Harold Boley¹, Omair Shafiq², Derek Smith³, Taylor Osmun³

¹Institute for Information Technology, National Research Council Canada, Fredericton, NB, Canada
²Department of Computer Science, University of Calgary, Calgary, AB, Canada
³Faculty of Computer Science, University of New Brunswick, Canada, Fredericton, NB, Canada

Keynote at
The 3rd Canadian Semantic Web Symposium (CSWS2011)

Vancouver, British Columbia, Canada, August 5, 2011
(Update: September 17, 2011)
PatientSupporter – Idea and Implementation

The **Social Web** (Web 2.0) meets the **Semantic Web** (ontologies and *rules*) in a **Social Semantic Web** (Web 3.0):

Patients advertise their **semantic** profiles to *socialize* with others in support groups

Implemented in initial **PatientSupporter** prototype
Part 1: PatientSupporter Foundation in Profile Knowledge Base Interoperation

- PatientSupporter Overview
- Ontologies & Rules, Global & Local Knowledge Bases
- Local Knowledge Base (Profile) Interoperation between Prolog-extending POSL & RDF-extending N3
  - Interoperation Principles for Relational (POSL) and Networked (N3) Languages
  - POSL ⇔ N3 Transformation
- Injuries Partonomy

Later seen in Part 2:
A use case demo describes an online-interactive Patients community through the PatientSupporter system in Rule Responder
PatientSupporter Overview

- PatientSupporter supports a patient community that is **online-interactive** and **rule-based**
- Each patient as participant of this community has ability to:
  - **Create profiles about themselves**, containing their preferences for injuries, healing stage, treatment level, possible collaboration times, etc.
  - **Compare and collaborate with others** in the community, to track progress and schedule group discussions
- Rules for patient profiles are:
  - **Authored** in rule languages such as POSL and N3
  - **Interoperated** within the community using RuleML/XML
Knowledge Bases

- **Ontologies — Global**
  - Group Responsibility Matrix (see later)
  - Body Partonomy (see later)
    - Structures sports injuries as commonsense ontology of affected body parts

- **Rules (including Facts) — Global and Local**
  - See below
Global Rule Base

- Contains knowledge relevant to everyone in the PatientSupporter community

- Knowledge Areas:
  - participation
    - Ensures that a patient’s preferred group size interval is within a requested group size interval
  - ageCheck
    - Ensures that a patient’s age is within a requested age interval
  - goodDuration
    - Ensures that a patient’s preferred call duration is within a requested start and end time
Local Rule Bases (Profiles)

- Contain local knowledge specific to each patient in the PatientSupporter community

- Knowledge Areas:
  - age
    - Own age
  - groupSize
    - Preferred group sizes for injuries
  - category
    - ‘In’ or ‘Out’ patient
  - treatment
    - Bandage, (Major or Medium or Minor)
    - Operation, (Major or Medium or Minor)
    - Medication or ChangeOfLifeStyle
  - healingStage
    - Fresh, Medium, Convalescent or Healed
  - duration
    - Preferred call duration
  - gender
    - Male or Female
  - communication
    - Channel and Contact names
  - timezone
    - Own time zone
  - MyDiscussion
    - Derive patients’ discussion preferences
Local MyDiscussion Sample Rule (in POSL)
– Centered on Participant p0004 as First Argument

dateTime[?StartYear, ?StartMonth, ?StartDay, ?StartHour, ?StartMinute],
dateTime[?EndYear, ?EndMonth, ?EndDay, ?EndHour, ?EndMinute],
MSN, ?Contact, ?Gender, ?TimeZone)
:-
ageCheck(p0004, ?MinAge, ?MaxAge, ?Age),
participation(p0004, ?Injury, ?MinRSVP, ?MaxRSVP),
communication(p0004, MSN, ?Contact),
event(p0004, ?Injury, Possible,
dateTime[?StartYear, ?StartMonth, ?StartDay, ?StartHour, ?StartMinute],
dateTime[?EndYear, ?EndMonth, ?EndDay, ?EndHour, ?EndMinute]),
goodDuration(p0004, ?Injury,
dateTime[?StartYear, ?StartMonth, ?StartDay, ?StartHour, ?StartMinute],
dateTime[?EndYear, ?EndMonth, ?EndDay, ?EndHour, ?EndMinute]),
category(p0004, ?Category),
treatment(p0004, ?Injury, ?Treatment),
healingStage(p0004, ?Injury, ?HealingStage),
gender(p0004, ?Gender),
timeZone(p0004, ?TimeZone).
Local MyDiscussion Profile Facts (in POSL) – Centered on Participant p0004 as First Argument

age(p0004,27:integer).
groupSize(p0004,?:Toe,2:integer,10:integer).
duration(p0004,?:Toe,dateTime[0:integer,0:integer,0:integer,0:integer,30:integer]).
category(p0004,Out).
treatment(p0004,?:Toe,MinorOperation).
healingStage(p0004,?:Toe,Medium).
communication(p0004,MSN,Sam24).
gender(p0004,Male).
timeZone(p0004,-400:integer).
Profile Interoperation (POSL & N3)

- Support for both logic-relational (e.g., POSL) and graph-networked (e.g., N3) knowledge representations
- Users may write their profile in either language
- Support for Prova, 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
POSLS integrates positional and slotted knowledge (e.g., Prolog’s positional and F-logic’s slotted knowledge)

PatientSupporter only uses positional POSL for logic-relational knowledge, displayed in a Prolog-like human-readable syntax

Positional relation descriptions (fact and query examples):
- **Relation names (f):**
  - Each fact and rule head has a relation name

- **Arguments (f):**
  - Constant arguments are strings or symbols, separated by commas (",")

- **Variables (q):**
  - Variables can be named (“?” prefix) or anonymous (stand-alone “?”)
Notation 3 (N3)

- **N3** is a compact, rule-extended version of RDF's XML syntax. Writes triples \((\text{subject}, \text{property}, \text{object})\) as subject descriptions.
- PatientSupporter uses **N3** for graph-networked knowledge.
- Slotted subject descriptions (fact and query examples):
  - **Subject names (f):**
    - Each fact/rule head has a subject name.
    - Example: `:treatment_1`.
  - **Arguments as property→object slots (f):**
    - Each argument must have a property (slot name).
    - Example: `:treatment_1`.
  - **Variables (q):**
    - Can be anonymous (stand-alone "?") or named ("?" prefix)

":" here denotes IRI of this local knowledge base.

"@" here denotes IRI of this local knowledge base.

\[ \begin{align*}
\text{treatment}_1 & \text{rdf:type :Treatment;} \\
& \text{:profileID :p0001;} \\
& \text{:injury :Toe;} \\
& \text{:treatment :Bandage.}
\end{align*} \]

\[ \begin{align*}
\text{treatment}_1 & \text{rdf:type :Treatment;} \\
& \text{:profileID :p0001;} \\
& \text{:injury :Toe;} \\
& \text{:treatment :Bandage.}
\end{align*} \]

\[ \begin{align*}
\text{treatment}_1 & \text{rdf:type :Treatment;} \\
& \text{:profileID :p0001;} \\
& \text{:injury :Toe;} \\
& \text{:treatment :Bandage.}
\end{align*} \]
Transformations are bi-directional; harder left-right reading focused here.

N3 uses subjects to provide named instances/relationships. The subject name is the ":"-prefixed relation ":name" extended by "_i" where "i" is an instance counter. A "?"-prefixed variable can stand for a subject name.

A POSL relation name becomes defined in an N3 rule head via an rdf:type property using the ":"-prefixed, uppercased version of the ":Name".

Starting with positional POSL, slot names (properties) can be generated and stored in signatures for reuse; variables and constants as slot values (objects) employ the same names as in POSL.

```
POSL treatment(p0001,?, ?Treatment) ↔ N3

?treatment rdf:type :Treatment;
  :profileID :p0001;
  :injury ?;
  :treatment ?Treatment.
```

```
POSL treatment(p0001,?, ?Treatment) ↔ N3

?treatment rdf:type :Treatment;
  :profileID :p0001;
  :injury ?;
  :treatment ?Treatment.
```
**Rule** transformation builds on atom transformation

- OO jDREW (using POSL) typically does **top-down** (\( \leftarrow \)) reasoning;
- Euler (using N3) always does **bottom-up** (\( \Rightarrow \)) reasoning:
  - ‘conclude ... from ---’ becomes ‘if --- then ---’

```posl
{...
  ?treatment rdf:type :Treatment;
  :profileID :p0004;
  :injury :Toe;
  :treatmentType ?Treatment.
  ...

?healingStage rdf:type :HealingStage;
...
}
\(
 \leftarrow
\)
```

```n3
{...
  ...
  _:myDiscussion rdf:type :MyDiscussion;
  :profileID :p0004;
  :injury :Toe;
  ...
  :treatment ?Treatment;
  :healingStage ?Stage;
  ...
}
```
Body Partonomy

- The PatientSupporter partonomy is organized as hierarchy of body parts.
- Each of these corresponds to an injury (e.g., Leg, Foot, Toe, etc.).
- Mapped to a taxonomy, using `rdf:type`, `rdfs:Class`, and `rdfs:subClassOf`.
- Taxonomy classes act as user-defined types to restrict rule variables.

**RDF** (used by POSL, equivalent to a version in N3)

```xml
<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="Body">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>

  <rdf:Description rdf:ID="Leg">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#Body"/>
  </rdf:Description>

  ...
```
Part 2: PatientSupporter for an Online-Interactive Patient Community via Rule Responder

- PatientSupporter as a Rule Responder
- PatientSupporter Conceptual View
- PatientSupporter Technical View
- PatientSupporter Agent Implementation

Discussion Scenario: Example Queries with Live Demo

Previously seen in Part 1:
Foundation of PatientSupporter in profile knowledge base interoperation, which is assisted through transformation techniques
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 patients
- **External Agents (EAs)**

The PatientSupporter instantiation of Rule Responder employs the **OA, PAs, and EAs** for **communication and query delegation** to support an **online-interactive community of patients**

- Similar to **WellnessRules2**, **WellnessRules**, and **SymposiumPlanner**
Rule Responder
Conceptual View
Communication using Mule Enterprise Service Bus

- Mule ESB Open Source
- Message Platform and distributed Object Broker
- Event Driven Architecture (EDA)
- Around 50 Protocols (JMS, HTTP, SOAP …)
- Synchronous and Asynchronous Communication
- Message Driven Event Processing
PatientSupporter Architecture

Legend:
- Data transfer via Mule
- Data transfer from File
- Possible data transfer from File

PatientSupporter Website (EA)

Prova Rulebase & Engine (OA)

Group Responsibility Matrix

Profile Responsibility Matrix

Patient Group PA_1

Patient Group PA_2

Patient Group PA_3

... Patient Group PA_n

P0001 (Patient)
P0002 (Patient)
P0003 (Patient)
P0004 (Patient)

... P000n (Patient)

Legend:
- Data transfer via Mule
- Data transfer from File
- Possible data transfer from File
Rule Responder Agents (EA, OA, PAs)

- **External Agent (EA):**
  - The PatientSupporter website (shown later)

- **Organizational Agent (OA):**
  - Contains a Prova knowledge base which is used for incoming queries: directs them to appropriate PAs via the Group 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 injury by using the Profile Responsibility Matrix
Group and Profile Responsibility Matrix

- Role assignment on two levels: **Group Responsibility Matrix (GRM)** and **Profile Responsibility Matrix (PRM)**

- The **GRM** contains information about PA **injury 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:

```xml
<Injury>
  <Leg>
    <ResponsibleProfile name="p0001" format="posl"/>
    <ResponsibleProfile name="p0002" format="n3"/>
    <ResponsibleProfile name="p0003" format="posl"/>
  </Leg>
  ...
</Injury>
```
Website (EA)

- EA is used to issue queries to the PatientSupporter OA
- Query can be written manually or constructed by using the Javascript based GUI
- Query is placed in the text box, in the 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

Online Demo: [http://ruleml.org/PatientSupporter/RuleResponder](http://ruleml.org/PatientSupporter/RuleResponder)
Discussion Group Formation Scenario: Query Refinement

Introduction:

- In this scenario a participant of PatientSupporter (Paul) uses the system to find one or more partners for a Discussion on his and others’ Toe injuries

Query 1:

- Paul first asks the community if some are interested in discussing Leg injuries with 2 to 10 people. Assessing the answers to this, he finds that there are too many candidates on the list, and decides to narrow down his question

Query 2:

- Paul fixes the date and limits possible times, which helps to reduce the numbers, but he still gets too many answers. While these constraints may be ‘hard’ for him (in this week), he should be aware that he may thereby decrease body-part recall (see Queries 3-4)

Query 3:

- Now he decides to further descend into the body partonomy, asking about Foot injuries (keeping the constraints of Query 2). The answer volume seems fine, but he now wants maximum body-part precision

Query 4:

- So, finally he focuses on Toe injuries. Paul now receives a single answer, from which he takes the contact name, Sam24, for agreeing on an MSN call
Is anybody interested in joining a group about leg injuries, between 20 and 50 years of age, for 2 to 10 people, for any category (In or Out) that are treated with any treatment at any healing stage? It would be on any date between any (start and end) times, for ... 0 hours and 30 minutes, on any channel, with any contact, gender, and time zone.
Example Query 1 – POSL & N3

POSL

```xml
...<Atom>
   <Rel>myDiscussion</Rel>
   <Var>ProfileID</Var>
   <Var type="Leg">Injury</Var>
   <Ind type="integer">20</Ind>
   <Ind type="integer">50</Ind>
   <Ind type="integer">2</Ind>
   <Ind type="integer">10</Ind>
   <Var>Category</Var>
   <Var>Treatment</Var>
   <Var>HealingStage</Var>
   <Expr><Fun>dateTime</Fun>...</Expr>
   <Expr><Fun>dateTime</Fun>...</Expr>
   <Expr><Fun>dateTime</Fun>...</Expr>
   <Var>Channel</Var>
   <Var>Contact</Var>
   <Var>Gender</Var>
   <Var>TimeZone</Var>
</Atom>
...```

RuleML

```xml
_:myDiscussion
   rdf:type :MyDiscussion;
   :profileID ?ProfileID;
   :injury Injury:Leg;
   :minAge 20;
   :maxAge 50;
   :minRSVP 2;
   :maxRSVP 10;
   :category ?Category;
   :treatment ?Treatment;
   :healingStage ?Healing;
   :startTime ?StartTime;
   :endTime ?EndTime;
   :duration [rdf:type :DateTime;
      :year 0;
      :month 0;
      :day 0;
      :hour 0;
      :minute 30];
   :channel ?Channel;
   :contact ?Contact;
   :timeZone ?TimeZone.

myDiscussion(?ProfileID,Leg,20:integer,50:integer,2:integer,10:integer,?Category,
   ?Treatment,?HealingStage,
   ?startTime,?EndTime,
   ?duration[year 0;month 0;day 0;hour 0;minute 30];
   ?Channel,?Contact,?Gender,?TimeZone).
```

N3

```n3
myDiscussion(?ProfileID,Leg,20:integer,50:integer,2:integer,10:integer,?Category,
   ?Treatment,?HealingStage,
   ?startTime,?EndTime,
   ?duration[year 0;month 0;day 0;hour 0;minute 30];
   ?Channel,?Contact,?Gender,?TimeZone).
```
Example Query 2

Is anybody interested in joining a group about Leg injuries, between 20 and 50 years of age, for 2 to 10 people, for any category (In or Out) that are treated with any treatment at any healing stage? It would be on 05/21/2010 between 10:00 and 11:40, for ... 0 hours and 30 minutes, on any channel, with any contact and gender, and on GMT -4:00.
Example Query 3

Paul focuses on Foot as part of Leg

RuleML

□ English Description:

Is anybody interested in joining a group about Foot injuries, between 20 and 50 years of age, for 2 to 10 people, for any category (In or Out) that are treated with any treatment at any healing stage? It would be on 05/21/2010 between 10:00 and 11:40, for ... 0 hours and 30 minutes, on any channel, with any contact and gender, and on GMT -4:00.
Paul focuses on Toe as part of Foot

RuleML

Example Query 4

English Description:

Is anybody interested in joining a group about Toe injuries, between 20 and 50 years of age, for 2 to 10 people, for any category (In or Out) that are treated with any treatment at any healing stage? It would be on 05/21/2010 between 10:00 and 11:40, for ... 0 hours and 30 minutes, on any channel, with any contact and gender, and on GMT -4:00.
p0004 is interested in joining a group about Toe injuries, between 20 and 50 years of age, for 2 to 10 people, as an Out patient that was treated with a Minor Operation at Medium healing stage. It would be on ..., channel MSN, with contact name Sam24, gender Male, and on GMT -4:00.
Conclusion

- The PatientSupporter 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-networked (e.g., N3) knowledge representations
  - Employs an extended Rule Responder architecture, containing the profile level underneath the PAs
  - Supports online-interactive communities of patients guided by a body partonomy mapped to a taxonomy
  - This Rule Responder version thus ontologizes the architecture of earlier instantiations WellnessRules (2) and SymposiumPlanner