

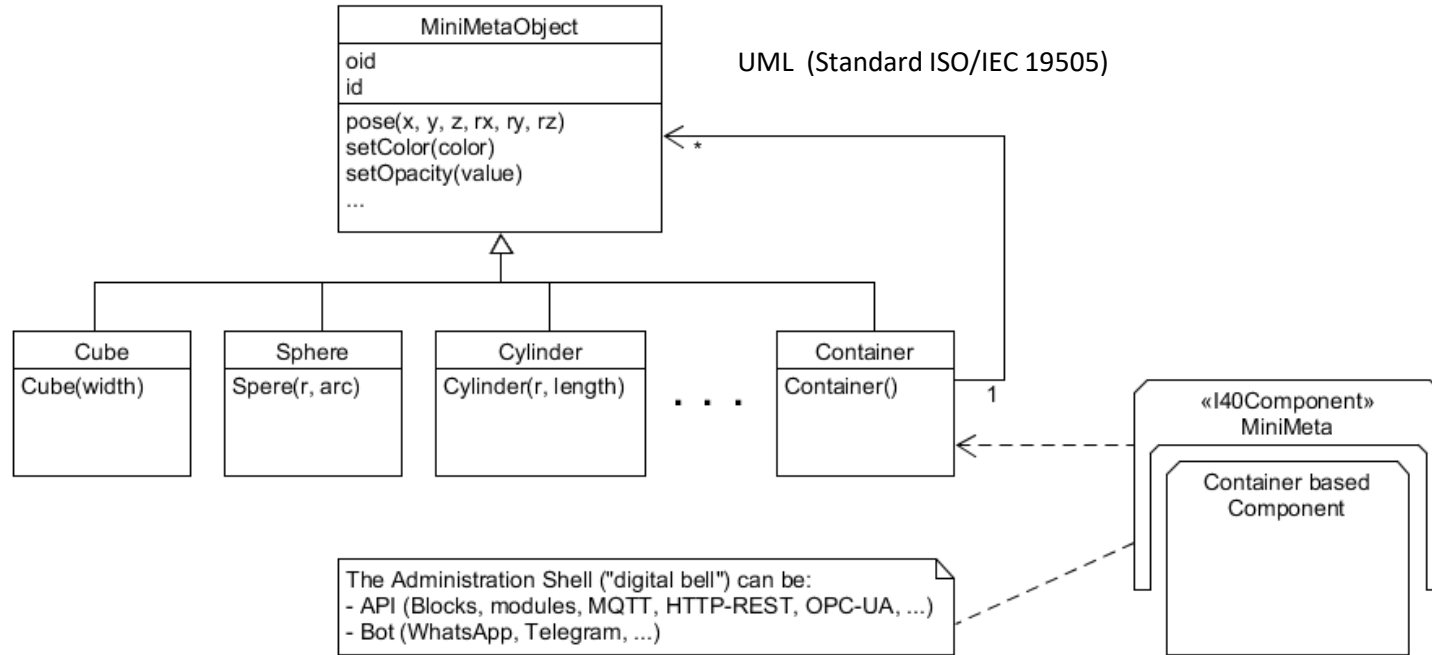
A minimal 4.0 Metaverse controlled with Snap!

Xavier Pi

Snap!Con 2022



The notion of I4.0 Component



Applied Design Patterns:

- Composite (Container)
- Proxy (JavaScript Three.js functions call from Snap! via MQTT)
- Façade (Table, Conveyor, ...)

RAMI 4.0 (Standard IEC PAS 63088)

A basic element: Cube

- The constructor:

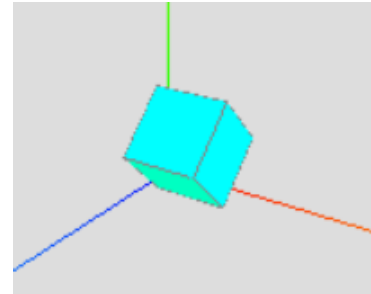
```
new cube width 1 with id c01
```

- Operations (methods):

```
pose id c01 to x 1 y 1 z 1 rx 45 ry ry rz rz
```

```
set color cyan to id c01
```

```
pose id c01 to x 1 y 1 z 1 rx 10++ ry ry rz rz
```



Object orientation without classes, according to Bertrand Meyer's ("Object-Oriented Software Construction" – 1997) approach (adding the object id as a calling parameter)

Container as a basis for Components

show axes of id with length 5

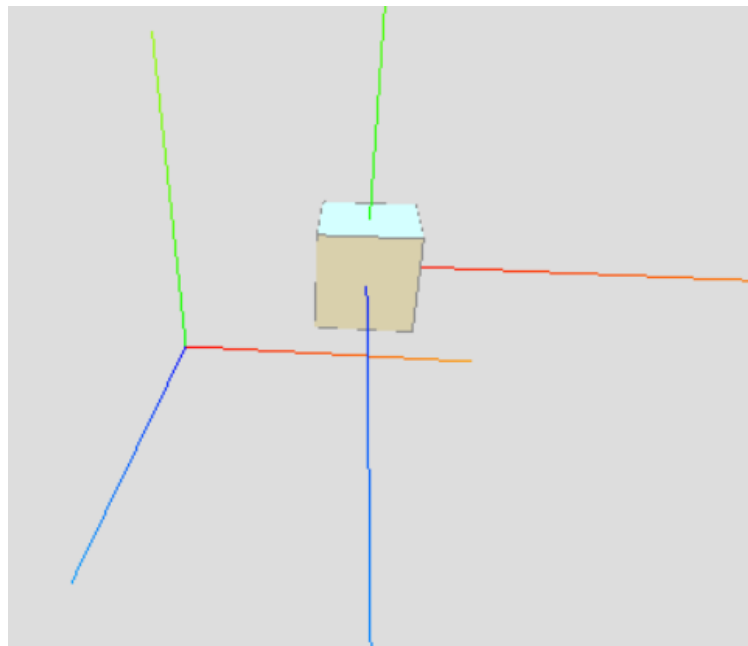
new container with id p01

show axes of id p01 with length 5

pose id p01 to x 3 y 3 z 3 rx ry rz

new cube width 1 with id c01

add id c01 to id p01



Objects naming issues

- Absolute naming vs relative naming

(John Postel's RFC 819 "The Domain Naming Convention for Internet User Applications " 1982)

- With relative names, the object name (id) changes depending on the container where it is.

add id p01;c01 to id scene

Back to the scene with relative name (id)

get oid from id c01

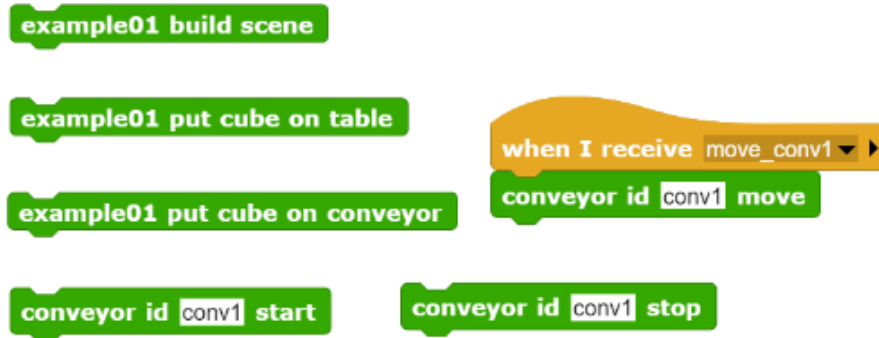
oid17

add id oid17 to id scene

Back to the scene with absolute name (oid)

Table-Box-Conveyor Example

- <https://pixavier.github.io/meta4snap/>
- Inspired by Beetle Blocks (www.beetleblocks.com), we can program animations



Some features to add

- Improve collision detection
- Add a measurement tools (distance, angles)
- Improve external 3D elements loading
- Better control of the camera and movement
- Add light and luminosity control

Thank you !

Xavier Pi

xpi@enginyers.net

<https://github.com/pixavier>