Knowledge Representation Using Logtalk Parametric Objects

Paulo Moura
Dep. of Computer Science, Univ. of Beira Interior, Portugal
Center for Research in Advanced Computing Systems
INESC Porto, Portugal

Someone said I needed an outline...

Spoilers

- Parametric objects
- Accessing object parameters
- Parameter passing
- **Programming patterns**
- **Parametric object proxies**
- Using both object proxies and regular objects

A warning...

- Your chairs are electrified and I have the switch... Beware of asking embarrassing questions!
- But please feel free to interrupt and ask nice ones that make the speaker shine!

Parametric objects

```
:- object(point(_X, _Y)).

    :- public(distance/1).

    distance(Distance) :-
        parameter(1, X),
        parameter(2, Y),
        Distance is sqrt(X*X + Y*Y).

:- end_object.
```

```
| ?- point(3.0, 4.0)::distance(Distance).

Distance = 5.0
yes
```

Parametric objects

```
:- object(ellipse(_Rx, _Ry, _Color)).

     :- public(area/1).

     area(Area) :-
        this(ellipse(Rx, Ry, _)),
        Area is Rx*Ry*pi.

:- end_object.
```

Parameter passing

```
:- object(ellipse(_Rx, _Ry, _Color)).
     ...
:- end_object.


:- object(circle(Radius, Color),
     extends(ellipse(Radius, Radius, Color))).
     ...
:- end_object.


:- object(red_circle(Radius),
     extends(circle(Radius, red))).
     ...
:- end_object.
```

```
| ?- red_circle(3.0)::area(Area).

Area = 28.274334
yes
```

Parametric objects are found in several systems

- Markus Fromherz's OL(P)
- Francis G. McGabe's L&O
- SICStus Objects
- Contextual Logic Programming (parametric units)
- ...

Programming patterns

Encapsulate related predicates

- anytime you have a set of predicates for reasoning about compound terms sharing the same functor and arity...
- ... just define a parametric object whose identifier unifies with the compound terms

Symplifying object interfaces

- by moving core object properties from predicate arguments to object parameters
- by making core properties visible without requiring the definition of accessor predicates

Symplifying object interfaces

```
:- object(rectangle(_Width, _Height)).

    :- public(area/1).

    area(Area) :-
        this(rectangle(Width, Height)),
        Area is Width*Height.

    :- public(perimeter/1).

    perimeter(Perimeter) :-
        this(rectangle(Width, Height)),
        Perimeter is 2*(Width + Height).

:- end_object.
```

Data-centric programming

- by representing data as *instantiations* of parametric object identifiers
- by sending messages to data instead of passing data as predicate arguments

Data-centric programming

```
% differentiate and then simplify the expression
% 2x^3 + x^2 - 4x

| ?- (2*x**3 + x**2 - 4*x)::diff(D), D::simplify(S).

D = 2*(3*x**2*1)+2*x**1*1-4*1
S = 2*(3*x**2)+2*x-4
yes
```

Restoring shared constraint variables

- by using object parameters as constraint variables
- by using parametric object identifiers to link constraint variables

Restoring shared constraint variables

```
:- protocol(process_description).

    :- public(domain/2).
    :- mode(domain(-list(object_identifier), -callable), one).
    :- info(domain/2, [
        comment is 'Process dependencies and constraints.',
        argnames is ['Dependencies', 'Constraints']]).

 :- end_ protocol.


:- object(a(_),
    implements(process_description)).

    domain([], (A #>= 2, A #=< 4)) :-
        parameter(1, A).

:- end_object.
```

Restoring shared constraint variables

```
:- object(b(_),
    implements(process_description)).

    domain([a(A)], (B #>= A, B #=< 3)) :-
        parameter(1, B).

:- end_object.


:- object(c(_),
    implements(process_description)).

    domain([a(A), b(B)], (C #= B + 1, C #= A + 1)) :-
        parameter(1, C).

:- end_object.
```

Restoring shared constraint variables

```
| ?- process_model::solve([c(C)], Dependencies).

C = _#59(3..4)
Dependencies = [b(_#21(2..3)),a(_#2(2..3))]
yes
```

Logical updates of object state

- by using one or more object parameters as an alternative to the database to represent dynamic state
- by using object parameters to keep a backtracable history of state changes

```
:- object(rectangle(_Width, _Height, _Position),
    imports(private::assignvars)).

    :- public([init/0, area/1, move/2, position/2]).

    init :-
        parameter(3, Position),
        ::assignable(Position, (0, 0)).

    area(Area) :-
        this(rectangle(Width, Height, _)),
        Area is Width*Height.

    move(X, Y) :-
        parameter(3, Position), ::Position <= (X, Y).

    position(X, Y) :-
        parameter(3, Position), ::Position => (X, Y).

:- end_object.
```

Logical updates of object state

```
| ?- rectangle(2, 3, _)::(
        init,
        area(Area), position(X0, Y0),
        move(1, 1), position(X1, Y1),
        move(2, 2), position(X2, Y2)
        ).

Area = 6
X0 = Y0 = 0
X1 = Y1 = 1
X2 = Y2 = 2
yes
```

Parametric object proxies

Parametric object proxies

- compound terms with the same functor and arity as a parametric object identifier
- sored in the database as Prolog facts
- used to represent different *instantiations* of a parametric object identifier

Parametric object proxies

```
% circle(Id, Radius, Color)

circle('#1', 1.23, blue).
circle('#2', 3.71, yellow).
circle('#3', 0.39, green).
circle('#4', 5.74, black).
```

```
:- object(circle(_Id, _Radius, _Color)).

    :- public([id/1, radius/1, color/1, area/1, perimeter/1]).

    id(Id) :-
        parameter(1, Id).

    radius(Radius) :-
        parameter(2, Radius).

    color(Color) :-
        parameter(3, Color).

    area(Area) :-
        parameter(2, Radius),
        Area is 3.1415927*Radius*Radius.

    perimeter(Perimeter) :-
        parameter(2, Radius),
        Perimeter is 2*3.1415927*Radius.

:- end_object.
```

Parametric object proxies

```
| ?- findall(Area, {circle(_, _, _)}::area(Area), Areas).

Areas = [4.75291, 43.2412, 0.477836, 103.508]
yes
```

Parametric object proxies

- useful for representing large number of immutable data objects...
- ... while providing a compact representation
- application logic represented by a set of hierarchies with the data objects as leaves...
- ... taking full advantage of Logtalk object-oriented features for representing and reasoning with taxonomic knowledge

Using both object proxies and regular objects

- parametric object proxies provide a compact representation but...
- ... complex domain objects are better represented as regular objects

Using both object proxies and regular objects

- the meaning of an object proxy argument is implicitly defined by its position in the compound term...
- ... while regular objects use predicates with meaningful names

Using both object proxies and regular objects

```
% fa(Id, InitialState, Transitions, FinalStates)

fa(fa1, 1, [1/a/1, 1/a/2, 1/b/2, 2/b/2, 2/b/1], [2]).
```

```
:- object(aibiciTM,
    instantiates(tm)).

    initial(q0).

    transitions([
        q0/'B'/'B'/'R'/q1,
        q1/a/'X'/'R'/q2,   q1/'Y'/'Y'/'R'/q5,  q1/'B'/'B'/'R'/q6,
        q2/a/a/'R'/q2,     q2/'Y'/'Y'/'R'/q2,  q2/b/'Y'/'R'/q3,
        q3/b/b/'R'/q3,     q3/'Z'/'Z'/'R'/q3,  q3/c/'Z'/'L'/q4,
        q4/a/a/'L'/q4,     q4/b/b/'L'/q4,      q4/'Y'/'Y'/'L'/q4,
        q4/'Z'/'Z'/'L'/q4, q4/'X'/'X'/'R'/q1,
        q5/'Y'/'Y'/'R'/q5, q5/'Z'/'Z'/'R'/q5,  q5/'B'/'B'/'R'/q6
    ]).

    finals([q6]).

:- end_object.
```

Using both object proxies and regular objects

```
:- object(fa(_Id, _Initial, _Transitions, _Finals),
    instantiates(fa)).

    initial(Initial) :-
        parameter(2, Initial).

    transitions(Transitions) :-
        parameter(3, Transitions).

    finals(Finals) :-
        parameter(4, Finals).

:- end_object.
```

That’s all folks!

Please don’t forget to buy the nice t-shirt!

Ive got you under my skin
Ive got you deep in the heart of me
So deep in my heart, that youre really a part of me
Ive got you under my skin

Ive tried so not to give in
Ive said to myself this affair never will go so well
But why should I try to resist, when baby will I know than well
That Ive got you under my skin

That's all folks!

Id sacrifice anything come what might
For the sake of having you near
In spite of a warning voice that comes in the night
And repeats, repeats in my ear

Dont you know you fool, you never can win
Use your mentality, wake up to reality
But each time I do, just the thought of you
Makes me stop before I begin
Cause Ive got you under my skin

Please don't forget to buy the nice t-shirt!