

PROJECT REPORT

ON

Project Title – Find me an Event!

COURSE NAME – ARTIFICIAL INTELLIGENCE

Submitted On – May 08, 2015

Submitted By – Kanika Kapoor

Kxk140230

- **Problem Description**

At UTD, there are several events and meeting that take place each day throughout the week and students are not sure which event to attend. This project aims at developing a system through which a student can decide the events, based on his mood, timings of event, kind/type of event, place of event, food etc. The system would return the most suitable event that the student must attend according to his mood and other parameters.

- **Proposed Solution**

1. Informed Search Algorithm:

a. The project will make use of A* algorithms to produce an optimal path from the source node (current destination at university) to the goal node (event at university).

2. Heuristic used

- Straight Line Distance to the event
- Hops required to reach the destination + Straight Line Distance

- **Full implementation details**

The project initially requires the creation of a knowledge base in PROTEGE and querying using SPARQL. These are several classes and subclasses with different data properties that are described below:

CLASSES

CLASS NAME	SubClassOf
Building	Thing
Event	Thing
Location	Thing
User	Thing
Cultural	Event
FraternityMeets	Event
Informational	Event
Programming	Event
Volunteer	Event
Contest	Programming
Workshop	Programming
OffCampus	Location
OnCampus	Location
Bored	User
Happy	User
Helpful	User
Neutral	User
Peaceful	User
Studious	User

INSTANCES

CLASSNAME	INSTANCES
Building	ActivityCenter
	ATEC
	ClarkCenter
	ErikJonsonn
	FoundersNorth
	GreenHall
	JSOM
Cultural	McDermottLibrary
	Bhangra_Night
	Chinese_New_Year
	ISA_Diwali
	St_Louis_Day_Celebration
FraternityMeets	Texas_Party
	Alpha_Kappa_Psi_Induction_Ceremony
	Alpha_Phi_theta_Fellowship
	Delta_tau_Delta_Chapter_meeting
	Golden_Key_Meeting
Informational	Living_Water_Student
	Model_based_Simulation_threat
	State_farm_Intern_Fair
	ETS_Workshop
	OPT_Workshop
Contest	JAVA_App_Contest
	RUBY_On_Rails
Programming	Formatting_Excel
	GMAT_prep_Course
	Android_beginner_Workshop
Volunteer	Learn_PERL
	Animal_Shelter_Drive
	Radio_UTD_Volunteering
	Catch_Comet_pride_Series
	Bronies_UTD_meeting
Bored	Preview_Friday
	Bored
	Happy
	Helpful
	Neutral
	Peaceful
Peaceful	Peaceful
Studios	Studios

OBJECT PROPERTIES

PROPERTYNAME	DOMAIN	RANGE
conducts	Building	Event
attend	User	Event

DATA PROPERTIES

PROPERTYNAME	DOMAIN	RANGE
has_Auditorium	Building	integer
has_Duration	Event	integer
has_Fees	Event	boolean
has_Food	Event	boolean
reach_By	Building	string
start_Time	Event	dateTime

- Examples

Query1: Select all the events that are OffCampus

```
SELECT ?eventName
WHERE
{
  ?eventName rdf:type my:OffCampus
}
```

The screenshot shows the Protege-4.5.5 SPARQL query interface. The query is executed, and the results are displayed in a table. The table has one column, 'eventName', and lists eight events: ETS_Workshop, Animal_Shelter_Drive, Bronies_UTD_Meet, Alpha_Kappa_Psi_Induction_Ceremony, Living_Water_Student_Fellowship_Meet, St_Louis_Day_Celebration, and RUBY_On_Rails_Contest. The interface also shows the SPARQL query text, the ontology metrics, and the class axioms.

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>

SELECT ?eventName
WHERE { ?eventName rdf:type my:OffCampus }
```

Ontology metrics:

Metrics	
Axiom	369
Logical axiom count	296
Class count	19
Object property count	2
Data property count	6
Individual count	39
DL expressivity	AL(D)

Class axioms:

Class axioms	
SubClassOf axioms count	15
EquivalentClasses axioms count	0
DisjointClasses axioms count	0
GCI count	0
Hidden GCI Count	0

Results:

eventName
ETS_Workshop
Animal_Shelter_Drive
Bronies_UTD_Meet
Alpha_Kappa_Psi_Induction_Ceremony
Living_Water_Student_Fellowship_Meet
St_Louis_Day_Celebration
RUBY_On_Rails_Contest

Query – 2 Show the user what events he can attend if he is feeling Studious.

```

SELECT ?ce
WHERE
{
  ?bi my:attend ?ce.
  ?bi rdf:type my:Studious}

```

The screenshot shows the Protégé application window titled "untitled-ontology-18". The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help), a toolbar with navigation icons, and a search bar labeled "Search for entity". Below the toolbar is a tabbed interface with tabs for "Data Properties", "Individuals by class", "DL Query", "OntoGraf", and "SPARQL Query". The "SPARQL Query" tab is active, displaying the following query:

```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>

SELECT ?ce
WHERE
{
  ?bi my:attend ?ce.
  ?bi rdf:type my:Studious
}

```

Below the query editor, the results are displayed in a table with a single column header "ce". The results are:

ce
GMAT_Prep_Course
RUBY_On_Rails_Contest
Android_Beginner_Workshop
Formatting_Excel
JAVA_App_Contest
Learn_PERL

Query – 3 If user is bored, want to attend Off Campus event and wants to have food at event.

```
SELECT ?ce
WHERE
{
  ?bi my:attend ?ce.
  ?bi rdf:type my:Bored
  ?ce rdf:type my:OffCampus}
```

The screenshot shows a web-based SPARQL query editor. The title bar indicates the ontology is 'untitled-ontology-18' from a specific URL. The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help) and a toolbar with navigation and search icons. Below the toolbar, there are tabs for 'Data Properties', 'Individuals by class', 'DL Query', 'OntoGraf', and 'SPARQL Query'. The 'SPARQL Query' tab is active, displaying the query text. Below the query, the results are shown in a table with one column labeled 'ce' and one row containing the value 'St_Louis_Day_Celebration'.

untitled-ontology-18 (<http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18>) : [/Users...

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-18 Search for entity

Data Properties x Individuals by class x DL Query x OntoGraf x SPARQL Query x

Active Ontology x Entities x Object Properties x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>

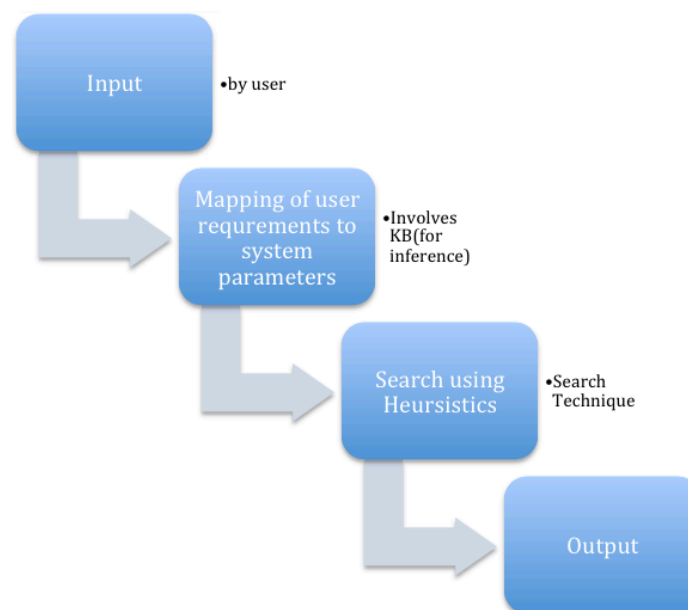
SELECT ?ce
WHERE
{
  ?bi my:attend ?ce.
  ?bi rdf:type my:Bored.
  ?ce rdf:type my:OffCampus
}
```

ce
St_Louis_Day_Celebration

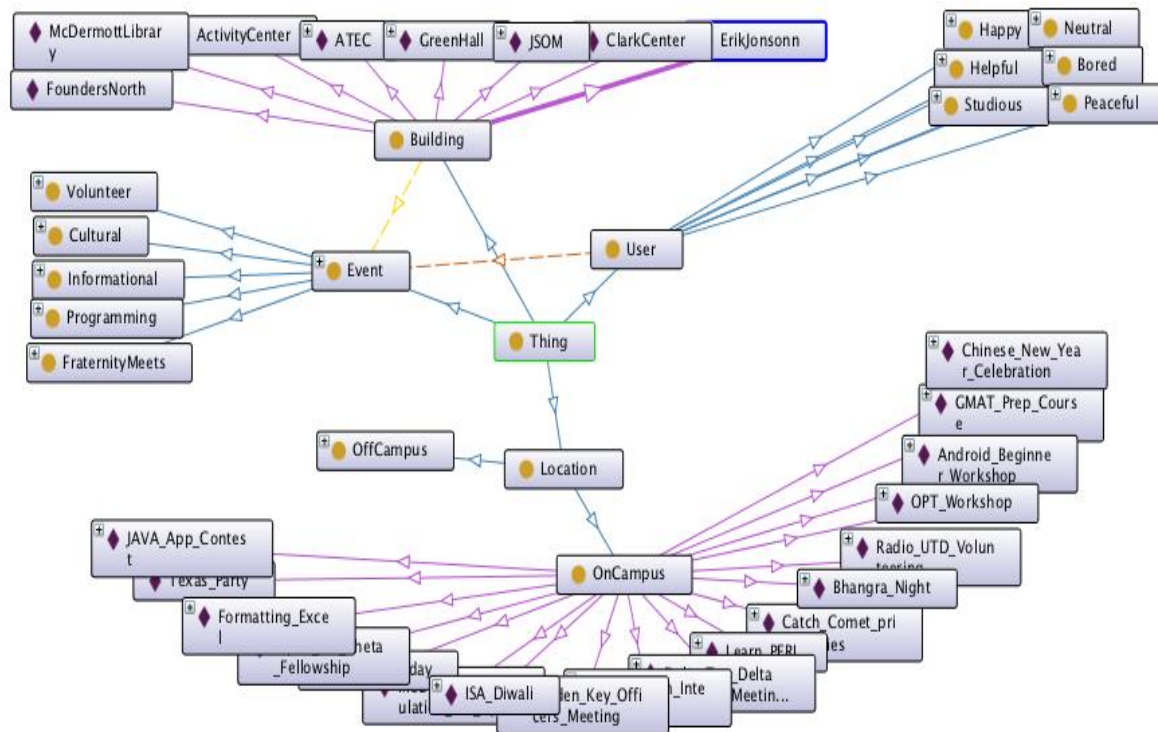
- **Programming tools**

- Protégé for creation of knowledge base
- Jena v 2.12.2 libraries for interfacing Protégé and Java
- Java Eclipse IDE

- **Architectural Diagram**



• ONTOGRAF



• Summary of the problems encountered

PROTEGE ISSUES

- Creation of the knowledge base and creation of classes for the representation
- Deciding the domain and ranges of each Object Property and Data Property
- Assigning values of data property to each of instance.

SPARQL ISSUES

- Understanding of triple pattern i.e. subject, object and Predicate.
- Parsing the Result Set in JAVA

Note: The use of SPARQL is easy otherwise if the triplet meaning can be understood wisely and it gives very appropriate and formatted results.

SEARCH ALGORITHM

- Implementation of A* algorithm was difficult as the project had dynamic Start Node and dynamic End Node. It was thus a challenge to keep track of all heuristic values and calculating the correct $f(n)$ value.

GENERAL ISSUES

- Mapping of user data to the knowledge stored in KB as user sometimes enters incorrect values.
- Understanding of classes, subclasses and Properties in SPARQL and their usage.

Pending Issues

While implementing the SPARQL queries using JENA, I am taking user from input and mapping it to KB at each step. This is because I have used 2-3 data properties as of now for querying the KB.

Potential Improvements

Mapping of all data properties for better and appropriate results.