Parametric objects programming idioms

Paulo Moura
Logtalk author and maintainer

https://logtalk.org/
pmoura@logtalk.org

Parametric objects

- Identifiers are compound terms
- All identifier arguments are variables
- Identifier variables interpreted as object *parameters*
- Parameters are *logical variables* shared by all object predicates and directives

Parameter access

- Write parameter variables using the syntax `_Name_` (allows abstracting parameter position, simplifying maintenance)
- Use the `parameter/2` built-in execution context method (inherited from Logtalk 2.x; uses parameter position)
- Use the `this/1` built-in execution context method (useful to retrieve all parameters at once)

A simple example

```
:- object(rectangle(_Width_, _Height_)).

    :- public([
        width /1, height/1, area/1, perimeter/1
    ]).

    width(_Width_).

    height(_Height_).

    area(Area) :-
        Area is _Width_ * _Height_.

    perimeter(Parimeter) :-
        Perimeter is 2 * (_Width_ + _Height_).

:- end_object.
```

Parameter binding

- When sending a message (parameters are logical variables)
- When declaring entity relations in object (or category) opening directives (allows defining default bindings by extending a parametric object)

Parameter binding

```
| ?- rectangle(3, 4)::area(Area).
Area = 12
yes
```

```
:- object(square(Side),
    extends(rectangle(Side, Side)).
    ...
:- end_object.
```

```
:- object(unit_square,
    extends(square(1)).
    ...
:- end_object.
```

Parametric object proxies

- Any term subsumed by a parametric object identifier can be used as a message receiver
- Parametric object identifiers can be defined as predicates
- Any such predicate clause is a *parametric object proxy*
- Dedicated syntax to use parametric object proxies: `{Proxy}::Message` calls `Proxy` in `user` and sends `Message` to the resulting bindings

Parametric object proxies

```
% facts as parametric object proxies
rectangle(1, 2).
rectangle(2, 3).
rectangle(3, 4).
...


| ?- findall(Area, {rectangle(_, _)}::area(Area), Areas).
Areas = [2, 6, 12]
yes
```

Choosing object parameters

- Parameters should be meaningful for most object predicates
- Parameters are often core properties of what the object represents
- Parameters may also configure object semantics

Parameters can represent

- Types
- Core properties
- Logical state
- Operations
- Constraints
- Anything a term can be used for!

Programming idioms

- Delegating operations
- Simplifying object protocols
- Data-centric programming
- Restoring shared constraint variables
- Representing logical state
- Enabling network modeling