Object-Oriented KRR with Flora-2 – Access Control and Privacy Control Example

CSE 505 – Computing with Logic
Stony Brook University
http://www.cs.stonybrook.edu/~cse505
Knowledge Representation and Reasoning with Flora-2

- Example: Social networks have complex information access and privacy policies.
  - In this example, we model such a network and use it to create various views based on these policies for friends, public, private and groups
- A user has several properties with various access policies:

```plaintext
User[]
// String_Object, Gender_Object,
// Location, and others are subclasses of
// Access_Controlled and can have access
// permissions
first_name => String_Object,
last_name => String_Object,
profile => Profile_Object,
```
Knowledge Representation and Reasoning with Flora-2

cover => Cover_Object,
gender {0..1} => Gender_Object,
email => String_Object,
birthday {0..1} => Birthday_Object,
userid {1..1} => String_Object,
user_name => String_Object,
education_history => School_Attendance,
job_history => Job_Held,
relationship => Relationship,
location => Location,
content => Content,

// created objects: Page, Event, Group
creates => Created_Objects,
likes => LikeContent,
timeline => Timeline_Object,
transaction => Transaction,
A class of all objects to which access can be controlled:

\[
\text{Access\_Controlled}[|
\text{read(Access\_Entity)} \Rightarrow \backslash boolean,
\text{write(Access\_Entity)} \Rightarrow \backslash boolean,
\text{find(Access\_Entity, Access\_Entity)} \Rightarrow \backslash boolean,
\text{default values inherited by all objects unless overwritten:}
\text{read(?) } \Rightarrow \backslash false,
\text{write(?) } \Rightarrow \backslash false,
\text{find(?, ?) } \Rightarrow \backslash false
|].
\]
Knowledge Representation and Reasoning with Flora-2

- Entities that can have access control privacy control:
  
  \{male,female\}:Gender_Object.
  Gender_Object::Access_Controlled[|
      value => Gender_Type
  |].

  Birthday_Object::Access_Controlled[|
      value => \date
  |].

  \{spouse,friend,girlfriend,parent,child\} : Relationship.
  Relationship::Access_Controlled[|
      type => Relationship_Type,
      person => User
  |].
Attendance::Access_Controlled[
  institution {1..1} => \string,
  start {0..1} => \date,
  end {0..1} => \date,
  address => Address
].
{School_Attendance,Job_Held} :: Attendance.

School_Attendance[
  status => \string,
  level => \string
].

Job_Held[
  position => \string
].

(c) Paul Fodor (CS Stony Brook)
Knowledge Representation and Reasoning with Flora-2

```
Location::Access_Controlled[ |
    country => string,
    region  => string,
    city    => string,
    latitude=> decimal,
    longitude=> decimal
| ].

// general address
Address :: Location[ |
    street  {0..1} => string,
    number  {0..1} => string,
    apartment{0..1} => string,
    zipcode {0..1} => string
| ].
```
Knowledge Representation and Reasoning with Flora-2

\{Photo, Video, Comment, Note, Group, Event, Link, StatusPost, Message\} :: Content.

Content::Access_Controlled[
  author \{1..1\} => User,
  creation_time \{1..1\} => \datetime,
  description => \string,
  comment => Comment,
  tags => Tag,
  audience => Audience
].

Timeline[
  content => Content
].

\{Public, Friends, FriendsofFriends, OnlyMe\} : Audience.
Social networks also provide access to purchasing products and store information about login and transactions:

\[
\text{Product}[] = \\
\text{name} \{1..1\} => \text{string}, \\
\text{owner} => \text{string}, \\
\text{price} \{1..1\} => \text{decimal}, \\
\text{description} => \text{string} \\
\].

\[
\text{Transaction::Access_Controlled}[] = \\
\text{account} => \text{Account}, \\
\text{time} => \text{datetime}, \\
\text{product} => \text{Product}, \\
\text{amount} => \text{decimal} \\
\].
Knowledge Representation and Reasoning with Flora-2

Account[
  bank_name {1..1} => \string,  
  account_number {1..1} => \string,  
  created {1..1} => \date,  
  owner {1..*} => User
].

Cookie[
  device => \string,  
  browser => \string,  
  os => \string,  
  location => Location,  
  IP_address => \string,  
  login => \string,  
  time => \datetime
].
Knowledge Representation and Reasoning with Flora-2

paul:User[
  relationship -> paul_mary_relationship,
  relationship -> paul_john_relationship,
  relationship -> paul_mike_relationship,
  relationship -> paul_jack_relationship,
  timeline -> paul_timeline
].

paul_mary_relationship:Relationship[
  type -> spouse,
  person -> mary
].

paul_john_relationship:Relationship[
  type -> child,
  person -> john
].
Knowledge Representation and Reasoning with Flora-2

```
paul_jack_relationship:Relationship[
  type -> friend,
  person -> jack
].
paul_timeline:Timeline[
  content -> post1,
  content -> photo1
].
post1:Post[
  value -> "I am in Berlin",
  audience -> Family
].
photo1:Photo[
  imageName -> "Berlin 1",
  audience -> Friends
].
```
Knowledge Representation and Reasoning with Flora-2

mary:User[
  relationship -> mary_paul_relationship,
  relationship -> mary_john_relationship,
  relationship -> mary_mike_relationship,
  relationship -> mary_jack_relationship,
  relationship -> mary_jane_relationship,
  timeline -> mary_timeline
].

mary_paul_relationship:Relationship[
  type -> spouse,
  person -> paul
].

mary_john_relationship:Relationship[
  type -> child,
  person -> john
].
mary_jack_relationship:Relationship[
  type -> friend,
  person -> jack
].
mary_jane_relationship:Relationship[
  type -> friend,
  person -> jane
].
mary_timeline:Timeline[
  content -> post2,
  content -> photo2
].
post2:Post[
    value -> "I am in Berlin",
    audience -> Family
].

photo2:Photo[
    imageName -> "Berlin 1",
    audience -> Friends
].

john:User[
    relationship -> john_paul_relationship,
    relationship -> john_mary_relationship,
    relationship -> john_mike_relationship,
    relationship -> john_jasmine_relationship,
    relationship -> john_sun_relationship,
    timeline -> john_timeline
]. ...
Knowledge Representation and Reasoning with Flora-2

// a user's timeline
timeline(?User, ?ListContent):-
    ?User:User[
        timeline -> ?UserTimeline
    ],

• We will filter it for a viewer depending on the relationship with the current user.
Knowledge Representation and Reasoning with Flora-2

\[
\text{filter_only_family(?User,?User2,?ListContent):-}
\text{\hspace{1cm} ?User:User[}
\text{\hspace{1cm} relationship -> ?Relationship },
\text{\hspace{1cm} (?Relationship[type->spouse,person->?User2];}
\text{\hspace{1cm} ?Relationship[type->child, person->?User2];}
\text{\hspace{1cm} ?Relationship[type->parent, person->?User2];}
\text{\hspace{1cm} ?Relationship[type->grandparent,person->?User2]);}
\text{\hspace{1cm} ?ListContent = setof{ ?Content |}
\text{\hspace{1cm} ?UserTimeline[ content -> ?Content[}
\text{\hspace{1cm} audience -> Family ] ] }].
\]
filter_only_friends(\(?User, \?User2, \?ListContent) :-
  \?User:User[
    relationship -> \?Relationship
  ],
  \?Relationship[type->friend,person->\?User2],
  \?ListContent = setof{ \?Content |
    \?UserTimeline[ content -> \?Content[
        audience -> Friends ] ] }.

Knowledge Representation and Reasoning with Flora-2

filter_only_friends_of_friends ...