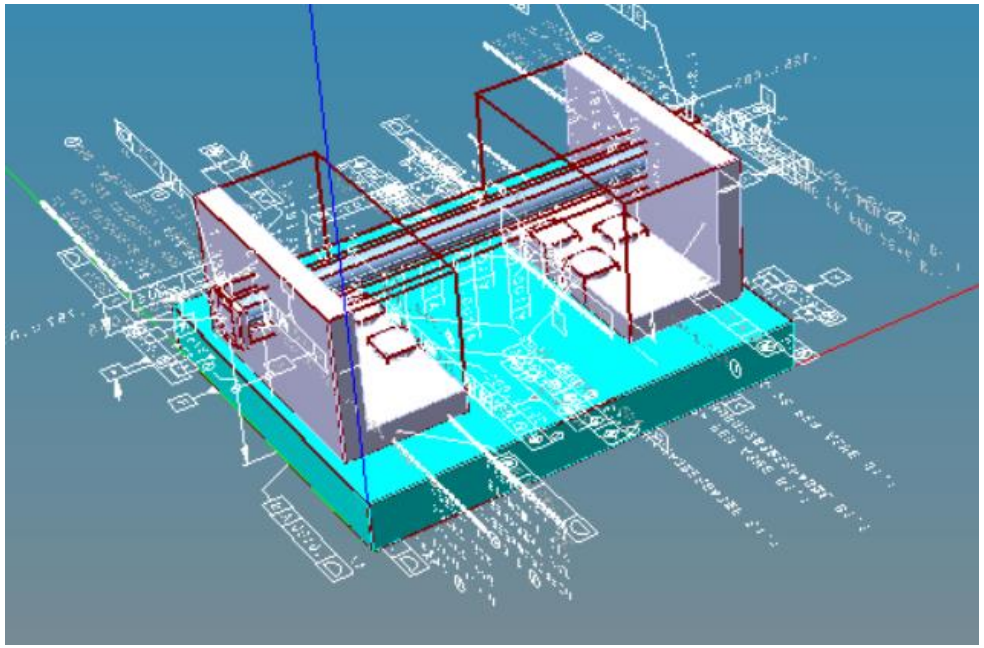
Digital Twins and the Digital Thread

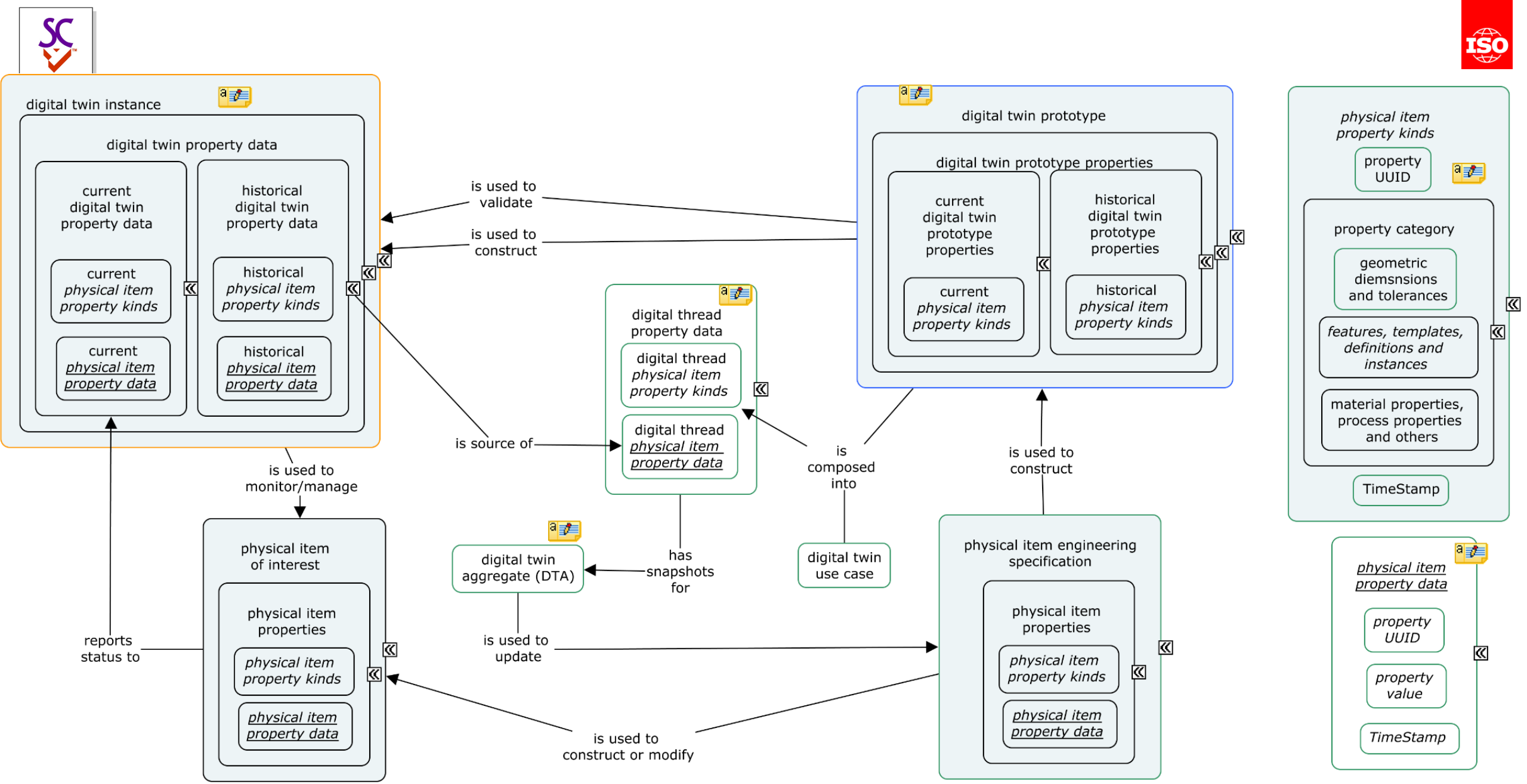
UUID meeting 10/05/2022

Martin Hardwick

Convenor WG15



AS1 Prototype


digital twin instance
digital twin property data
current digital twin property data
historical digital twin property data
current physical item property kinds
historical physical item property kinds
current physical item property data
historical physical item property data
is used to validate
is used to construct
digital twin prototype
digital twin prototype properties
current digital twin prototype properties
historical digital twin prototype properties
current physical item property kinds
historical physical item property kinds
physical item property kinds
property UUID
property category
geometric diemsnsions and tolerances
features, templates, definitions and instances
material properties, process properties and others
TimeStamp
digital thread property data
digital thread physical item property kinds
digital thread physical item property data
is source of
is composed into
is used to construct
is used to monitor/manage
physical item of interest
physical item properties
physical item property kinds
physical item property data
digital twin aggregate (DTA)
has snapshots for
digital twin use case
physical item engineering specification
physical item properties
physical item property kinds
physical item property data
reports status to
is used to update
is used to construct or modify
physical item property data
property UUID
property value
TimeStamp

Prototypes and Instances

Fastener Prototype UUID's

Fastener Instance UUID's

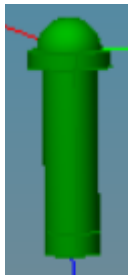
Hole Instance UUID's

Hole Prototype UUID's

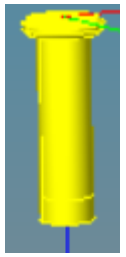
ACME Prototype

Bolts R Us Prototype

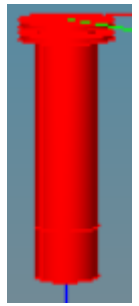
SGR Prototype



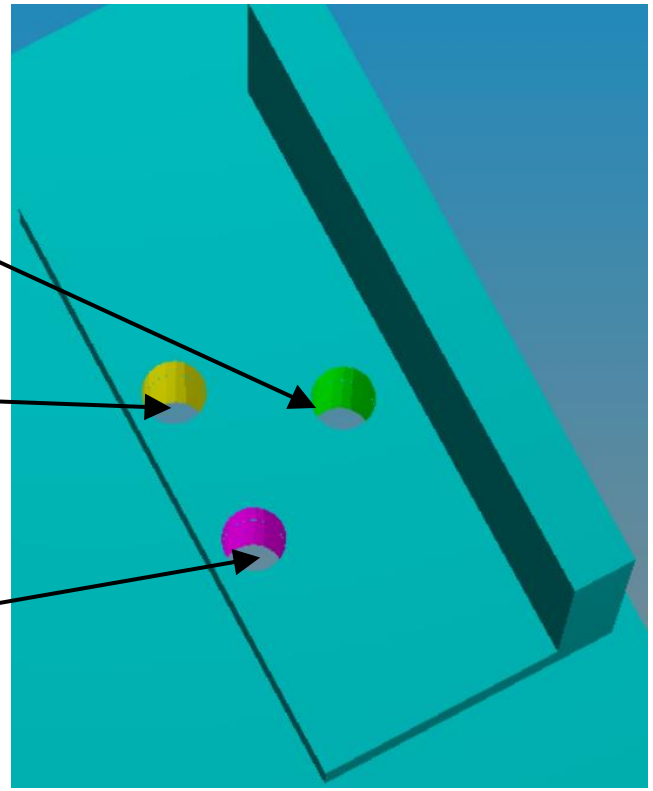
ACME instance



Bolts R Us instance



SGR instance



AS1 Instance

AS1 Prototype

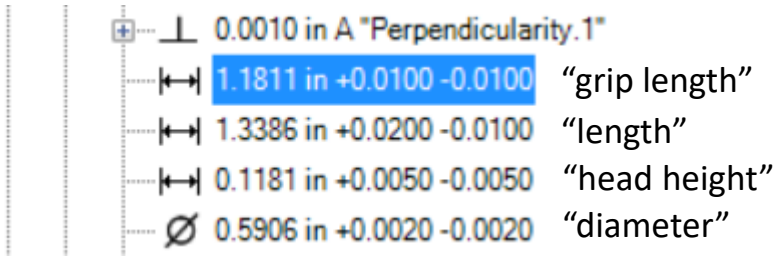
What gets a UUID

- On Digital Twin Prototype
  - UUID for each part number (in case there are duplicates across suppliers)
  - UUID for each part usage in an assembly (gives us part and location)
  - UUID for each process step in a manufacturing solution
  - UUID for each engineering requirement (lots of these)

- On Digital Twin Instances
  - UUID for serial number (in case there are duplicates across suppliers)

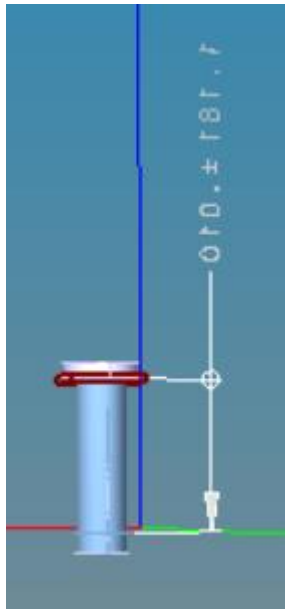
Engineering Requirement UUIDs

- UUID registration can enable flexible naming and representation
  - E.g “Grip length” means `e00a534e-4a44-4fd1-a530-e4f0abc34037`



“grip length”
“length”
“head height”
“diameter”

Design



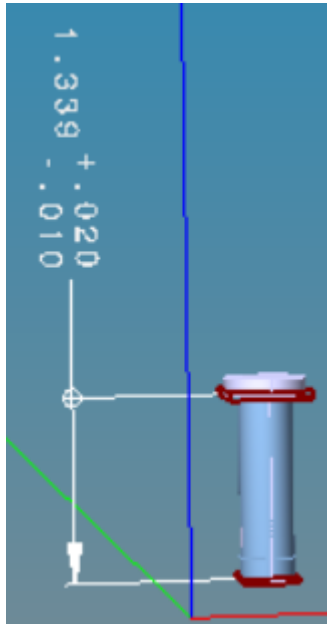
Drill hole at location = “fastener.1”
depth=“grip length”
diameter=“diameter”

Manufacturing

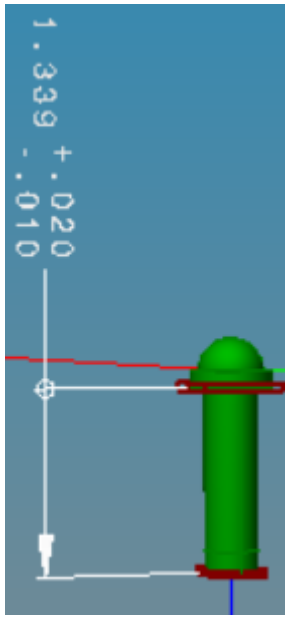
Checking of Prototypes

- Good – manufacturing dimensions identical to engineering
- Bad – manufacturing dimensions different but instance maybe OK
- Ugly – manufacturing dimensions mean instance can never be OK

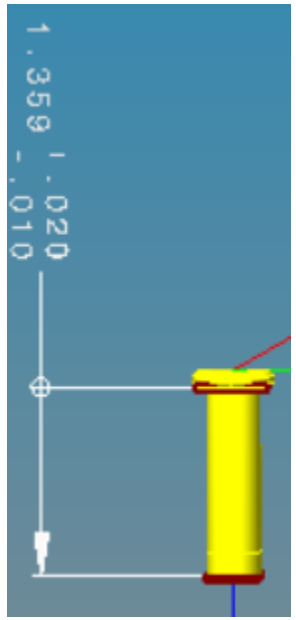
Engineering Prototype



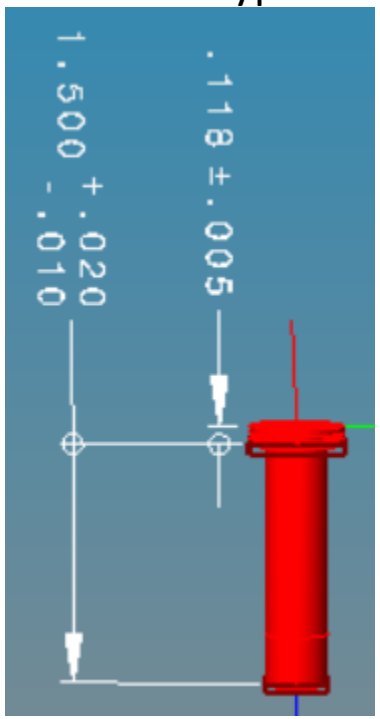
Good Supplier Prototype



Bad Supplier Prototype



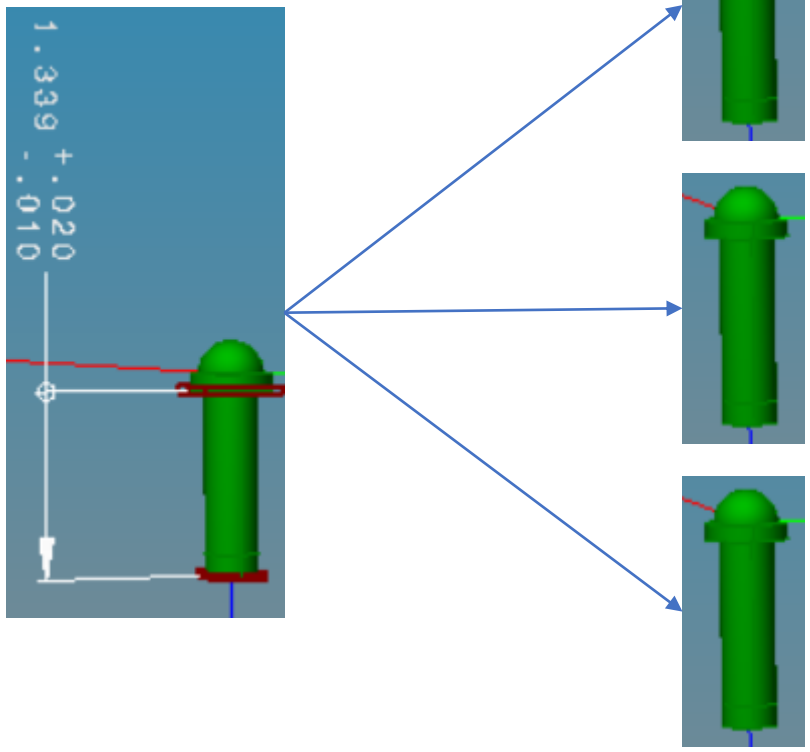
Ugly Supplier Prototype

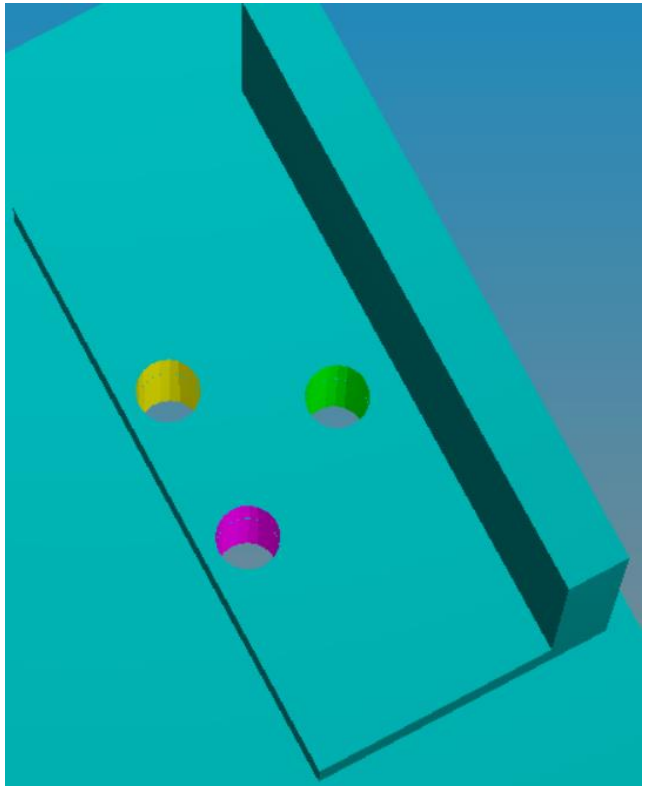


Checking of Instances

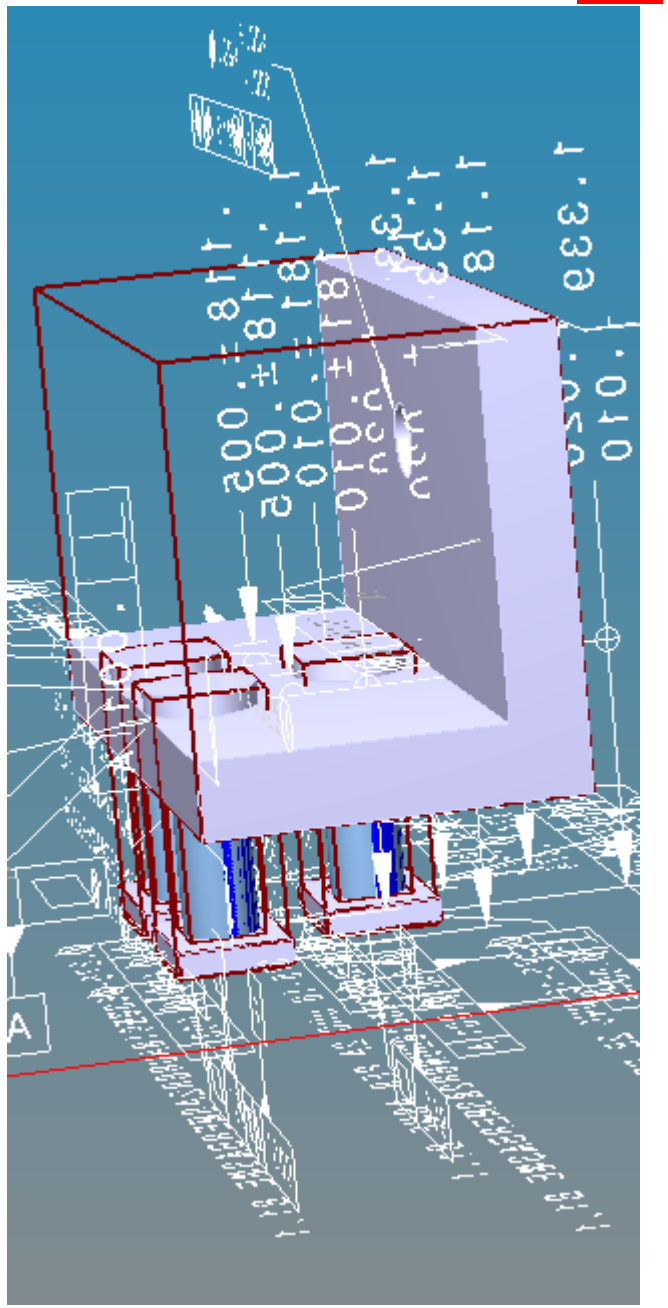
Good Instances ← ? → Hole Instances

Good Prototype





AS1 Prototype



Hole depth options

- Depth of hole compared with grip length of fastener
- How the hole depth might be defined
  - Linear distance dimension
  - Depth parameter on a design feature
  - Linear path on a manufacturing feature
  - Depth on a drilling operation
  - Distance between trim planes on a cylinder
  - Value in a magic string

`d84ab9c3-7112-4356-b529-723683bacc1c` – means “hole depth”

`e00a534e-4a44-4fd1-a530-e4f0abc34037` – means “grip length”

Levels of UUID

- Level 1 – UUID for each class of prototype definition
  - UUID that means grip length, UUID that means hole depth
  - UUID that means fastener, UUID that means L-bracket
- Level 2 – UUID for each prototype definition
  - UUID on a dimension, UUID on a parameter, UUID on a geometry item
  - UUID on a product definition, UUID on a feature definition
- Level 3 – UUID for each instance
  - UUID on a serial number
  - UUID on a data value

Questions

- What levels should be supported
  - Level 1 is most specific, Level 3 is most general
  - Level 1 has most reuse, Level 3 has most generation
- What rules for representing the data of a Level 1 definition
  - What entities can represent a grip length?
  - What entities can represent a hole depth?
  - What products can represent a fastener?
  - What features can represent a hole?
  - How to describe and enforce these rules