Digital Thread Proposal

Martin Hardwick

Convenor

ISO TC184/SC4 WG15 Digital Manufacturing

Why digital thread

- Requirements are met at multiple stages of the design to manufacturing pipeline

- Linking requirements to solutions is difficult because of islands (or stovepipes) of automation in which design uses one set of tools, engineering another set, and manufacturing yet another

- We propose to link the islands to create a digital thread that can be used to trace and maintain solutions to requirements.

- The proposed thread uses decentralized identifiers (**DiDs**) to link the islands

- The proposed thread uses a **triple store** to trace the requirements
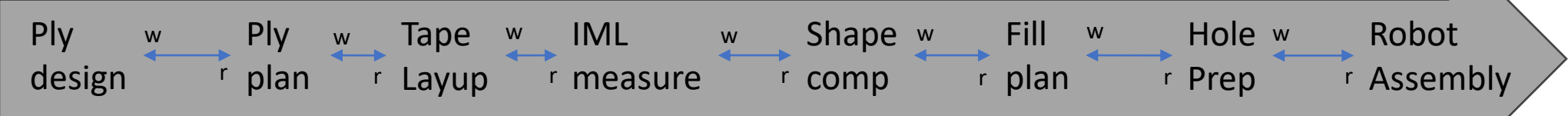
Example – tracing requirements

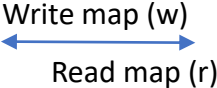Multiple design disciplines

Multiple manufacturing solutions

"How my requirements were met"

Thread Query

Requirements

End to end identification

| Ply design | w / r | Ply plan | w / r | Tape Layup | w / r | IML measure | w / r | Shape comp | w / r | Fill plan | w / r | Hole Prep | w / r | Robot Assembly |

Completeness Analytics

Forward flow

Stability in references

Write map (w)
Read map (r)

Thread Message

"How my design was used"

ISO 23247-5 Digital Thread for digital twin

Model Content in STEP-NC

Read map | Write map

Id1
Id2
Id3
Id4
Id5
id6

UUIDa
UUIDb

UUIDx
UUIDy
UUIDz

Read map | **Write map**

Id10
Id20
Id30
Id40

UUIDx
UUIDp
UUIDpp
UUIDppp

Model Content In MTC

Read map | **Write map**

Id100
Id200
Id300
Id400

UUIDy
UUIDq
UUIDqq

Model Content In QIF
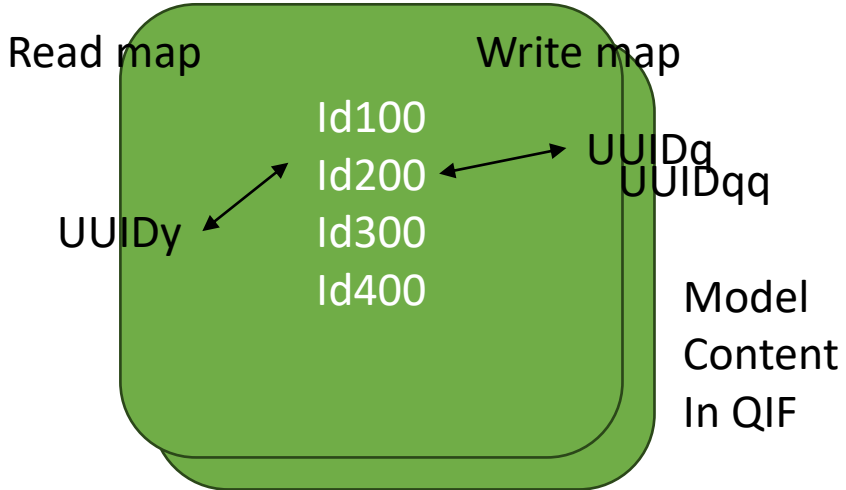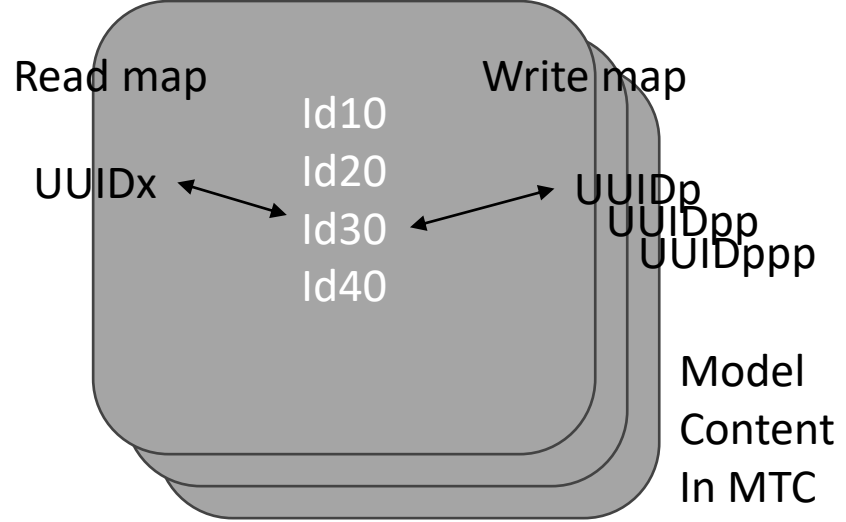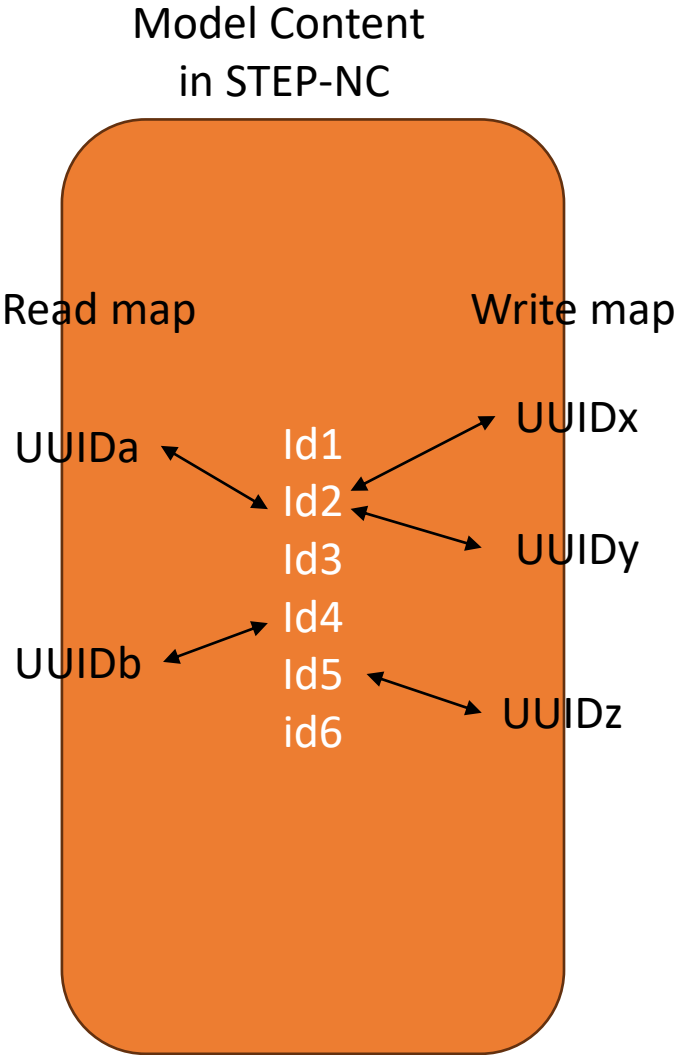
did:thread:step:
position tolerance:UUIDa
machining workingstep:UUIDx
measurement workingstep:UUIDy

did:thread:mtc:
feature twin:UUIDp
feature twin:UUIDpp
feature twin:UUIDppp

did:thread:qif:
measurement twin:UUIDq
measurement twin:UUIDqq

Triples
Makes (UUIDa, UUIDx)
Instance(UUIDx, UUIDp)
Instance(UUIDx, UUIDpp)
Instance(UUIDx, UUIDppp)
measures(UUIDa, UUIDy)
Instance(UUIDy, UUIDq)
Instance(UUIDy, UUIDqq)
Setup(UUIDb, UUIDz)

ISO 23247-6 Digital Twin Composition

Read/write maps for internal and external identifiers as design components are merged to make manufacturing solutions
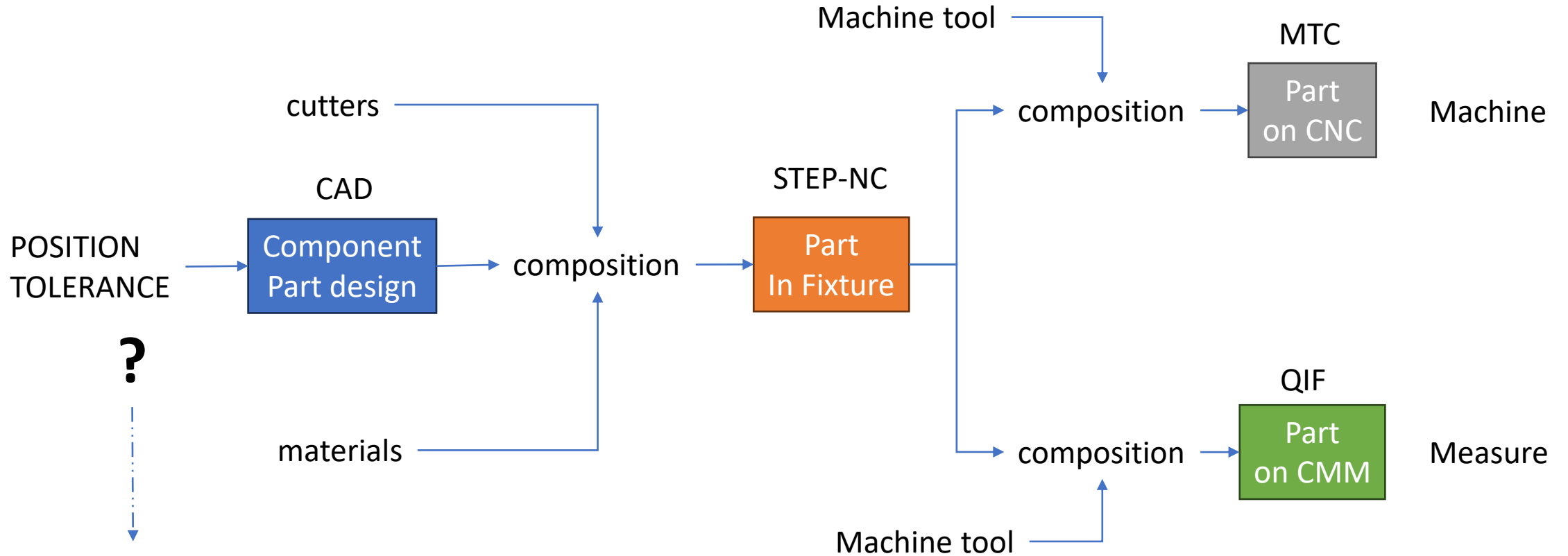
Key data requirements

LOTAR

Design

LOTAR
Mutability

Manufacturing

LOTAR
Mutability
Traceability

Maintenance

LOTAR
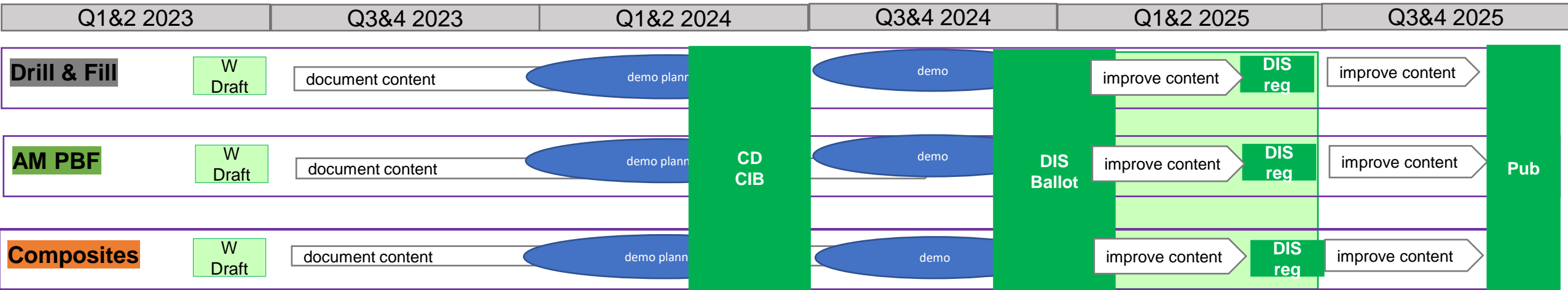
Design

LOTAR
Mutability

Build

LOTAR
Mutability
Traceability

Maintain

Toolkits

Level 1 | Part 21 Parser | Read data files, looks for keywords, resolves numbers, checks syntax

Level 2 | EXPRESS Parser | Make class libraries, populate objects from files, check conformance

Level 3 | STEP Modeler | Simulate manufacturing operations, mutate objects, validate results

AP238 E4 – Model Based Manufacturing