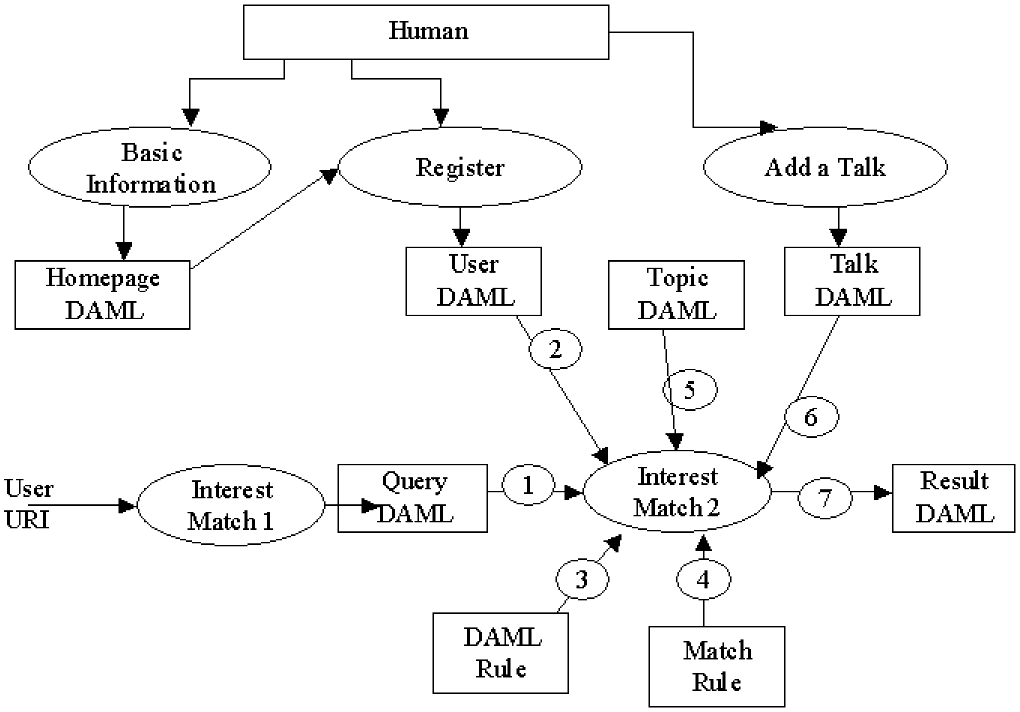
DAML Inference Engine

1. Prolog based Inference Engine
   http://www.cs.umbc.edu/~yzou1/inferengine.htm



1.1 User put Daml content file in his homepage.

Every user must generate a DAML file and put it in their homepage. It can generate using notepad, vi, or use http://www.cs.umbc.edu/~yzou1/cgi-bin/inputinfo.cgi , which get the input of user, generate a DAML file based on ITTalk schema 's definition of class Person, but it only include some attributes like name, email, phonenumber.

1.2 User register at Ittalk website

When user goto http://www.cs.umbc.edu/~yzou1/cgi-bin/register.cgi to register, they can either fill the form and input everything, or, submit a URI to their homepage daml file. The register program will fetch the user 's daml file, set the value in the file as the default form value. It will generate a DAML file based on ITTalk schema 's definition of class Person.

1.3 ITTalk editor add a talk

Editor of ITTalk can add a new talk into database at http://www.cs.umbc.edu/~yzou1/cgi-bin/addtalk.cgi. It will generate a DAML file based on ITTalk schema 's definition of class Talk.

1.4 Query and Result daml schema

We assume inference engine only accept input of DAML file. So, we need to define the daml schema for query and query result. http://www.cs.umbc.edu/~yzou1/cgi-bin/query.daml

Query class include property:

Query string: interest_match(“http://www.ittalk.com/user1.daml”,Result)

Scope: search one file or search all IT talks

Limit: limit result number

Result class include property:

Resultitem: one result, it defined as Thing, so can be any type of resource.

Resultsize : result item number

1.5 DAML parse rule

To easily parse DAML term in prolog, we must define rule for every DAML term.

Such as: daml:subClassOf , we define subclass(A,B), A is a subclass of B

```
subclassof(A,B) :- rdf(A, 'http://www.daml.org/2000/10/daml-ont#subClassOf' B).
subclassof(A,B) :- subclassof(A,C), rdf(C, 'http://www.daml.org/2000/10/daml-ont#subClassOf' B).
```

1.6 Query rule

We will do five kinds of query in this system. We need to define query rule for each query.

1.6.1 Interest match http://www.cs.umbc.edu/~yzou1/cgi-bin/interestmatch.htm

Define the match rule for this query,

```
interest_match(Querydaml,Result):-
        rdf_load(Useruri)
        rdf(_,'http://www.ittalk.org/ittalk#interest',Interest1),
        rdf(_,'http://www.ittalk.org/ittalk#name',Name1),
        Name1 = literal(Name0),

        rdf_load('topic.daml'),
        rdf(Des,'http://www.ittalk.org/ittalk#topicname',Interest1),
        match(Interest1,Sub1),

        rdf_load('ittalk.daml'),
        rdf(Des1,'http://www.ittalk.org/ittalk#topic',Sub1),
        rdf(Des1,'http://www.ittalk.org/ittalk#talkdate',Talkdate1),… get all the item ,
generate daml file and return in Result.
```

1.6.2 Search all talk between time 1 and time 2

1.6.3 …..

1.6.4 …..

1.6.5 …..

1.7 ACM Topic DAML file

Define ACM topic which used to match user interest, such as :

```
<topic>
    <topicname> ai </topicname>
    <subtopic>agent </subtopic>
</topic>
```

2. Java based Inference Engine
    - User submit a URI of DAML file (about one IT talk)
    - Servlet program start: call class SirPAC parse file into triple, save it into prolog through jasper interface and Sicstussoma (by Harry Chen).
    - User submit a URI of DAML file (query)
    - Servlet program start: call class SirPAC, parse file into triple, through jasper interface, load RDF and DAML rule (rdf.pl, daml,pl), (need load other rules like topic match, personal interest, maybe in DAML file or pl file), read ittalk.list, put triple into sicstus as TERM, put query triple into sicstus as a openQuery, get query result, put result into a daml file. Return result to user.

    May also make use of DAML Viewer http://www.daml.org/viewer/ when do daml parsing.

3. Related software
    - Libwww: http://www.w3.org/Library/, **C language**, RDF Parser HTRDF.c based on Expat, similar to java based SirPAC.
    - SirPAC: http://www.w3.org/RDF/Implementations/SiRPAC/ , **java language,** based on SAX
    - SiLRI: http://www.aifb.uni-karlsruhe.de/~sde/rdf/, http://www.ontoprise.de/, Logic-based RDF Interpreter, **java language** , based on sirpac. Define rule, query, rdf file.
    - RDF-Parser from Pro-Solution: http://www.pro-solutions.com/download/Parser-api, **perl and java,** Promenade-RDF-Parser, based on Apache Xerces XML parser and rdf-api
    - RDF-API: http://www-db.stanford.edu/~melnik/rdf/api.html, **Java language.** The API defines interfaces for parsing (org.w3c.rdf.syntax) and accessing RDF models as sets of statements (org.w3c.rdf.model). Default implementations for those interfaces are included in the org.w3c.rdf.implementation package (SirPAC) , use by DAML Crawler http://www.daml.org/crawler/
    - Sgml RDF parser for SWI-Prolog: http://www.swi.psy.uva.nl/projects/SWI-Prolog/packages/sgml/ **prolog** based RDF parser, use c library sgml2pl to do parser XML file. Use rdf.pl rdf_db.pl and rdfs.pl to do RDF parser.
    - VRP: http://www.ics.forth.gr/proj/isst/RDF/, **java(cup,jflex) ,** RDF validation, checking RDF Schema constraints.
    - PiLLoW web programming library: http://www.clip.dia.fi.upm.es/miscdocs/pillow/pillow.html, HTML/XML/HTTP conectivity package for Logic Programming and Constraint Programming systems.
    - Search and result : http://www-diglib.stanford.edu/diglib/ginf/WD/ginf-dl-app/examples.html
    - RDFdb http://web1.guha.com/rdfdb/query.html used by DAML Crawler .