior based integration of composite business processes. In: BPM. Volume 3649 of LNCS. Springer (2005) 186–204
23. Grossmann, G., Ren, Y., Schrefl, M., Stumptner, M.: Definition of business process integration operators for generalization. In: ICEIS (3), Springer (2005) 510–517
24. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
25. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic web services. IEEE Int. Sys. **16** (2001) 46–53
26. Boucher, Q., Classen, A., Heymans, P., Bourdoux, A., Demonceau, L.: Tag and prune: a pragmatic approach to software product line implementation. In: Proc. of the IEEE/ACM Int. Conf on Aut SW Eng. ASE '10, ACM (2010) 333–336
27. Akhtar, W., Kopecký, J., Krennwallner, T., Polleres, A.: Xsparql: traveling between the xml and rdf worlds - and avoiding the xslt pilgrimage. In: Proc. of the 5th European Sem. Web Conf. ESWC'08, Springer (2008) 432–447
28. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: Sawsdl: Semantic annotations for wsdl and xml schema. IEEE Internet Computing **11** (November 2007) 60–67
29. Kiefer, C., Bernstein, A.: The creation and evaluation of isparql strategies for matchmaking. In: Proc. of the 5th European Sem. Web Conf. ESWC'08, Springer (2008) 463–477
30. Klusch, M., Kaufer, F.: Wsmo-mx: A hybrid semantic web service matchmaker. Web Intelli. and Agent Sys. **7** (January 2009) 23–42
31. Kiefer, C., Bernstein, A., Lee, H.J., Klein, M., Stocker, M.: Semantic process retrieval with isparql. In: Proc. of the 4th E. Conf. on the Sem. Web. ESWC '07, Springer (2007) 609–623
32. Dijkman, R.M., Dumas, M., van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Inf. Syst. **36**(2) (2011) 498–516
33. Bagheri, E., Asadi, M., Gašević, D., Soltani, S.: Stratified analytic hierarchy process: Prioritization and selection of software features. In: The 14th Int. SPL Conf., Springer (2010)
34. Batory, D., Sarvela, J.N., Rauschmayer, A.: Scaling step-wise refinement. In: Proc. of the 25th ICSE. ICSE '03, IEEE Computer (2003) 187–197
35. Apel, S., Lengauer, C., Möller, B., Kästner, C.: An algebraic foundation for automatic feature-based program synthesis. Sci. Comp. Prog. **75**(11) (2010) 1022–1047
36. van Ommering, R., Bosch, J.: Widening the scope of software product lines from variation to composition. In: Software Product Lines. Volume 2379 of LNCS. Springer (2002) 31–52
37. van Ommering, R.: Building product populations with software components. In: Proc. of the 24th ICSE Conf. ICSE '02, ACM (2002) 255–265
38. van Ommering, R.: Software reuse in product populations. IEEE Transactions on Software Engineering **31** (July 2005) 537–550
39. Bosch, J., Bosch-Sijtsema, P.: From integration to composition: On the impact of software product lines, global development and ecosystems. J. of Sys. and Softw. **83**(1) (2010) 67–76

SADI for GMOD: Semantic Web Services for Model Organism Databases

Ben Vandervalk[13], Michel Dumontier[2], E Luke McCarthy[1], and Mark D Wilkinson[1]

[1] James Hogg Research Centre, Heart + Lung Institute, University of British Columbia
[2] Department of Biology, Carleton University
[3] ben.vvalk@gmail.com

Abstract. Here we describe work-in-progress on the SADI for GMOD project (SADI: Semantic Automated Discovery and Integration; GMOD: Generic Model Organism Database), a distribution of ready-made Web services that will bring additional model organism data onto the Semantic Web. SADI is a lightweight standard for implementing Web services that natively consume and generate RDF, while GMOD is a widely-used toolkit for building model organism databases (e.g. FlyBase, ParameciumDB). The SADI for GMOD services will provide a novel mechanism for analyzing data across GMOD sites, as well as other bioinformatics resources that publish their data using SADI.

Keywords: Semantic Web, Web services, SADI, GMOD, model organism databases, bioinformatics, sequence features

1 Introduction

One of the most pervasive problems in bioinformatics is the integration of data and software across research labs. While the prevailing method of sharing data is through centrally controlled repositories such as GenBank [6], manual curation of submissions imposes a bottleneck on the quantity and types of data that can be integrated. In addition, centralization also places limits on the types of visualization and analysis tools that can readily be used with the data.

One prominent example of a system for integrating distributed biological data is the Distributed Annotation System (DAS) [7]. A DAS server provides access to sequence annotations (also known as sequence features) via a RESTful [8] interface, and returns the annotations in a simple, standardized XML format. Client applications (e.g. genome browsers) that understand the DAS protocol and XML format are able to provide users with a unified view of sequence annotations from multiple sites. Nevertheless, DAS has its limitations. The XML datasets returned by DAS servers cannot be integrated without specialized software, and cannot be readily combined with other types of data (e.g. protein-protein interaction networks). In addition, the majority of bioinformatics analysis tools (e.g. BLAST) do not natively understand DAS, and thus they require specialized conversion scripts in order to process data from DAS servers.

In this paper we describe work-in-progress on SADI for GMOD, a collection of Semantic Web services that implement DAS-like functionality. The goal of SADI for GMOD is to provide a more general solution for federating sequence data that is compatible with the Semantic Web, and which facilitates automated integration with analysis software and other types of bioinformatics data. Toward this goal, we propose a standard model for representing sequence features in RDF/OWL. The services are implemented according to the SADI (Semantic Automated Discovery and Integration) standard, and are targeted toward maintainers of GMOD (Generic Model Organism Database) sites. Additional information about these two projects is provided in the following section.

2 Related Projects

SADI (Semantic Automated Discovery and Integration) SADI [1] is a lightweight standard for the implementation of Semantic Web services. Services adhering to the SADI recommendations natively consume and generate data in RDF form, and can be invoked by issuing an HTTP POST to the service URL with an input RDF document as the payload. One of the principal strengths of SADI is that there are no specialized protocols or messaging formats. The interfaces to each service – that is, the expected structure of the input and output RDF documents – are described by means of a provider-specified input OWL class and output OWL class, respectively. Further details about SADI are given in [1].

GMOD (Generic Model Organism Database) The GMOD project [2] is a popular collection of open source software which facilitates the construction of a model organism database and its associated website. The central component of GMOD is a database schema called Chado [3], which houses a variety of datatypes such as sequences, sequence features, controlled vocabularies, and gene expression data. Scripts are provided for creating and loading a Chado instance as a Postgres database.

3 Services

SADI for GMOD consists of five services which provide fundamental operations for accessing sequence feature data, as shown in Table 1. A sequence feature is an annotated region of a biological sequence (DNA, RNA, or amino acid) such as a gene, an exon, or a protein domain. Related features are accessible through a hierarchy of parent-child relationships, and the GMOD wiki provides a set of recommendations [3] indicating where particular feature types should be located in the hierarchy. For example, the GMOD conventions assert that a gene should be a child feature of a chromosome and that an mRNA transcript should be a child feature of a gene. The relationship connecting the parent and child feature will be either “has part” or “derives into”, depending on whether the features are spatially or temporally related. For instance, the relationship between a chromosome and a gene is “has part”, whereas the relationship between a gene and a transcript is “derives into”.

Table 1. A functional description of the five SADI services implemented by the SADI for GMOD project. The fundamental input/output datatypes are genomic coordinates, feature descriptions, and database identifiers; further details about the representation of these entities is given in the following section.

Service Name	Input	Relationship	Output
get_feature_info	a database identifier	is about	a feature description
get_features_overlapping_region	a set of genomic coordinates	overlaps	a collection of feature descriptions
get_sequence_for_region	a set of genomic coordinates	is represented by	a DNA, RNA, or amino acid sequence
get_child_features	a feature description	has part / derives into	a collection of feature descriptions
get_parent_features	a feature description	is part of / derives from	a collection of feature descriptions

4 Proposal for Modeling Sequence Features in RDF

The implementation of the SADI for GMOD services is relatively straightforward. The main point of interest is how the data is modeled in RDF/OWL. The entities that need to be modeled are feature descriptions, genomic coordinates, and database identifiers, as shown in Table 2.

In Listing 1, we show an example feature description for a tRNA gene in *Drosophila melanogaster*, encoded in TURTLE format. The principal ontology used for the encoding is SIO (Semantic Science Integrated Ontology) [4], which provides a large collection of properties for capturing mereological, temporal, and other types of relationships. In addition, features are typed using terms from the Sequence Ontology [5]. Some readers may initially balk at the apparent complexity and opacity of Listing 1; however, it is important to emphasize that the primary goal of the encoding is to facilitate automatic integration of data, whereas simplicity and human-readability are secondary considerations. There are several data modeling practices that, when understood, should help to clarify Listing 1:

1. **Distinct entities are always modeled as distinct nodes in the graph.** In non-RDF formats (e.g. relational databases), it is easy to conflate related entities. For example, the sequence of a chromosome and the chromosome itself are often thought of as the same entity. However, this is not precisely true; the sequence is an abstract string representation of one of the strands of the chromosome. In order to facilitate accurate and automated processing of the data, it is often helpful to make such distinctions explicit. In Listing 1, the tRNA gene has a ranged sequence position in relation to a sequence that represents the minus strand of a chromosome.

Table 2. The fundamental input/output datatypes of the SADI for GMOD services.

Entity	Components	Example
feature description	• a feature type • a set of genomic coordinates • one or more database identifiers	Lines 11..41 of Listing 1
genomic coordinates	• a start position • an end position • a reference sequence	Lines 17..23 of Listing 1
database identifier	• a identifier type • an identifier string	Lines 14..15 of Listing 1

2. **URIs are frequently opaque.** Ontologies providers (e.g. OBI, GO, SO) assign numeric URIs to classes and relationships in their ontologies for two reasons: i) the URIs can have labels in multiple languages, and ii) the labels can be updated without requiring updates to dependent datasets.
3. **Literals are modeled as typed resources.** It is simplest to represent literals in RDF as plain strings or numbers, with the type of the literal indicated by the XSD datatype (e.g. `xsd:float`). Here, literals are modeled as instances of a particular `rdf:type` (e.g. `range:StartPosition`), with the actual values being specified by the "has value" property (i.e. `SIO_000300`). This approach provides a more flexible typing mechanism and allows additional information such as provenance to be attached to the values.
4. **Database identifiers are modeled as typed string values.** In Listing 1, the feature URI `http://lsrn.org/FLYBASE:FBgn0011935` has an attached identifier with an `rdf:type` of `lsrn:FLYBASE_Identifier` and a value of "FBgn0011935". This may seem redundant, as the URI already acts as a unique identifier for the feature. We have adopted the practice of attaching typed, string-encoded database identifiers to URIs in order to address a common problem on the Semantic Web, namely the tendency of data providers to invent their own URI schemes. For example, the URI for UniProt protein P04637 is alternatively represented on the Semantic Web as `http://purl.uniprot.org/uniprot/P04637` (UniProt and LinkedLifeData), `http://bio2rdf.org/uniprot:P04637` (Bio2RDF and Linked Open Drug Data), and `http://lsrn.org/UniProt:P04637` (SADI). While the existence of multiple URIs for the same entity impedes data integration across sites, data providers often create their own URI schemes so that the URIs will resolve to datasets

or webpages on their own sites. We propose attaching database identifiers to URIs as shown here, so that equivalent URIs can automatically be reconciled across sites, while still allowing the URIs created by each provider to resolve to their own data.

Listing 1. Example RDF encoding for a tRNA gene in *Drosophila melanogaster*.

```
@prefix feature: <http://sadiframework.org/ontologies/GMOD/Feature.owl#> .
@prefix range: <http://sadiframework.org/ontologies/GMOD/RangedSequencePosition.owl#> .
@prefix strand: <http://sadiframework.org/ontologies/GMOD/Strand.owl#> .
@prefix FlyBase: <http://lsrn.org/FLYBASE:> .
@prefix GB: <http://lsrn.org/GB:> .
@prefix lsrn: <http://purl.oclc.org/SADI/LSRN/> .
@prefix sio: <http://semanticscience.org/resource/> .
@prefix so: <http://purl.org/obo/owl/SO#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

FlyBase:FBgn0011935
     a so:SO_0001272; # 'tRNA_gene'
     sio:SIO_000008 # 'has attribute'
        [ a lsrn:FLYBASE_Identifier;
          sio:SIO_000300 'FBgn0011935'^^xsd:string ]; # p = 'has value'
     sio:SIO_000008 # 'has attribute'
        [ a range:RangedSequencePosition;
          range:in_relation_to _:minus_strand;
          sio:SIO_000053 # 'has proper part'
            [ a range:StartPosition; sio:SIO_000300 2077634 ];
          sio:SIO_000053 # 'has proper part'
            [ a range:EndPosition; sio:SIO_000300 2077707 ]
        ] .

GB:AE013599 # chromosome arm '2R'
     a so:SO_0000105; # 'chromosome_arm'
     sio:SIO_000008 # 'has attribute'
        [ a lsrn:GB_Identifier;
          sio:SIO_000300 'AE013599'^^xsd:string ] . # p = 'has value'

_:plus_strand
    a sio:SIO_000030; # o = 'sequence'
    sio:SIO_000210 # 'represents'
        [ a strand:PlusStrand;
          sio:SIO_000093 GB:AE013599 ] . # p = 'is proper part of'

_:minus_strand
    a sio:SIO_000030; # o = 'sequence'
    sio:SIO_000210 # 'represents'
        [ a strand:MinusStrand;
          sio:SIO_000093 GB:AE013599 ] . # p = 'is proper part of'
```

5 Deploying the Services

The SADI for GMOD services are implemented as Perl CGI (Common Gateway Interface) scripts. There will be three main steps to deploy the services at a GMOD site:

1. **Set up a Bio::DB::SeqFeature::Store database.** For performance reasons, the services do not query a Chado database directly, but instead use a `Bio::DB::SeqFeature::Store` database which must be loaded separately

by the GMOD site maintainer. The most common scenario is to load the data from a set of GFF files into a mysql database; `Bio::DB::SeqFeature::Store` provides the `bp_seqfeature_load.pl` script for this purpose.
2. **Unpack the SADI for GMOD tarball in the cgi-bin directory.** The tarball will be unpacked into a `SADI` directory tree which will contain the Perl CGI scripts as well as the required Perl modules.
3. **Add database connection parameters to the SADI for GMOD configuration file.** The configuration file will be located in the `SADI` subdirectory of `cgi-bin`.

6 Conclusion

While the majority of existing biological Web services use XML for data exchange, SADI services use RDF/OWL in order to facilitate automatic integration of data across service providers. As such, the SADI for GMOD services will provide a novel tool for conducting analyses across model organism databases, as well as other biological data sources and tools that are published using SADI.

7 Acknowledgements

Initial development of SADI and SHARE has been funded by a special initiatives award from the Heart and Stroke Foundation of British Columbia and Yukon, with additional funding from Microsoft Research and an operating grant from the Canadian Institutes for Health Research (CIHR). In addition, core laboratory funding has been supplied by the National Sciences and Engineering Research Council of Canada (NSERC). Development of SADI for GMOD, as well as hundreds of other SADI services, has been funded by a grant from Canada's Advanced Research and Innovation Network (CANARIE).

References

1. Wilkinson, M.D., Vandervalk, B.P., McCarthy E.L.: SADI Semantic Web Services - cause you cant always GET what you want! Services Computing Conference (APSCC) 2009, 13-18 (2009)
2. GMOD homepage, http://gmod.org
3. Introduction to Chado, GMOD Wiki, http://gmod.org/wiki/Introduction_to_Chado
4. Semantic Science on Google Code, http://code.google.com/p/semanticscience/
5. Eilbeck, K., Lewis, S.E., Mungall, C.J., et al.: The Sequence Ontology: a tool for the unification of genome annotations. Genome Biology 6:5 (2005)
6. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., et al.: GenBank. Nucleic Acids Research 36, D25-D30 (2008)
7. Dowell, R.D., Jokerst, R.M., Day, A. and et al.: The Distributed Annotation System. BMC Bioinformatics 2:7 (2001)
8. Fielding, R.T.: Architectural styles and the design of network-based software architectures. University of California, Irvine (2000)

An Ontological Approach for Querying Distributed Heterogeneous Information Systems in Critical Operational Environments

Atif Khan, John A. Doucette, and Robin Cohen

David R. Cheriton School of Computer Science, University of Waterloo
{atif.khan,j3doucet,rcohen}@uwaterloo.ca

Abstract. In this paper, we propose a decision making framework suited for knowledge and time constrained operational environments. We draw our motivation from the observation that large knowledge repositories are distributed over heterogeneous information management systems. This makes it difficult for a user to aggregate and process all relevant information to make the best decision possible. Our proposed framework eliminates the need for local aggregation of distributed information by allowing the user to ask meaningful questions. We utilize semantic knowledge representation to share information and semantic reasoning to answer user queries. We look at an emergency healthcare scenario to demonstrate the feasibility of our approach. The framework is contrasted with conventional machine learning techniques and with existing work in semantic question answering. We also discuss theoretical and practical advantages over conventional techniques.

1 Introduction

As electronic information systems become mainstream, society's dependence upon them for knowledge acquisition has increased. Over the years, the sophistication of these systems has evolved, making them capable of not only storing large amounts of information in diverse formats, but also of reasoning about complex decisions. The increase in technological capabilities has revolutionized the syntactic interoperability of modern information systems, allowing for a heterogeneous mix of systems to exchange a wide spectrum of data in many different formats. The successful exchange of raw information is, however, only the first step towards solving the bigger *semantic challenge* of information exchange. This is analogous to the "ontology challenge" defined by [15].

In recent years a focused effort in the semantic web domain has resulted in technological advancements, providing sophisticated tools for intelligent knowledge representation, information processing and reasoning. Domain specific knowledge can be managed by utilizing a diverse set of ontological solutions, which capture key domain concepts and the relationships between them. Knowledge

regarding the domain can then be shared by publishing information in a domain specific ontology. A semantic reasoning engine can then be applied to a knowledge-base to answer complex user queries. The semantic reasoning process allows for enhanced knowledge discovery that may not be possible via consumption of the raw data alone. Latent relationships can be discovered by applying inference rules to the ontological knowledge-base.

Although the premise of the semantic web technology is sound in principle and the use of an ontology can significantly enhance how users consume and process information, practical implementation all but demands that the distributed heterogeneous knowledge be represented by local ontological representations [13]. Consequently, it is still difficult to share knowledge across diverse heterogeneous sources to answer specific questions. Furthermore, under adverse conditions (i.e. constraints on time, communication and/or knowledge), the usefulness of the aggregated data decreases sharply, since human agents are required to (manually) process and reason with the data.

For example, in a health care setting, a physician may need to consult various medical information systems in order to determine the best possible solution for a patient. Given ideal conditions, a physician will be able acquire and process information from various systems and make the ideal diagnosis. If the same scenario is now constrained by the available time, communication bandwidth and the skill level of the physician, the same quality of medical care may not be possible.

Motivated by this, we propose a framework where a user will pose questions directly (in natural language), rather than aggregate knowledge locally in an attempt to find the answer. The framework will

- Process the user query.
- Aggregate information from various sources.
- Create a semantic representation of the aggregated data.
- Process information using a semantic reasoner.

Each answer generated (in response to the user query) is backed up by a semantic proof. The semantic proof has the desirable property that it can be independently validated by any third party. Our approach does not require the exchange of large data-sets to make a decision, and consequently is more suitable for the above mentioned adverse scenarios.

2 Proposed Solution

We propose a framework for reliable information exchange between distributed heterogeneous parties, using semantic web technologies under constrained operational conditions. We observe that under normal circumstances, such an exchange can easily be accomplished using existing techniques. These techniques fail to be of practical use under adverse situations. For example, consider the following time and information constrained setting: A patient is in a critical life threatening situation, and is being treated by an emergency response (EMR)

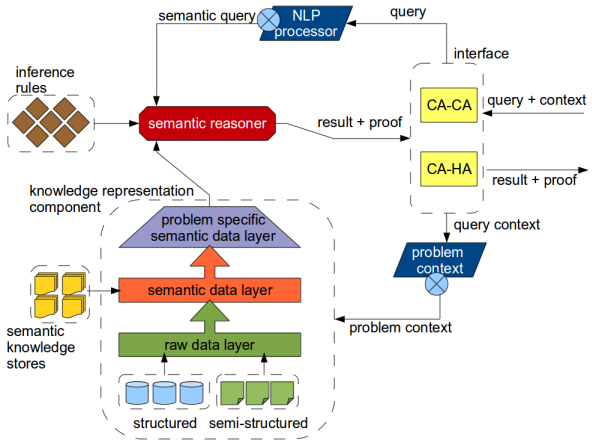


Fig. 1. System architecture

team member. Under these conditions the EMR team member may not be able to provide the best personalized care, because of difficulty accessing patient medical records in a timely manner or correctly interpreting those records.

Our proposed framework builds on top of the semantic web technologies. We use ontological models for knowledge representation. We acknowledge the fact that diverse heterogeneous information will be represented by an array of local or domain ontologies. Therefore, our framework provides support for working with multiple data-sets represented by different ontological models. Given the almost infinite amount of information in the world, we utilize a problem context to identify and limit the amount of knowledge that needs to be processed. We create a problem specific information model using this context. We utilize a semantic reasoner that takes as its input a knowledge-base, a set of inference rules and a user query. The reasoner generates a two part result-set, where the first element is the answer to the provided user query and the second element is a semantic proof.

We will now discuss the details of the various components of our proposed framework along with some examples.

2.1 System Architecture

We present a flexible architectural style for our proposed framework. Previous approaches utilizing similair frameworks tend to be domain specific (e.g. [17]). In contrast, our approach is domain indipendent. We now illustrate the salient components of our design (Refer to Fig 1).

System Interface The system interface component facilitates interaction by allowing a user to pose a query to the system. The user may also provide a

query specific context. We provide support for two types of user communications based on the following two user classifications (i) a computational agent (CA) – represents an artificially intelligent automated system and (ii) a human agent (HA) – representing a human being. The first type of agent communication is between two CAs. A local CA receives a query (and a context) from a remote CA. This type of communication represents distributed automated systems interacting with each other. The second type of communication utilizes a local CA and a remote HA. This allows human beings to pose queries to a local system. For each query, the interface receives a response from the reasoning module, and forwards this response to the remote user.

The system interface component provides a queryable abstraction around the heterogeneous knowledge stores, so that the actual data (utilized for answering the query) does not have to be transmitted. This characteristic of the framework facilitates knowledge sharing under adverse conditions.

Knowledge-Representation The knowledge-representation component of our framework follows a multi-tiered design that is capable of accepting data from a wide array of heterogeneous sources. It also utilizes the problem-context (generated from the user query context) to limit the amount of data which must be processed to answer the query.

The raw data layer provides a useful abstraction to deal with all non-semantic data sources. These data-sources are composed of structured data (such as in the case of distributed relational database systems) and semi-structured data (such as content repositories and web pages). We assume that this raw data does not have any semantic capabilities built into it.

Information from the raw data layer is then annotated using appropriate ontologies. This semantic data layer provides the appropriate abstraction. It is important to note that we do not constrain the choice of the ontologies used. The main goal here is to be able to convert raw data into its semantically equivalent representation. The semantic data layer is also capable of incorporating data from other semantic data repositories.

The problem-specific semantic layer provides a normalization of the semantic data layer. The main goal of this layer is to provide mappings between various ontological representations of the data in use. For example a single semantic concept (such as *name*) that may be defined by different ontologies can be normalized and represented by a concept from a single consistent ontology.

Reasoning and Inference The reasoning layer is responsible for processing the various inputs from other modules such as the semantic query (representing the initial user query), the inference rules, and the knowledge-base from the problem-specific semantic layer. It utilizes a semantic reasoner [27] to reason about the user query over the selected knowledge-base. The reasoner generates a two part result-set. The first element of the result-set contains the answer to the user query. The second element contains a semantic proof in support of the response.

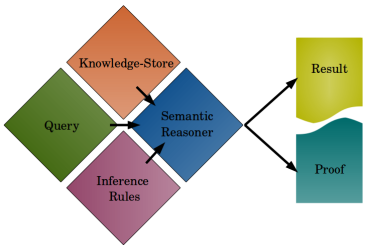


Fig. 2. Semantic Reasoning and inference

A semantic proof has the desirable property that it can be validated by any party. In a heterogeneous multi-agent distributed environment, knowledge changes with time. Therefore, the same query may not result in the same answer at a different time. Having a semantic proof generated for each user query allows the validation an answer against the knowledge-base representation (that was aggregated by the problem-specific semantic layer) at any given instant in time by any party.

Motivation In this section we consider two simple scenarios for knowledge sharing under adverse conditions, constrained by lack of time and lack of knowledge. The purpose of these examples is to highlight the various components of our proposed architecture and their interactions with one other. Fig 3 depicts a semantic model capturing the high level entities for a medical scenario. This semantic model represents the normalized view of the information gathered from various distributed sources. The model describes not only the entities, but also the semantic relationships between these entities.

The main entities defined in our model are patients, health care providers, drugs, diseases and various medical conditions. For the sake of simplicity, we define various simple relationships between these entities. The main relationship is the IS_A relationship (sometimes called "subsumption"). For example a doctor IS_A health care provider which IS_A person. Similarly Insulin IS_A allopathic drug which IS_A drug. In addition to the IS_A relationship, we also define several other varieties of attribute-value relationship. For example the disease Ulcer has a *condition* called Bleeding, the drug Nitroglycerin has a *contraindication* to the drug Viagra (Fig. 3). Using the triple notation [22] we capture the semantic model in a triple-store.

Example Scenario Consider a hypothetical scenario where an emergency response team member would like to administer Warfrain (an anticoagulant drug) to Alice in order to treat her for potential blood clotting. Alice is currently early in her pregnancy. The EMR member has had no past interactions with Alice, and is not aware of her medical condition and history. We add the following two constraints to this scenario to incorporate the (adverse) time and knowledge factors.

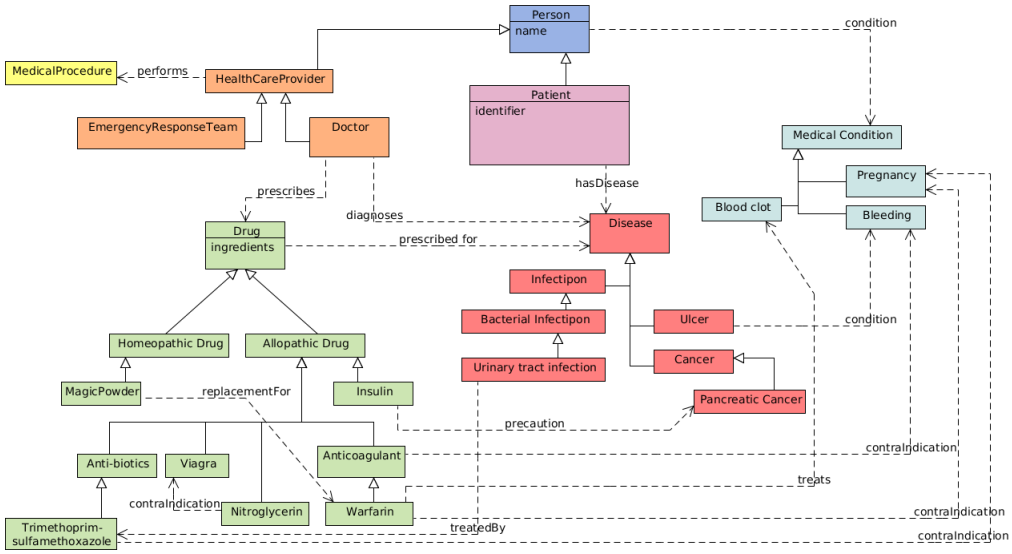


Fig. 3. Semantic model for medical question answering

- The current conditions prevent the EMR person from accessing and reviewing Alice's medical records.
- Alice's blood clot condition needs to be treated urgently.

Instead of aggregating information related to this scenario (such as Alice's medical records, drug interaction guidelines and such), the EMR person would launch a natural language query such as "can Alice be given Warfrain?" against a medical information system based on our framework. The system would identify Alice and Warfrain, and would compile the required information from various heterogeneous sources. The compiled knowledge is then translated into its' semantic representation. Fig. 4 shows a simplified contextual model based on the global knowledge store presented in Fig. 3. The semantic reasoner will consume this information along with the rules and semantic (user) query, and will generate a result and a proof as follows:

User Query

```
:Alice :canNotBeGiven :Warfrain.
```

Inference Rule

```
{?PATIENT :condition ?CONDITION.
  ?DRUG :contraIndication ?CONDITION. } => {?PATIENT :canNotBeGiven
?DRUG}.
```

Semantic Reasoning & Proof

{{:Alice :condition :Pregnancy} ***e:evidence*** <knowledge-base#_27>.
{:Warfrain :contraIndication :Pregnancy} ***e:evidence*** <knowledge-base#_22>}
=>
{ {:Alice :canNotBeGiven :Warfrain} ***e:evidence*** <rules#_9>}.
Proof found in 3 steps (2970 steps/sec) using 1 engine (18 triples) }.

Based on the facts and the inference rules, the semantic reasoner concludes that Alice can not be given Warfrain since she is pregnant and the Warfrain has a *contraindication* relationship with Pregnancy. The N3 representation of the user query, inference rules and semantic proof are shown above.

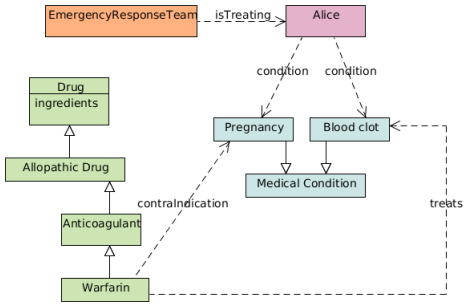


Fig. 4. Example Scenario: Should Alice be given Warfrain?

The scenario discussed above has been kept simple for ease of understanding. A more realistic knowledge-base would be quite rich in semantic concepts and a large number of relationships between the concepts. Similarly there will be a larger array of rules defined to provide the required level of inferencing capabilities to a complex semantic model.

3 Framework Realization

It is important to note that we are proposing a framework that can have many different realizations based on given system requirements. For example certain implementations can omit the natural language query interface if the interacting components are artificially intelligent machine agents. Similarly, different semantic reasoners can be used to achieve implementation goals of performance. Our proposed framework identifies the critical system components and their interactions.

Our realization of the system was solely focused on validating the proposed framework. As semantic knowledge representation and reasoning represent the most important components of the proposed framework, our proof-of-concept

realization was focused around the workings and validation of these components. We used N3 [22] notation to represent all knowledge (raw facts), inference rules and the system queries. Considering that N3 utilizes triple format to represent knowledge, any other representation capable of using the triple notation would be compatible with our approach.

We utilized the Euler proof mechanism [27] for semantic reasoning in our realization. Our choice was mainly driven by Euler's support for N3 notation and its support for the Java programming language (since our application was written in Java). The Euler project also provided an extensive set of examples where the OWL rules and concepts were already translated into N3 representation.

4 Related Work

In this section we establish both theoretical and practical concerns motivating the use of ontologies in question answering, and discuss previous work incorporating ontologies into knowledge querying.

4.1 Ontology-Free Approaches to Querying

There is considerable recent work suggesting that conventional querying techniques, though extremely powerful, might not be suitable for use in environments where queries are frequent, time-dependent, and arbitrarily complex. The principal reason for this is that, in the absence of semantic reasoning and inference rules, all information available for querying must be available in explicit form. This poses a problem in domains where there exists an enormous amount of information, precluding of the possibility explicit codification. For example, in the medical domain, there are hundreds of thousands of codified relationships between various concepts [28], but the hierarchical nature of these relationships means that the number of *implicit* relationships can be much higher. For example, viral pneumonia is explicitly defined as a type infectious pneumonia, but implicitly it is also a type of lung disease.

This motivates the use of machine learning techniques as a possible method of answering ad-hoc queries to an information system which may not encode all possible relationships. By taking a sufficiently large sample of the data, it may be possible to infer the answer to a user's query. For example, if a user asks whether a particular patient can be given a drug, a predictive classification system could be dynamically constructed and utilized to answer the query.

4.2 Motivations for Ontological Approaches to Querying

There are both theoretical and practical motivations for avoiding the use of traditional machine learning techniques to answer the kind of questions described above. Many machine learning algorithms, including popular decision trees (C4.5, ID3 [23]), maximum margin classifiers (e.g. Support Vector Machines

[7]), and clustering techniques (e.g. KNN [9]), operate by phrasing queries as optimization problems. For example, if a doctor wants to know whether their patient is likely to experience an adverse reaction to a drug, then a system might collect a large sample of patient records and use them to build a classification model. Although machine learning algorithms are often very effective in practice, there are theoretical reasons to suppose they might be less useful in time-critical domains where arbitrary queries are being made. "No Free Lunch" theorem (NFL) [30] shows that all optimization techniques are expected to produce identical mean performance across a set of arbitrary queries, in the absence of domain specific knowledge. This suggests that, over a large set of possible queries, no conventional machine learning technique is likely to answer all queries better than using completely random optimization strategies. In critical scenarios like ours, the possibility of receiving a poor result might be too large a risk for users to trust the system's answers.

There are also practical considerations, especially the opacity of the answers obtained using conventional query techniques. Continuing with our example above, what the doctor receives in response to a query about adverse reactions to a drug is a classification model based on a sample of patient data. The understandability of these models to computational laypeople varies from model to model. A support vector machine for example, is practically impossible for a layperson to understand, since it operates by building the maximally separating hyper-plane for a high-dimensional extrapolation of the given data. When the doctor asks "Why does the system believe my patient will have an adverse reaction?", she may not trust a system which answers "I put your patient's record into a 500 dimensional space, and it fell on this side of a line". This is true even if the system is highly reliable, because human users may have concerns about the ethics of entrusting life-saving decisions to a "black box". The system cannot easily explain its decision in terms of medical conditions and the relationships between them, and so it is impossible to tell whether the answer provided is based on sound reasoning, or an unfortunate hiccup in the algorithm's usual consistency.

4.3 Previous Ontological Approaches to Querying

There has been considerable previous work utilizing ontologies for answering queries, but the general focus is on preprocessing of queries to facilitate the use of conventional machine learning techniques. This is a reasonable approach insofar as it obviates the NFL issues described above by introducing domain specific knowledge into the optimization process. In medicine, for example, there has been a focus on isolating the queries used by doctors most frequently, and preprocessing them using semantic information[8,12]. By utilizing ontological information, previous researchers have created frameworks capable of automated contextualization of doctor queries. For example, a doctor whose patient has type I diabetes would have queries regarding that patient and "diabetes" automatically translated to instead include "Type I Diabetes" [21]. An alternative approach considers the incorporation of meta-data into search queries, which

can be utilized to return more relevant documents during information retrieval [11]. Finally, recent research in question answering systems utilizes ontologies to translate doctors' questions into lists of relevant terms for an ordinary search engine [14].

The use of a semantic reasoner in place of a conventional machine learning algorithm to answer search queries offers several immediate advantages. First, because a semantic reasoner does not rely on optimization to construct a predictive model, it is not subject to the problems posed by the No Free Lunch theorems for optimization. This eliminates the need for extensive incorporation of a priori knowledge by the end user, as in [11]. Second, the opacity problem is solved by the ability of the framework to both provide a proof of its answer (i.e. the chain of reasoning used to determine the answer), and to formulate that proof in terms easily understood by a layperson (i.e. via conversion of triple formated data into simple natural language statements). For example, if our doctor wishes to ask "Why does the system believe my patient will have a reaction to this drug?", instead of being told, somewhat tautologically, that their patient fits the system's model of patients who had reactions, the doctor can be provided with a patient-specific proof based on medical evidence. By providing a semantic proof, the framework asserts that answer to a user's query is correct, based on the present data.

5 Future Work

Future work will take two directions. First, we plan to implement and benchmark a prototype system, and compare its performance with that of a system based around conventional machine learning techniques for question answering. Second, we plan to extend the framework by overlaying probabilistic models onto the ontological model, to provide a more precise answer to a users' queries. For example, a user who reports cracks in a bridge might be told that there is a 60% chance of bridge failure, rather than simply being told that the bridge *will* collapse if they drive over it. A drawback associated with this extension is the "curse of dimensionality" which arises when there are many possible *combinations* of factors that have different interactions. For example, a bent bridge might have a 30% chance of collapsing, but a bent and cracked bridge a 99% chance of collapsing. The problem worsens as additional factors are added, and each combination of factors in turn must be considered.

To avoid this problem, we plan to consider the introduction of heuristic techniques for providing estimated probabilities. For example, we might have the system take a random sampling of past bridges with both characteristics, and produce an observed probability estimate. Alternatively, the framework could provide "reasonable" bounds in the absence of additional information by assuming no interaction and a positive interaction of strength proportionate to the criticality of the task. Thus, if the bridge is only 3ft off the ground, estimates of the risk would tend to be more liberal (i.e. smaller interaction estimates) than

if the bridge is 300ft off the ground. Neither scheme is ideal, and experimental validation might be required to determine appropriate estimates of risk.

6 Conclusion

In this paper we present a proposal for a general purpose ontology-based information exchange framework, intended for use in time critical, knowledge sparse scenarios. The framework utilizes ontologies to retrieve contextually relevant facts from external data sources; reason about those facts in the context of a problem-dependent rule base; and produce both answers and human readable proofs relevant to user queries.

The framework is demonstrated through two example scenarios with a prototype, and contrasted with existing work on semantic data mining (which tends to focus on pre- and post-processing, rather than rule discovery and query answering), and conventional, non-semantic machine learning approaches. Our framework eliminates the problems posed by the No Free Lunch theorem for optimization [30], and provides transparent answers which are easily understood by computational laypersons. Future work will focus on the implementation of a fully functional system, user studies of the system's effectiveness as compared with conventional techniques, and on incorporating probabilistic reasoning into the model.

Bibliography

1. Bettina Berendt, Andreas Hotho, and Gerd Stumme. Towards semantic web mining. In *In International Semantic Web Conference (ISWC*, pages 264–278. Springer, 2002.
2. Tim Berners-lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. N3logic: A logical framework for the world wide web. *Theory and Practice of Logic Programming*, 8(03):249–269, 2008.
3. Abraham Bernstein, Foster Provost, and Shawndra Hill. Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. *IEEE Transactions on Knowledge and Data Engineering*, 17:503–518, 2005.
4. Laurent Brisson and Martine Collard. How to semantically enhance a data mining process? In Will Aalst, John Mylopoulos, Norman M. Sadeh, Michael J. Shaw, Clemens Szyperski, Joaquim Filipe, and Jose Cordeiro, editors, *Enterprise Information Systems*, volume 19 of *Lecture Notes in Business Information Processing*, pages 103–116. Springer Berlin Heidelberg, 2009.
5. Silvana Castano and Alfio Ferrara. Knowledge representation and transformation in ontology-based data integration. In *Knowledge Transformation for the Semantic Web*, pages 107–121. 2003.

6. Harry Chen, Filip Perich, Dipanjan Chakraborty, Tim Finin, and Anupam Joshi. Intelligent agents meet semantic web in a smart meeting room. *Autonomous Agents and Multiagent Systems, International Joint Conference on*, 2:854–861, 2004.
7. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
8. Manning PR Covell DG, Uman GC. Information needs in office practice: are they being met? *Annals of Internal Medicine*, 103:596–9, 1985.
9. T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory J*, 13:21–27, 1967.
10. Randy D. Smith David W. Embley, Douglas M. Campbell. Ontology-based extraction and structuring of information from data-rich unstructrued documents. In *Proceedings of the 1998 ACM International Conference on Information and Knowledge Management, Bethesda, Maryland, USA*, 1998.
11. Dina Demner-Fushman and Jimmy Lin. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 2007.
12. JW Ely, JA Osheroff, MH Ebell, GR Bergus, BT Levy, ML Chambliss, and ER Evans. Analysis of questions asked by family doctors regarding patient care. *British Medical Journal*, 319:358–361, 1999.
13. James Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 16:30–37, 2001.
14. Pierre Jacquemart and Pierre Zweigenbaum. Knowledge and reasoning for medical question-answering. In *Proceedings of the 2009 Workshop on Knowledge and Reasoning for Answering Questions*, 2009.
15. Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, January 1998.
16. Igor Jurisica, John Mylopoulos, and Eric Yu. Using ontologies for knowledge management: An information systems perspective. In *In Proceedings of the 62nd Annual Meeting of the American Society for Information Science (ASISÅ 99), Oct. 31 - Nov*, pages 482–496, 1999.
17. Pavandeep Kataria and Radmila Juric. Sharing e-health information through ontological layering. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, 2010.
18. Haishan Liu. Towards semantic data mining. In *9th International Semantic Web Conference (ISWC2010)*, November 2010.
19. Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. Mafra - a mapping framework for distributed ontologies. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, pages 235–250, London, UK, 2002. Springer-Verlag.
20. Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16:72–79, 2001.
21. Eneida A. Mendonca, James J. Cimino, Stephen B. Johnson, and Yoon-Ho Seol. Accessing heterogeneous sources of evidence to answer clinical questions. *Journal of Biomedical Informatics*, 34:85–98, 2001.
22. Notation 3 (n3): A readable RDF syntax. http://www.w3.org/DesignIssues/Notation3.
23. JR Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers, 1993.
24. Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001. 10.1007/s007780100057.

25. Sarunas Raudys and Alvydas Pumputis. Group interests of agents functioning in changing environments. In *Multi-Agent Systems and Applications IV*, volume 3690 of *Lecture Notes in Computer Science*, pages 559–563. Springer Berlin / Heidelberg, 2005.
26. Å arunas Raudys. Survival of intelligent agents in changing environments. In *Artificial Intelligence and Soft Computing - ICAISC 2004*, volume 3070 of *Lecture Notes in Computer Science*, pages 109–117. Springer Berlin / Heidelberg, 2004.
27. Jos De Roo. Euler proof mechanism. Website.
28. Kent Spackman. Snomed-ct overview nehta presentation. http://www.nehta.gov.au/component/docman/doc_details/589-snomed-ct-overview-nehta-presentation-august-2008.
29. Andrew B. Williams. Learning to share meaning in a multi-agent system. *Autonomous Agents and Multi-Agent Systems*, 8:165–193, March 2004.
30. D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997.
31. D.H. Wolpert and W.G. Macready. Coevolutionary free lunches. *IEEE Transactions on Evolutionary Computation*, 9(6):721 – 735, 2005.
32. Youyong Zou, Tim Finin, Li Ding, Harry Chen, and Rong Pan. Using Semantic web technology in Multi-Agent systems: a case study in the TAGA Trading agent environment. In *Proceeding of the 5th International Conference on Electronic Commerce*, pages 95–101, September 2003.