

Social Semantic Rule Sharing and Querying in Wellness Communities

Harold Boley, Taylor Michael Osmun, and Benjamin Larry Craig

Institute for Information Technology
National Research Council of Canada
Fredericton, NB, E3B 9W4, Canada

Abstract. In this paper we describe the Web 3.0 case study WellnessRules, where ontology-structured rules (including facts) about wellness opportunities are created by participants in rule languages such as Prolog and N3, and translated for interchange within a wellness community using RuleML/XML. The wellness rules are centered around participants, as profiles, encoding knowledge about their activities, nutrition, etc. conditional on the season, the time-of-day, the weather, etc. This distributed knowledge base extends fact-only FOAF profiles with a vocabulary and rules about wellness group networking. The communication between participants is organized through Rule Responder, permitting translator-based reuse of wellness profiles and their distributed querying across engines. WellnessRules interoperates between rules and queries in the relational (Datalog) paradigm of the pure-Prolog subset of POSL and in the frame (F-logic) paradigm of N3. These derivation rule languages are implemented in the engines OO jDREW and Euler, and connected via Rule Responder to support wellness communities. An evaluation of Rule Responder instantiated for WellnessRules found acceptable Web response times.

1 Introduction

Medicine 2.0 [Eys08], Health 2.0 [MO09], and Wellness 2.0 have emerged as interconnected application areas of Web 2.0 (Social Web) techniques. Web 2.0 combined with Semantic Web techniques is currently leading to Web 3.0 (Social Semantic Web) techniques, which can also be applied in these areas. As part of NRC-IIT's Health & Wellness and Learning & Training efforts, we are exploring Wellness 3.0, employing Social Semantic Web rules plus ontologies to plan wellness-oriented activities and nutrition.

We focus here on WellnessRules¹, a system supporting the management of wellness practices within a community based on rules plus ontologies. The idea is the following. As in Friend of a Friend (FOAF)², people can choose a (community-unique) nickname and create semantic profiles about themselves, here about their wellness practices, for their own planning and to network with

¹ <http://ruleml.org/WellnessRules/>

² <http://www.foaf-project.org/>

other people supported by a system that ‘understands’ those profiles. As in FindXpRT [LBBM06], such FOAF-like fact-only profiles are extended with rules to capture conditional person-centered knowledge such as each person’s wellness activity depending on the season, the time-of-day, the weather, etc. People can use rules of various refinement levels and rule languages ranging from pure Prolog to N3, which will be interoperated through RuleML/XML [Bol07]. Like our (RuleML-20xy) SymposiumPlanner [CB08] (and unlike FindXpRT), WellnessRules is based on Rule Responder [PBKC07,CB08], which is itself based on the Mule Enterprise Service Bus (ESB).

We will discuss an example where John (p0001) advertises Prolog-style rules on his wellness community profile, including a refinement of the following: p0001 may do outdoor running if it is summer and not raining. Hence, Peter and Paul can find p0001 via Prolog or N3 queries to Rule Responder expressing their own preferences, so that an initial group might be formed. Interoperating with translators, WellnessRules thus frees participants from using any single rule language. In particular, it bridges between Prolog as the main Logic Programming rule paradigm and N3 as the main Semantic Web rule paradigm.

The distributed nature of Rule Responder profiles, each queried by its own (copy of an) engine, permits scalable knowledge representation and processing. Since participants of a wellness community are supposed to meet in overlapping groups for real-world events such as skating, this kind of community (unlike a virtual-only community) has a maximal effective size (which we estimate to be less than 1000 participants). Beyond that size, it can be split into two or more subcommunities based on preferred wellness practices, personal compatibility, geographic proximity, etc. Rule Responder support thus needs to extend only to that maximal size, but can be cloned as subcommunities emerge.

The rest of the paper is organized as follows. Section 2 details the design goals with a focus on Rule Responder. Section 3 discusses the hybrid global knowledge bases of WellnessRules. Section 4 explains its local knowledge bases distributed via Rule Responder. Section 5 focusses on the interoperation between Prolog and N3. Section 6 explains and evaluates Rule Responder querying of WellnessRules knowledge. Section 7 concludes the paper. Appendix A contains two rule signatures of WellnessRules. Appendix B shows its RDFS taxonomy.

2 Design Goals and Rule Responder Instantiation

A range of design goals is being pursued with the WellnessRules prototype:

1. Identify a language of appropriate expressiveness for wellness rules in the layered family of RuleML [Bol07], RIF [BK09], and N3 [BLCK⁺08] such as unary/binary vs. n-ary Datalog and rules with or without negation as failure (Naf), with or without slots, objects, etc.
2. Identify a language for wellness ontologies in the layered family of OWL, OWL 2, and RDFS such as `subclassOf` taxonomies, description logics from ALC to SHOIN, etc. The ontology language should be combined with the rule language.

3. Permit rule plus ontology authoring in different human-oriented syntaxes, while translating the rulebases plus ontologies to and fro XML for interchange.
4. Permit different rule plus ontology engines such as OO jDREW, Prova, Euler, Racer, and Pellet to execute local rulebases plus ontologies, while transporting XML rulebases, ontologies, queries, and answers over an ESB.
5. Create profile examples and templates of typical wellness practitioners from which new participants can gather ideas and copy & edit for their own profiles.
6. Create an infrastructure to support wellness communities in the management of their existing practices, the adoption of new practices from participants, and the formation of ad hoc groups through profile querying.
7. Explore the appropriateness of the rule plus ontology languages and engines for expressing and translating the knowledge required in wellness profiles.
8. Explore the efficiency and networkability of the distributed WellnessRules architecture, including its ESB and engines, w.r.t. the different languages employed.

The current version of WellnessRules focusses goal 1 on Datalog with Naf and on N3 [BLCK⁺08] with scoped Naf, interoperating between these rule paradigms through RuleML/XML (a Naf Core is not currently planned for RIF [BK09]). It focusses goal 2 on light-weight ontologies, namely a sorted Datalog and an N3 with RDFS `subClassOf` taxonomies, for both using sorted (or typed) variables in RuleML/XML. Accordingly, we focuss goal 3 on POSL [Bol04] and N3, and goal 4 on their engines OO jDREW³ and Euler⁴, respectively. Goal 5 has been a joint effort by the wellness community at NRC-IIT Fredericton.

Goal 6 instantiates the Rule Responder multi-agent architecture as follows: Rule Responder's virtual organization is instantiated to a wellness community. An organizational agent (OA) becomes an assistant for an entire wellness community. Each personal agent (PA) becomes an assistant for one participant. Newcomers and participants can assume the role of an external agent (EA), (indirectly) querying participants' profiles.

Rule Responder uses the following sequence of steps: An EA asks queries to an OA. The OA maps and delegates each query to the PA(s) most knowledgeable about it. Each PA poses the query to its local rulebase plus ontology, sending the derived answer(s) back to the OA. The OA integrates relevant answers and gives the overall answer(s) to the EA, by default not revealing the coordinates of the answering PA(s).

In this way, the OA acts as a mediator that protects the privacy of profiles of participants in a wellness community. Participants within the same community can of course later decide to reveal their real name and open up their wellness profiles for (direct) querying by selected other participants.

³ <http://www.jdrew.org/ojdrew/>

⁴ <http://eulerssharp.sourceforge.net/>

The above Rule Responder steps have been instantiated earlier, including to the SymposiumPlanner system [CB08]. The new instantiation for the WellnessRules system relates to this in interesting ways:

Like SymposiumPlanner’s OA stands for a specific symposium such as RuleML-2009, WellnessRules’ OA stands for a specific wellness community such as the one at NRC-IIT Fredericton. Like SymposiumPlanner’s PAs formalize knowledge to assist (publicity, panel, etc.) co-chairs within a symposium, WellnessRules’ PAs formalize knowledge to assist (running, skating, etc.) co-organizers within a wellness community.

Unlike in SymposiumPlanner, where, besides the co-chairs, regular (symposium) participants *are not* supported by PAs, in WellnessRules, besides the co-organizers, regular (wellness) participants *are* supported by PAs: Each WellnessRules participant publishes a more or less specific profile used by his/her PA to respond to queries about his/her wellness preferences, constraints, etc.

Goal 7, the expressiveness exploration of WellnessRules, so far found that both the Prolog paradigm of top-down n-ary relational rules and the N3 paradigm of bottom-up binary frame rules can express the required Web 3.0 knowledge, although more directly and compactly in pure Prolog and more closely to Semantic Web standards in N3. Goal 8, the efficiency exploration, so far found that Rule Responder instantiated for WellnessRules has acceptable Web efficiency and provides a good balance between knowledge distribution and centralization.

3 Hybrid Global Knowledge Bases in WellnessRules

WellnessRules employs a hybrid combination [Bol07] of ontologies and rules. While the entire ontology and a portion of the rulebase is globally shared by all participants (agents), the other portion of the rulebase is locally distributed over the participants (agents).

As its (light-weight) ontology component, WellnessRules employs `subClassOf` taxonomies. We reuse parts of the Nuadu ontology collection [SLKL07], especially from the Activity and Nutrition ontologies. WellnessRules currently employs an `Activity` taxonomy using Nuadu classes (based on activity codes [AHW⁺00]) `Running`, `Walking`, `WaterSports` subsuming `SwimmingCalm`, `WinterSports` subsuming `IceSkating`, and `Sports` subsuming a WellnessRules class `Baseball`, as well as WellnessRules classes `Hiking`, and `Yoga`. Appendix B contains the corresponding RuleML- and N3-readable RDFS `subClassOf` statements.

As its rule component, WellnessRules employs Naf Datalog POSL and N3 with scoped Naf. We restrict the use of Naf Datalog POSL to atoms with positional arguments, leaving F-logic-like frames with property-value slots to N3, thus demonstrating the range of our approach through complementary rule styles. For that reason, the POSL syntax corresponds to pure-Prolog syntax except that POSL variables are prefixed by a question mark while Prolog variables are upper-cased.

This Datalog POSL sublanguage uses (positional) n-ary relations (or, predicates) as its central modeling paradigm. N3 instead uses (unordered) sets of

binary relations (or, properties) centered around object identifiers (OIDs, in the role of ‘subjects’ in N3). For example, in POSL we use the 4-ary predicate `meetup` with a positional signature: `meetup(?MapID,?Activity,?Ambience,?Location)`. In N3 this becomes a slotted signature with subject `_:meetup`, an `rdf:type` of `:Meetup`, and the 4 arguments as the remaining slots:

```
_:meetup
  rdf:type      :Meetup;
  :mapID        ?MapID;
  :activity     ?Activity;
  :inOut        ?Ambience;
  :location     ?Location.
```

Generally, as a foundation for our interchange study, the signatures of all WellnessRules relations are described in a Datalog-N3-generalizing manner, using the same mnemonic names for predicate arguments in Datalog and properties in N3. The WellnessRules signatures needed here are given in appendix A.

Shared rules (including underlying facts) of the WellnessRules system were extracted and collected for the rulebase of the OA. They formalize global knowledge of the WellnessRules system.

For example, this is a global `meetup` fact according to the above positional signature: `meetup(m0001,walk,out,conniesStation)`. Similarly, this is its slotted counterpart:

```
:meetup_1
  rdf:type      :Meetup;
  :mapID        :m0001;
  :activity     :run;
  :inOut        :out;
  :location     :conniesStation.
```

Both express that one `meetup` for `walk` activities of the supported wellness community is `conniesStation` as found on map `m0001`.

An example of a global POSL rule defines a `participation` as follows:

```
participation(?ProfileID,?Activity,?Ambience,?MinRSVP,?MaxRSVP) :-
  groupSize(?ProfileID,?Activity,?Ambience,?Min,?Max),
  greaterThanOrEqualTo(?MinRSVP,?Min),
  lessThanOrEqualTo(?MaxRSVP,?Max).
```

As in FindXpRT, the first argument of a WellnessRules conclusion predicate always is the person the rule is about. Similar to Prolog, the rule succeeds for its five positional arguments if the acceptable `groupSize` of the participant with `?ProfileID`, for an `?Activity` in an `?Ambience`, is between `?Min` and `?Max`, and the emerging group has size between `?MinRSVP ≥ ?Min` and `?MaxRSVP ≤ ?Max`, where `greaterThanOrEqualTo` and `lessThanOrEqualTo` are SWRL built-ins as implemented in OO jDREW 0.961.

The corresponding global N3 rule represent this in frame form as follows:

```

{
  ?rsvpQuery
    rdf:type      :RSVPQuery;
    :profileID    :p0001;
    :minRSVP      ?MinRSVP;
    :maxRSVP      ?MaxRSVP.

  ?groupSize
    rdf:type      :GroupSize;
    :profileID    ?ProfileID;
    :activity     ?Activity;
    :inOut        ?Ambience;
    :min          ?Min;
    :max          ?Max.

  ?MinRSVP math:notLessThan ?Min.

  ?MaxRSVP math:notGreaterThan ?Max.
}
=>
{
  _:participation
    rdf:type      :Participation;
    :profileID    :p0001;
    :activity     ?Activity;
    :inOut        ?Ambience;
    :min          ?MinRSVP;
    :max          ?MaxRSVP.
}.

```

Here, the first premise passes the input arguments `?MinRSVP` and `?MaxRSVP` into the rule (cf. its use in section 6). The remaining premises correspond to those in the POSL version, where `math:notLessThan` and `math:notGreaterThan` are N3 built-ins as implemented in Euler.

The global OA rulebase is being maintained in both languages at <http://ruleml.org/WellnessRules/WR-Global.posl> and `*.n3`.

4 Locally Distributed Knowledge Bases in WellnessRules

Each PA has its own local rules, which were selected from profiles created by participants of the NRC-IIT Fredericton wellness community.

This is an example of a local POSL rule from the PA rulebase of a participant `p0001`, defining the main predicate `myActivity` for running:

```

myActivity(p0001,?:Running,out,?MinRSVP,?MaxRSVP,?StartTime,?EndTime,
          ?Place,?Duration,?Level) :-
  calendar(p0001,?Calendar),
  event(?Calendar,?:Running,possible,?StartTime,?EndTime),
  participation(p0001,run,out,?MinRSVP,?MaxRSVP),

```

```

season(?StartTime,summer),
forecast(?StartTime,sky,?Weather),
notEqual(?Weather,raining),
map(p0001,?Map),
meetup(?Map,run,out,?Place),
level(p0001,run,out,?Place,?Duration,?Level),
fitness(p0001,?StartTime,?ExpectedFitness),
greaterThanOrEqual(?ExpectedFitness,?Level),
goodDuration(?Duration,?StartTime,?EndTime).

```

The rule conclusion starts with the person's profile ID, p0001, followed by the kind of activity, run, and its ambience, outdoors, followed by variables for the group limits ?MinRSVP and ?MaxRSVP, the earliest ?StartTime and ?EndTime, its actual ?Duration and its ?Level. The rule premises query p0001's ?Calendar, an event of a possible (or tentative) ?Running (the anonymous variable "?" has type Running), the participation (see above), an appropriate season and forecast, p0001's ?Map, a meetup ?Place, the required level less than the expected fitness, as well as a goodDuration.

The corresponding local N3 rule is given abridged below (complete, online at <http://ruleml.org/WellnessRules/PA/p0001.n3>):

```

{
...

?forecast
  rdf:type      :Forecast;
  :startTime    ?StartTime;
  :aspect       :sky;
  :value        ?Weather.

?Weather log:notEqualTo :raining.

...
}
=>
{
  _:myActivity
    rdf:type      :MyActivity;
    :profileID    :p0001;
    :activity     :Running;
    :inOut        :out;
    :minRSVP      ?MinRSVP;
    :maxRSVP      ?MaxRSVP;
    :startTime    ?StartTime;
    :endTime      ?EndTime;
    :location     ?Place;
    :duration     ?Duration;
    :fitnessLevel ?FitnessLevel.
}.

```

The online version of the above POSL rule employs the premise `naf(forecast(?StartTime,sky,raining))` instead of separate `forecast` and `notEqual` premises. For the N3 online version, the above `log:notEqualTo` built-in call is more convenient. An irreducible Naf used in POSL’s online version adds the following premises in the `myActivity` rule (after the current `event` premise):

```
yesterday(?StartTime,?StartTimeYtrday,?EndTime,?EndTimeYtrday),
naf(event(?Calendar,?:Running,past,?StartTimeYtrday,?EndTimeYtrday)),
```

It makes sure that `p0001`’s calendar does not contain a running event on the day before. The counterpart in N3 could use `log:notIncludes`, which in Euler, as in our online version, is replaced with `e:findall`, checking that the result is the empty list, `()`.

The resulting PA rulebases, which require Datalog with Naf and N3 with `()-e:findall`, are being maintained in both of these languages at <http://ruleml.org/WellnessRules/PA>, e.g. those for `p0001` at <http://ruleml.org/WellnessRules/PA/p0001.posl> and `*.n3`.

5 Cross-Paradigm Rulebase Alignment and Translation

The WellnessRules case study includes a testbed for the interoperability (i.e., alignment and translation) of rulebases in the main two rule paradigms: Prolog-style (positional) relations and N3-style (slotted) frames. In our interoperability methodology, we make 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 leads to better translations, etc. Using this methodology for WellnessRules, we are maintaining a relational as well as a frame version of the rules, both accessing the same, independently maintained, RDFS ontology.

For rulebase alignment, the signatures of WellnessRules relations and frames are aligned in a shared signature document (cf. appendix A), discussed in section 3, which includes the slot names for frames.

For rulebase translation, we first use a pair of online translators (<http://ruleml.org/posl/converter.jnlp>) between the human-oriented syntax POSL and its XML serialization in OO RuleML. Based on the RDF-RIF combinations in [dB09], similar translators are being developed between N3, RIF, and RuleML.

The interoperability between WellnessRules PAs that use different rule paradigms is then enabled by RuleML, which has sublanguages for both the relational and the frame paradigms, so that the cross-paradigm translations can use the common XML syntax of RuleML.

The alignment of sample relations and frames in sections 3 and 4 suggests translations between both paradigms. We consider here translations that are ‘static’ or ‘at compile-time’ in that they take an entire rulebase as input and return its entire transformed version. We can thus make a ‘closed-arguments’ assumption of fixed signatures for relations and frames. In particular, the arity of relations cannot change at run-time and no slots can be dynamically added

or removed from a frame. The translations work in both directions:

Objectify: Mapping from an n-ary relation `rel` to a frame, this constructs a new frame with a generated fresh OID `rel_j`, where $j > 0$ is the first integer making `rel_j` a unique name, and with the argument positions `p_1`, `p_2`, ..., `p_n` as slot (or property) names.

Positionalize: Mapping from a frame to a relation, this constructs a new relationship with the first argument taking the frame OID and the remaining arguments taking the slot values of the sorted slot names from all frames of OID's class (null values for those properties not used in the current frame).

Formally, positionalizing is specified as follows, using POSL's frame notation with slots arrows (\rightarrow) as in F-logic and an OID separated from its slots by a hat ($\hat{}$).

1. Unite all slot names from all frames whose OID is an instance of a class `cl` into a finite set SN_{cl} of n_{cl} elements.
2. Introduce $(SN_{cl}, <)$ as a total order ' $<$ ' over SN_{cl} , where ' $<$ ' usually is the lexicographic order. Assume without loss of generality that the elements of SN_{cl} are $prop_1 < \dots < prop_{n_{cl}}$.
3. For each frame $frel = cl(\hat{oid} \text{ prop}_{k_1} \rightarrow TERM_{k_1}; \dots; \text{prop}_{k_m} \rightarrow TERM_{k_m})$ assume without loss of generality that $prop_{k_1} < \dots < prop_{k_m}$. Replace `frel` by a relation $frel' = cl(\hat{oid}, TERMCOMP_1, \dots, TERMCOMP_{n_{cl}})$, where for $1 \leq i \leq n_{cl}$ and $1 \leq j \leq m$ we have $TERMCOMP_i = TERM_{k_j}$ if $i = k_j$ and $TERMCOMP_i = \perp$ otherwise (' \perp ' is the null value formalized as the bottom element of the taxonomy, e.g. owl:NOTHING, which is equal only to itself, not to any other sort, constant, or variable).

Step 3 can be thought of as 'replenishing' the lexicographically sorted slots of a frame `frel` with slots $prop_x \rightarrow \perp$ for all slot names $prop_x$ 'missing' for their class `cl`, and then making `cl` the relation name, inserting the `oid`, and omitting all slot names (keeping only their slot values).

An XSLT implementation of such a translator is available online (<http://ruleml.org/ooruleml-xslt/oo2prml.html>).

For example, the POSL `meetup` fact of section 3 is serialized in positional RuleML thus:

```
<Atom>
  <Rel>meetup</Rel>
  <Ind>m0001</Ind>
  <Ind>walk</Ind>
  <Ind>out</Ind>
  <Ind>conniesStation</Ind>
</Atom>
```

Since RuleML allows intermediate forms, a translation between such a relation and a frame need not insert the relationship OID as a new first positional argument but can directly introduce it as a role tag:

```

<Atom>
  <oid><Ind iri=":meetup_1"/></oid>
  <Rel>meetup</Rel>
  . . .
</Atom>

```

Extending the mappings at <http://ruleml.org/indoo/n3ruleml.html>, this is an intermediate step towards the following frame in OO RuleML, where the RuleML Rel represents the `rdf:type`:

```

<Atom>
  <oid><Ind iri=":meetup_1"/></oid>
  <Rel iri=":Meetup"/>
  <slot>
    <Ind iri=":mapID"/>
    <Ind iri=":m0001"/>
  </slot>
  <slot>
    <Ind iri=":activity"/>
    <Data>walk</Data>
  </slot>
  <slot>
    <Ind iri=":inOut"/>
    <Data>out</Data>
  </slot>
  <slot>
    <Ind iri=":location"/>
    <Data>conniesStation</Data>
  </slot>
</Atom>

```

Practical objectifying looks up the slot names in the shared signature document (cf. appendix A).

For the translation of a rule, the above relation-frame translation is applied to the relation (frame) in the conclusion and to all the relations (frames) 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 `myActivity` rule in section 4 can thus be translated via Prolog, POSL, RuleML (relations, frames), and N3. These translators permit rule and query interoperation, via RuleML/XML, for the Rule Responder infrastructure of WellnessRules communities.

6 Distributed Rule Responder Querying of WellnessRules

On the basis of Rule Responder (cf. section 2) the knowledge bases of sections 3 and 4 can be queried, using the translators of section 5 for interoperation. The implemented Rule Responder for WellnessRules is available for online use at <http://ruleml.org/WellnessRules/RuleResponder/>.

For example, this is a POSL query regarding `p0001`'s wellness profile, for execution by a top-down engine such as OO jDREW TD:

```
myActivity(p0001,?:Running,out,1:Integer,20:Integer,"2009-06-10T10:15:00",
          "2009-06-10T11:15:00",?Place,?Duration,?Level)
```

It uses the rule from sections 4 to check whether p0001 will possibly be `?:Running`, outdoors, in a group of `1:Integer` to `20:Integer`, between start time "2009-06-10T10:15:00" and end time "2009-06-10T11:15:00". Using further rules and facts from p0001's profile (<http://ruleml.org/WellnessRules/PA/p0001.posl>), it produces multiple solutions, binding the meetup `?Place`, the `?Duration`, and the required fitness `?Level`.

The corresponding N3 query for execution by a bottom-up engine such as EulerSharp EYE uses a temporary fact to pass the input arguments:

```
:rsvpQuery
  rdf:type      :RSVPQuery;
  :profileID    :p0001;
  :minRSVP      1;
  :maxRSVP      20.
```

The N3 query itself then is as follows:

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

```
_:myActivity
  rdf:type      :MyActivity;
  :profileID    :p0001;
  :activity     :Running;
  :inOut        :out;
  :minRSVP      ?MinRSVP;
  :maxRSVP      ?MaxRSVP;
  :startTime    "2009-06-10T10:15:00";
  :endTime      "2009-06-10T11:15:00";
  :location     ?Place;
  :duration     ?Duration;
  :fitnessLevel ?FitnessLevel.
```

After declaring two prefixes, it builds an existential '(_)' node, `_:myActivity`, using slots for the fixed parameters and the fact-provided `?MinRSVP` and `?MaxRSVP` bindings to create slots with the `?Place`, `?Duration`, and `?Level` bindings.

An evaluation of the response times of the Mule infrastructure and the Rule Responder engines (OO jDREW, Euler, and Prova) instantiated for WellnessRules has been conducted using the previously discussed scenario. We found that this Rule Responder instantiation operates with acceptable Web response times.

Specifically, the execution times for the above `myActivity` query in Euler (N3), OO jDREW (POSL), and Rule Responder on average were 157ms, 1483ms, and 5053ms, respectively, measured as the Java system time, running in Java JRE6, Windows XP Professional SP3, on an Intel Core 2 Duo 2.80GHz processor.

For this and similar WellnessRules queries, the major contribution to the overall execution time has come from the ESB (Mule), which is not the focus of this work. Rule Responder operates using a star-like connection architecture,

where the OA dispatches network traffic to and fro the most appropriate PA. A separate study in distributed querying has worked on minimizing this communication overhead.⁵

The above query could be specialized to produce exactly one solution, e.g. by changing the parameter `outdoors` to `indoors`. It would fail for `?MaxRSVP` greater than 20. Using such queries, WellnessRules participants can check out profiles of other participants to see if they can join an activity group.

7 Conclusion

The WellnessRules case study demonstrates FOAF-extending Web 3.0 profile interoperation between a pure Prolog subset (Datalog with Naf) and N3 through RuleML/XML. With all of its source documents available, it has become a major use case for exploring various aspects, including scalability, of (distributed) knowledge on the (Social Semantic) Web, starting with derivation rules and light-weight ontologies. While WellnessRules so far has probed the OO jDREW, Euler, and Prova engines, its open Rule Responder architecture will make it easy to bring in new engines.

WellnessRules currently emphasizes rulebase translation and querying. These constitute basic services that we intent to extend by superimposed update services, e.g. for changing calendar entries for activities from status `possible` to `planned`; this will require production rules. The next extension will be relevant for `performing` wellness events, which will call for event-condition-action rules. Both of these extended rule types are covered by Reaction RuleML [PKB07].

This case study will also provide challenges for RIF [BK09]: WellnessRules' current derivation rules, for RIF-BLD implementations; its planned production rules, for RIF-PRD updates & implementations; and its envisioned reaction rules, for the evolving RIF-RRD.

8 Acknowledgements

We thank the wellness community at NRC-IIT Fredericton for their advice & enthusiasm. Thanks also go to Jos de Roo for his help with the Euler engine. NSERC is thanked for its support through a Discovery Grant for Harold Boley.

References

- [AHW⁺00] B. Ainsworth, W. Haskell, M. Whitt, M. Irwin, A. Swartz, S. Strath, W. O'Brien, D. Bassett JR., K. Schmitz, P. Emplaincourt, D. Jacobs JR, and A. Leon. Compendium of Physical Activities: an update of activity codes and MET intensities. 2000.
- [BK09] H. Boley and M. Kifer. RIF Basic Logic Dialect. W3C Working Draft. <http://www.w3.org/2005/rules/wiki/BLD>, July 2009.

⁵ <http://ruleml.org/papers/EvalArchiRule.pdf>

- [BLCK⁺08] 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 (TPLP)*, 8(3), May 2008.
- [Bol04] Harold Boley. POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge. <http://www.ruleml.org/submission/ruleml-shortation.html>, May 2004.
- [Bol07] Harold 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.
- [CB08] Benjamin Larry Craig and Harold Boley. Personal Agents in the Rule Responder Architecture. In Nick Bassiliades, Guido Governatori, and Adrian Paschke, editors, *RuleML*, volume 5321 of *Lecture Notes in Computer Science*, pages 150–165. Springer, 2008.
- [dB09] Jos de Bruijn. RIF RDF and OWL Compatibility. W3C Working Draft. <http://www.w3.org/2005/rules/wiki/SWC>, July 2009.
- [Eys08] Gunther Eysenbach. Medicine 2.0: Social Networking, Collaboration, Participation, Apomediation, and Openness. Jul-Sep 2008.
- [LBBM06] Jie Li, Harold Boley, Virendrakumar C. Bhavsar, and Jing Mei. Expert Finding for eCollaboration Using FOAF with RuleML Rules. In *Montreal Conference of eTechnologies 2006*, pages 53–65, 2006.
- [MO09] Heather Molyneaux and Susan O’Donnell. Patient Portals 2.0: The Potential for Online Videos. In *COACH Conference: e-Health: Leadership in Action*, May/June 2009.
- [PBKC07] Adrian Paschke, Harold Boley, Alexander Kozlenkov, and Benjamin Craig. Rule Responder: RuleML-Based Agents for Distributed Collaboration on the Pragmatic Web. In *2nd ACM Pragmatic Web Conference 2007*. ACM, 2007.
- [PKB07] Adrian Paschke, Alexander Kozlenkov, and Harold Boley. A Homogenous Reaction Rule Language for Complex Event Processing. In *Proc. 2nd International Workshop on Event Drive Architecture and Event Processing Systems (EDA-PS 2007)*. Vienna, Austria, September 2007.
- [SLKL07] Anna Sachinopoulou, Juha Leppänen, Hannu Kaijanranta, and Jaakko Lähteenmäki. Ontology-Based Approach for Managing Personal Health and Wellness Information. In *Engineering in Medicine and Biology Society (EMBS 2007). 29th Annual Int’l Conference of the IEEE*, August 2007.

A Signatures of WellnessRules

We give exemplary signatures, for Activity and Meetup. The complete Wellness signatures are at <http://ruleml.org/WellnessRules/Signatures/>.

Activity

The objective of the wellness rules is to suggest a wellness activity to the user according to their profile preferences. Each activity’s requirements can be described by the user in an incremental manner.

POSL Syntax:

```
myActivity(?ProfileID,?Activity,?Ambience,?MinRSVP,?MaxRSVP,
           ?StartTime,?EndTime,?Location,?Duration,?FitnessLevel).
```

N3 Syntax:

```
_: myActivity
  rdf:type          :SuggestedActivity;
  :profileID        ?ProfileID;
  :activity          ?Activity;
  :inOut            ?Ambience;
  :minRSVP          ?MinRSVP;
  :maxRSVP          ?MaxRSVP;
  :startTime        ?StartTime;
  :endTime          ?EndTime;
  :location         ?Location;
  :duration         ?Duration;
  :fitnessLevel     ?FitnessLevel.
```

Usage:

ProfileID = The profile identification corresponding to the suggested activity.

Activity = The type of suggested activity determined.

Indoors/Outdoors = Whether the suggested activity is indoors or outdoors.

MinRSVP = Submitted minimum number of participants.

MaxRSVP = Submitted maximum number of participants.

StartTime = The start time of the suggested activity.

EndTime = The end time of the suggested activity.

Location = The meet up location for the suggested activity.

Duration = The duration of the suggested activity.

FitnessLevel = The required fitness level of the suggested activity.

Meetup

This defines the possible meetup locations for activities of participants. Each location is tied to its map and activity type.

POSL Syntax:

```
meetup(?MapID,?Activity,?Ambience,?Location).
```

N3 Syntax:

```
_: meetup
  rdf:type          :Meetup;
  :mapID           ?MapID;
  :activity         ?Activity;
  :inOut           ?Ambience;
  :location        ?Location.
```

Usage:

MapID = The map that the meetup location is assigned to. In order for profiles to share meetup locations, their MapID's must be the same.

Activity = The kind of activity being performed.

Indoors/Outdoors = Whether the activity is being performed inside or outside.

Location = The name of the meetup location.

B RDFS Taxonomy of WellnessRules

These RDFS type definitions are being maintained at <http://ruleml.org/WellnessRules/WR-Taxonomy.rdf>.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">
  <rdf:Description rdf:ID="Wellness">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Activity">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Wellness"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Walking">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Activity"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Running">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Activity"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Swimming">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Activity"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Skating">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Activity"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Yoga">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Activity"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Hiking">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Activity"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Baseball">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Activity"/>
  </rdf:Description>
</rdf:RDF>
```