



# **Social Semantic Rule Sharing and Querying in Wellness Communities**

Harold Boley, Taylor Osmun, Benjamin Craig

Institute for Information Technology, National Research Council, Canada

Fredericton, NB, Canada

4th Annual Asian Semantic Web Conference, Shanghai, China, Dec. 7-9, 2009



# Part 1: WellnessRules Foundation in Profile Knowledge Base Interoperation

- WellnessRules Overview
- Global and Local Knowledge Bases
- Profile Interoperation (POSL & N3)
  - ▣ Interoperation of Relational (POSL) and Graph (N3) Languages
  - ▣ POSL  $\Leftrightarrow$  N3 Transformation
- Taxonomy



*Later seen in Part 2:*

A use case demo describes an **online-interactive wellness community** through the WellnessRules system in Rule Responder

# WellnessRules Overview

- WellnessRules supports a wellness community that is **online-interactive** and **rule-supported**. Each participant of this community has the ability to:
  - ▣ **Create profiles about themselves**, containing their preferences for activities and nutrition, their event times, and their fitness levels
  - ▣ **Compare and collaborate with others** in the community, to track progress and schedule group events
- Rules about wellness opportunities are
  - ▣ **authored** in rule languages such as Prolog and N3
  - ▣ **interoperated** within the community using RuleML/XML



# Global Knowledge Base

- Contains knowledge relevant to everyone in the WellnessRules community
- Knowledge Areas:
  - **Season**
    - Defines timeframe of the seasons
  - **Forecast**
    - Describes the weather forecast within timeframes
  - **Meetup**
    - Contains activity meetup locations for maps



*Global Knowledge Base is available in [POSL](#) and [N3](#)*



# Local Knowledge Base

- Contains local knowledge specific to each participant in the WellnessRules community
- Knowledge Areas:
  - **Calendar**
    - Used for event planning. Allows for sharing of calendars between profiles
  - **Map**
    - Links to meetup locations. Allows for sharing of maps between profiles
  - **Fitness**
    - Defines expected fitness level for a specific period of time (scale of 1-10)
  - **Event**
    - Possible/Planned/Performing/Past
  - **MyActivity**
    - Derive participants' individual activity preferences



# Local MyActivity Sample Rule (in POSL)

```
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),
  greaterThanOrEqualTo(?ExpectedFitness,?Level),
  goodDuration(?Duration,?StartTime,?EndTime).
```

Orange  
designates a  
profile  
preference

- Based on this rule the following are p0001's preferences for Running outdoors:
  - The number of **participants** must be **within** the **minimum** and **maximum**
  - The season must be **summer**
  - It must **not** be **raining** outside
  - p0001's **fitness level** is **greater than or equal** to the **required fitness level**



# Profile Interoperation (POSL & N3)

- Support for both **logic-relational** (e.g., POSL) and **graph-oriented** (e.g., N3) knowledge representations
- Users may write their profile in either language
- Support for **OO jDREW** and **Euler** engines to execute queries issued to **POSL** and **N3** knowledge bases, respectively
- *Later seen in Part 2:*  
By using a RuleML subset as the exchange language and Rule Responder as the platform, queries and answers can be exchanged with all supported engines



# POSL

- POSL integrates **p**ositional and **s**lotted knowledge for humans (e.g., Prolog's positional and F-logic's slotted knowledge)
- WellnessRules only uses *positional POSL* for **logic-relational** knowledge, displayed in a Prolog-like syntax

- Positional Notations:

- Relation names: `season(?StartTime,summer).`

- Each fact and rule head has a **relation name**

- Arguments: `season(?StartTime,summer).`

- Arguments can be upper or **lower case**, separated by **commas** (“,”)

- Variables: `season(?StartTime,?).`

- Can be named (“**?**” **prefix**) or anonymous (**stand-alone** “**?**”)





# Notation 3 – N3

- N3 is a compact and readable version of **RDF**'s XML syntax. Uses triples (**subject, property, object**) to represent knowledge
- WellnessRules uses **N3** for *graph-oriented* knowledge
- Slotted Notations:

- Subject names:

“:” here denotes a local knowledge base

- Each fact / rule head has a **subject name**

```
:season_1
  rdf:type    :Season;
  :startTime  ?StartTime;
  :period     :summer.
```

- Variables:

- Can be named (“?” **prefix** ), or anonymous (**stand-alone** “?”)

```
:season_1
  rdf:type    :Season;
  :startTime  ?StartTime;
  :period     ?.
```

- Arguments (property-object pairs):

- Each argument must have a **property** (slot **name**):

```
:season_1
  rdf:type    :Season;
  :startTime  ?StartTime;
  :period     :summer.
```

- Each argument must also have an **object** (slot **value**):

```
:season_1
  rdf:type    :Season;
  :startTime  ?StartTime;
  :period     :summer.
```



# POSL $\Leftrightarrow$ N3 Transformation (Atoms)

- Transformations are bi-directional; harder left-right reading focused here
- N3 requires the use of subjects for naming relationships.  
The **subject name** is the “:”-prefixed relation “:name” extended by “\_i”, where “i” is an instance counter
- A *POSL relation name* becomes defined in an N3 rule head via an **rdf:type** property using the “:”-prefixed, uppercased version of the “:Name”

**POSL** `season(?StartTime,?Season).`



```
:season_1
rdf:type   :Season;
:startTime ?StartTime;
:period    ?Season.
```

**N3**

- Starting with *positional POSL*, **slot names** (properties) must be generated for N3, while **variables** and **constants** as **slot values** (objects) use the **same names** as in POSL

**POSL** `season(?StartTime,?Season).`



```
:season_1
rdf:type   :Season;
:startTime ?StartTime;
:period    ?Season.
```

**N3**



# POSL $\Leftrightarrow$ N3 Transformation (Rules)

- **Rule** transformation builds on atom transformation
- OO jDREW (using POSL) typically does **top-down** ( $\text{:-}$ ) reasoning; Euler (using N3) always does **bottom-up** ( $\text{=>}$ ) reasoning: 'conclude ... **from** ...' becomes 'if ... **then** ...'

```
myActivity(p0001,Running,in,?MinRSVP,?MaxRSVP,  
           ?StartTime,?EndTime,?Place,?Duration,?Level)
```

$\text{:-}$

```
...  
forecast(?StartTime,sky,?Weather),  
notEqual(?Weather,raining),  
...
```



```
{  
...  
?forecast  
  rdf:type           :Forecast;  
  :startTime         ?StartTime;  
  :aspect            :sky;  
  :value             ?Weather.  
  
  ?Weather log:notEqualTo :raining.  
...  
}  
  
 $\text{=>}$   
{  
  _:myActivity  
    rdf:type           :MyActivity;  
    :profileID        :p0001;  
    :activity         :Running;  
    :inOut            :in;  
    :minRSVP          ?MinRSVP;  
    :maxRSVP          ?MaxRSVP;  
    :startTime        ?StartTime;  
    :endTime          ?EndTime;  
    :location         ?Place;  
    :duration         ?Duration;  
    :fitnessLevel     ?FitnessLevel.  
}
```

N3



# POSL $\Leftrightarrow$ N3 Transformation (naf and built-ins)

- The POSL handling of **negation as failure** (**naf**) is via a primitive:

```
naf( event(?Calendar, ?Running, past, ?StartTimePast, ?EndTimePast))
```

POSL

Euler's N3 doesn't recommend the naf primitive.

Instead, naf is encoded by an **e:findall** expecting an **empty solution list** **()**

```
?NAF e:findall
  (?event
    {?event
      rdf:type      :Event;
      :calendarID   ?CalendarID;
      :aspect       :Running;
      :tense        :past;
      :startTime    ?StartTimePast;
      :endTime      ?EndTimePast.}
```

N3

() )

- POSL has **built-in** math operations.

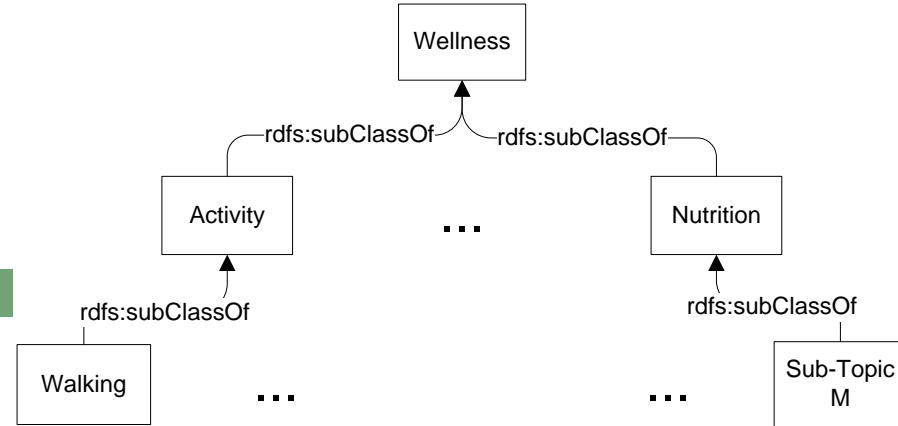
POSL

N3 uses **package-prefixed** math operations

N3

`greaterThanOrEqualTo(?ExpectedFitness,?Level),`  $\Leftrightarrow$  `?ExpectedFitness math:notLessThan ?FitnessLevel.`

# Taxonomy



- ❑ The WellnessRules taxonomy is broken into two topics: Activity and Nutrition
- ❑ Each of these contains multiple subtopics (e.g., Walking or Running)
- ❑ Both representations use **rdf:type**, **rdfs:Class** and **rdfs:subClassOf**
- ❑ Taxonomy classes act as user-defined types to restrict rule variables

## RDF (used by POSL)

N3

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <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>

  ...

```



```

@prefix : <wellnessRules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

:Wellness      rdf:type          rdfs:Class.
:Activity      rdf:type          rdfs:Class;
               rdfs:subClassOf  :Wellness.
:Walking      rdf:type          rdfs:Class;
               rdfs:subClassOf  :Activity.
...

```

## Part 2: WellnessRules for an Online-Interactive Wellness Community via Rule Responder

- WellnessRules as a Rule Responder
- WellnessRules Architecture
- WellnessRules Agent Implementation and Role Assignment Matrix
- Activity Scenario: Example Queries with Live Demo



*Previously seen in Part 1:*

Foundation of WellnessRules in **profile knowledge base interoperation**, which is assisted through **transformation techniques**

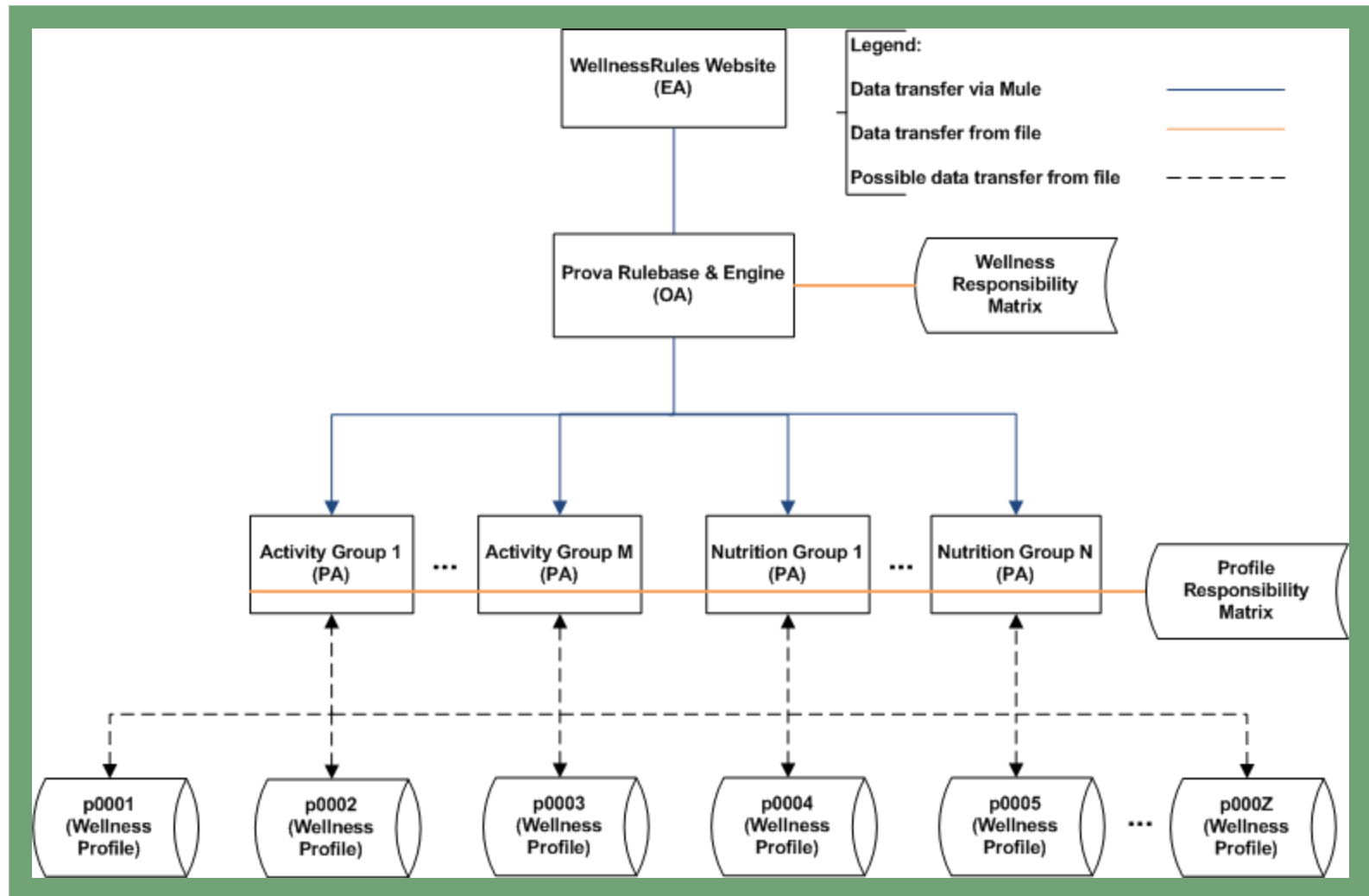


# WellnessRules as a Rule Responder

- Rule Responder is an intelligent multi-agent infrastructure for **collaborative** teams and **virtual communities**
- Rule Responder uses three kinds of agents:
  - ▣ **Organizational Agent (OA)**
  - ▣ **Personal Agents (PAs)**
    - **PAs extended to select relevant profiles of participants**
  - ▣ **External Agents (EAs)**
- The WellnessRules instantiation of Rule Responder (similar to [SymposiumPlanner](#)) employs the OA, PAs, and EAs for **communication** and **query delegation** to support an **online-interactive wellness community**



# WellnessRules Architecture



# Rule Responder Agents (OA, PAs, EAs)

- External Agent (EA):
  - ▣ The WellnessRules website (shown later)
- Organizational Agent (OA):
  - ▣ Contains a [Prova](#) knowledge base which is used for incoming queries: **directs** them to **appropriate PAs** via the **Wellness Responsibility Matrix**
- Personal Agent with Profiles (PA):
  - ▣ Consists of a Java servlet and – using the two embedded engines, **OO jDREW** and **Euler** – forwards the query to **POSL** and **N3** profile knowledge bases, respectively
  - ▣ It only has **access** to **profiles** which contain **relevant information** for its responsible activity by using the **Profile Responsibility Matrix**



# Role Assignment Matrix

- Composed of the **Wellness Responsibility Matrix (WRM)** and **Profile Responsibility Matrix (PRM)**
- The **WRM** contains information about PA **wellness responsibility**, written as an OWL light ontology. It defines which PA is best suited for different kinds of queries
- The **PRM** contains information about PA **profile responsibility** and the **format** of each profile knowledge base:

```
<Activity>
  <Walking>
    <ResponsibleProfile name="p0001" format="pos1"/>
    <ResponsibleProfile name="p0002" format="n3"/>
    <ResponsibleProfile name="p0003" format="pos1"/>
  </Walking>
  ...
</Activity>
```





# Website (EA)

- Used to **issue queries** to the WellnessRules **OA**
- Query is placed in the text box, in **RuleML** format
- The Send Message will issue the query to the OA
- A new screen containing a list of **answers** in **RuleML** will be presented
- **Query examples** are provided with their subsequent English descriptions. Can be modified to suit your query

## Navigation

[WellnessRules Home](#)

[Rule Responder](#)

## WellnessRules - The Activity Rule Responder

Use below box to send a query in [Reaction RuleML format](#) to the WellnessRules Organizational Agent:

```
<RuleML xmlns=
  "http://www.ruleml.org/0.91/xsd"
  xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://www.ruleml.org/0.91/xsd
    http://ibis.in.tum.de/research/
    ReactionRuleML/0.2/tr.xsd"
  xmlns:ruleml2007=
    "http://ibis.in.tum.de/projects/paw#">

  <Message mode="outbound"
    directive="query-sync">
    <oid>
      <Ind>WellnessRules</Ind>
    </oid>
    <protocol>
      <Ind>esb</Ind>
    </protocol>
    <sender>
      <Ind>User</Ind>
    </sender>

    <Atom>
      <Rel>myActivity</Rel>
      <Var>ProfileID</Var>
      <Ind>Running</Ind>
      <Var>InOut</Var>
      <Ind type="integer">1</Ind>
      <Ind type="integer">5</Ind>
      <Var>StartTime</Var>
      <Var>EndTime</Var>
      <Var>Location</Var>
      <Var>Duration</Var>
      <Var>FitnessLevel</Var>
    </Atom>
  </Message>
</RuleML>
```

Send Message

### Query Selection:

The drop-down boxes show sample queries you -- as an External Agent -- can send to the WellnessRules Organizational Agent. These examples can also act as initial templates that you can edit to create your own queries.

Activity Example Query 1

### English Description:

Is anyone interested in general Running (indoors or outdoors), with one to five people, for any (start and end) time, location, duration, and fitness level?

## Latest News

**October 15th, 2009**

Call for profiles: If you are interested in assisting the WellnessRules initiative, please write your own profiles (in either POSL or N3) and submit to the webmaster.

**October 14th, 2009**

WellnessRules fully implemented into Rule Responder. Support for both Euler and OOjDREW engines.

**September 14th, 2009**

Full Rule Responder implementation of WellnessRules begins.

**August 25th, 2009**

WellnessRules website updated to provide navigation and latest news.

**August 14th, 2009**

WellnessRules website launched. Basic Rule Responder implementation provided.

Online Demo:

<http://www.ruleml.org/WellnessRules/RuleResponder>

Sources Used by WellnessRules:

# Activity Scenario: Structured English

## **Introduction:**

- In this scenario a participant of WellnessRules (Peter) uses the system to find one or more partners for Running some time in the near future

## **Query 1:**

- Peter first asks the community if anyone at all is interested in running with 2 to 6 people. Assessing the answer to this, he finds that there are far too many candidates on the list, and decides to narrow down his question

## **Query 2:**

- He feels that he will continue to have a fitness level of 5 for Running, and so asks a refined question wanting only Level-5 activities. In the answer list he notices p0001 (John), who is someone he has previously performed cycling with. (He finds John's fitness level of 5 for running surprising, as he did not realize he was also a good Runner)

## **Query 3:**

- Now he wishes to run with John (perhaps in a race?) and so targets p0001, and that he prefers Joe's Gym as the location. Peter now receives a single, final answer on the list, from which he takes the type of running, time, and duration, to contact John for scheduling this event





# Example Query 1

Peter would like to go for a run at some point in time.  
He poses the following question:

**Green** designates a  
relation name

□ English Description:

Is **anyone** interested in  
general **Running**  
(**indoors** or **outdoors**),  
with **2**  
to **6** people,  
for any (**start**  
and **end**) time,  
**location**,  
**duration**,  
and **fitness level**?

**Blue**  
designates a  
constant.  
More will get  
**blue**  
as we progress

**Orange** designates  
a variable

RuleML



# Example Query 1 – POSL & N3

## RuleML

```
...  
<Atom>  
  <Rel>myActivity</Rel>  
  <Var>ProfileID</Var>  
  <Ind>Running</Ind>  
  <Var>InOut</Var>  
  <Ind type="integer">2</Ind>  
  <Ind type="integer">6</Ind>  
  <Var>StartTime</Var>  
  <Var>EndTime</Var>  
  <Var>Location</Var>  
  <Var>Duration</Var>  
  <Var>FitnessLevel</Var>  
</Atom>  
...
```

## N3

```
_:myActivity  
  rdf:type          :MyActivity;  
  :profileID        ?ProfileID;  
  :activity          :Running;  
  :inOut             ?InOut;  
  :minRSVP           2;  
  :maxRSVP           6;  
  :startTime         ?StartTime;  
  :endTime           ?EndTime;  
  :location           ?Location;  
  :duration           ?Duration;  
  :fitnessLevel      ?FitnessLevel.
```

## POSL

```
myActivity(?ProfileID,Running,?InOut,2:integer,6:integer,  
           ?StartTime,?EndTime,?Location,?Duration,?FitnessLevel).
```



# Example Query 2

Peter feels that he will continue to have a fitness level of 5 for Running. He poses the following question:

RuleML

```
...
<Atom>
  <Rel>myActivity</Rel>
  <Var>ProfileID</Var>
  <Ind>Running</Ind>
  <Var>InOut</Var>
  <Ind type="integer">2</Ind>
  <Ind type="integer">6</Ind>
  <Var>StartTime</Var>
  <Var>EndTime</Var>
  <Var>Location</Var>
  <Var>Duration</Var>
  <Ind type="integer">5</Ind>
</Atom>
...
```

## □ English Description:

Is **anyone** interested in general **Running** (**indoors** or **outdoors**), with **2** to **6** people, for any (**start** and **end**) time, **location**, and **duration**, at a fitness level of **5**?



# Example Query 3

Now he wishes to run with John, and so addresses p0001, and that he prefers Joe's Gym as the location. He poses the following question:

RuleML

```
...
<Atom>
  <Rel>myActivity</Rel>
  <Ind>p0001</Ind>
  <Ind>Running</Ind>
  <Var>InOut</Var>
  <Ind type="integer">2</Ind>
  <Ind type="integer">6</Ind>
  <Var>StartTime</Var>
  <Var>EndTime</Var>
  <Ind>joesGym</Var>
  <Var>Duration</Var>
  <Ind type="integer">5</Ind>
</Atom>
...
```

## □ English Description:

Is p0001 interested in general Running (indoors or outdoors), with 2 to 6 people, for any (start and end) time, at Joe's Gym, for any duration, at a fitness level of 5?



# Answer to Query 3

WellnessRules will return the answer seen below. This gives Peter all of the information he needs to contact John about scheduling this event.

RuleML

```
...
<Atom>
  <Rel>myActivity</Rel>
  <Ind>p0001</Ind>
  <Ind>Running</Ind>
  <Ind>in</Ind>
  <Ind type="integer">2</Ind>
  <Ind type="integer">6</Ind>
  <Ind>2009-06-15T10:15:00</Ind>
  <Ind>2009-06-15T11:15:00</Ind>
  <Ind>joesGym</Ind>
  <Ind>P60M</Ind>
  <Ind type="integer">5</Ind>
</Atom>
...
```

## □ English Description:

p0001 is interested  
in Running  
Indoors,  
with 2  
to 6 people,  
between 10:15AM and  
11:15AM  
on June 15th, 2009,  
at Joe's Gym,  
for 60 minutes,  
at a fitness level of 5.



# Conclusion

- The WellnessRules case study:
  - Uses a global as well as distributed local **knowledge bases** to support profile interoperation and querying
  - Demonstrates **profile interoperation** between ***logic-relational*** (e.g., POSL) and ***graph-oriented*** (e.g., N3) knowledge representations
  - Introduces an **extended** Rule Responder **architecture**, adding the **profile level** underneath the PAs
  - Supports an **online-interactive wellness community** through the WellnessRules system in Rule Responder



# Example of Interoperation

- According to the PRM, p0001's profile has a format of POSL:

```
<WellnessRules>
  <Activity>
    <Walking>
      <ResponsibleProfile name="p0001" format="posl"/>
      <ResponsibleProfile name="p0002" format="n3"/>
      <ResponsibleProfile name="p0003" format="posl"/>
    </Walking>

    <Running>
      <ResponsibleProfile name="p0001" format="posl"/>
      <ResponsibleProfile name="p0002" format="n3"/>
      <ResponsibleProfile name="p0003" format="posl"/>
    </Running>

    ...
  </Activity>
</WellnessRules>
```

- Since p0001's profile is transformable to N3, the format can be, too:

```
<WellnessRules>
  <Activity>
    <Walking>
      <ResponsibleProfile name="p0001" format="n3"/>
      <ResponsibleProfile name="p0002" format="n3"/>
      <ResponsibleProfile name="p0003" format="posl"/>
    </Walking>

    <Running>
      <ResponsibleProfile name="p0001" format="n3"/>
      <ResponsibleProfile name="p0002" format="n3"/>
      <ResponsibleProfile name="p0003" format="posl"/>
    </Running>

    ...
  </Activity>
</WellnessRules>
```

