



University of Stuttgart
Institute of Industrial Automation
and Software Engineering



1 modular production facility
2 Development of a digital twin with a semantic query
3 and command interface for a

Conception Project

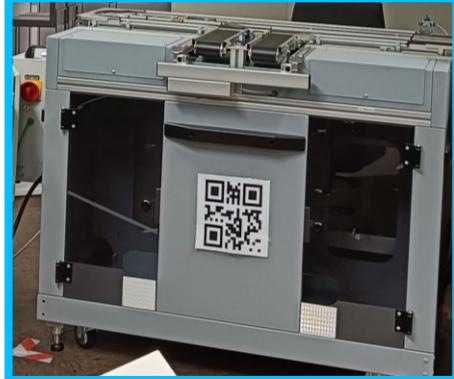


Motivation

- CP factory
 - Proprietary systems from Festo
 1. MES
 2. Fleet Manager
 - Problems:
 - Modularity
 - Interoperability
 - Difficulty to maintain and operate

Not flexible ;
difficulties in
reconfiguration

CP Branch



CP Storage



Robotino v3



Solution

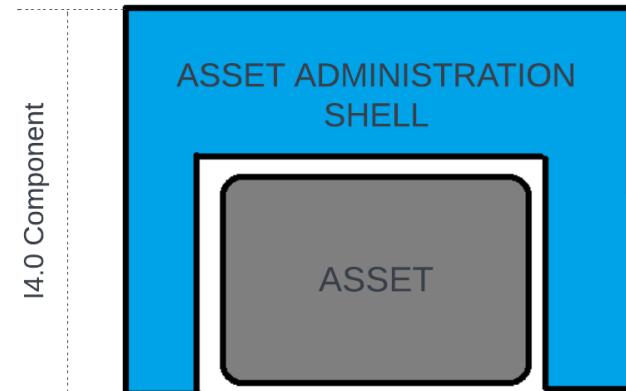


- Need for provision of harmonised interface using “Digital Twin” (digitizing industrial production)
- To achieve intuitive HMI, allowing workers to operate the system just using Natural Language
 - Semantics
 - Neural Language Model for language understanding

Basics

Digital Twin- Asset Administration Shell (AAS)

- What is Digital Twin (DT)? Generally, a Digital Twin (DT) is a “*virtual representation of a physical asset in a CPS, capable of mirroring its static and dynamic characteristics*” [1]
- AAS is a implementation for Digital Twin
 - Advantages & reasons:
 1. Semantically described information
 2. Promotes standardized DT to ensure interoperability
 3. It has good ready-to-use middleware



AAS Middleware

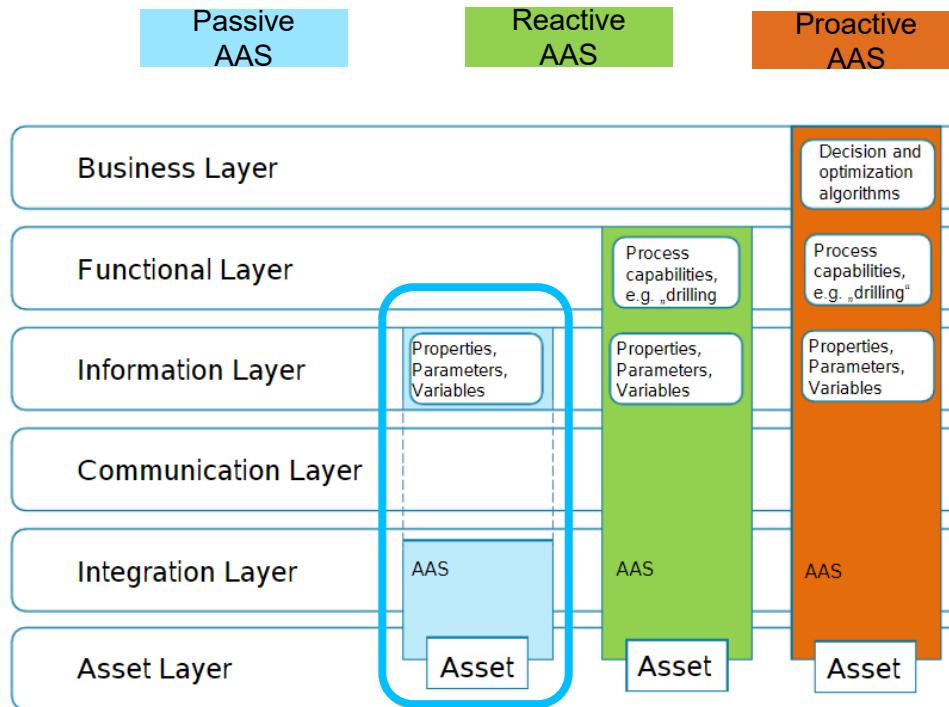
Evaluation

Criteria	1.2.0	BaSyX	0.2.1	FA ³ ST
Asset connections possibility (incl. with tool)	o		+	
AASX file (static data upload)	+		o	
Data query (Runtime) (REST API)	+		-	
Configurations (AAS Registry, Database)	+		- (only file-based, memory)	
Deployment (docker available)	+		+	
Code maturity	+		- (still under development)	
Documentation	o [1]		o [2]	
AAS metadata template	o [v2.0.1]		+ [v3.0RC01]	

+: fulfilled o: conditionally fulfilled -: not fulfilled

Reference Architecture Model for Industry 4.0 (RAMI 4.0)

For Passive AAS

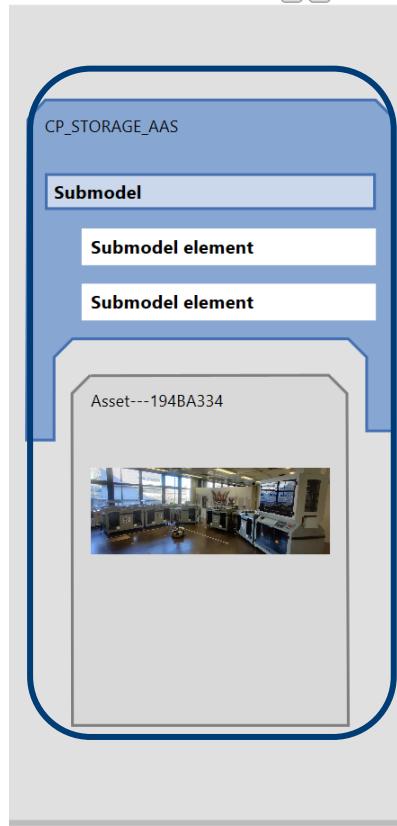


[4] adapted from

AAS at design time

Passive AAS

AASX Package Explorer - local file: D:\RP_IAS_softwares\iasx-package-explorer.2022-08-06\basyx_server\demo-final.iasx buffered to: C:\Users\manth\AppData\Local\Temp\tmp_5A13.iasx
File Workspace Options Help



AAS "CP_STORAGE_AAS" [Custom, CP_STORAGE_AAS] of [Custom, Asset---194BA334, Instance]

- SM "TechnicalData" [IRI, https://example.com/ids/sm/7405_7180_0122_9590]
- SM "OperationalData" [IRI, https://example.com/ids/sm/1500_1101_1122_6481]
- SM "Documentation" [IRI, https://example.com/ids/sm/6414_1271_0122_6904]

AAS "CP_BRANCH_2_AAS" [Custom, CP_BRANCH_2_AAS] of [Custom, Asset---0BDE084A, Instance]

- SM "TechnicalData" [IRI, https://example.com/ids/sm/0100_8180_0122_5267]
- SM "OperationalData" [IRI, https://example.com/ids/sm/2210_1101_1122_7473]
- SM "Documentation" [IRI, https://example.com/ids/sm/4114_2203_0122_6509]

AAS "CP_BRANCH_5_AAS" [Custom, CP_BRANCH_5_AAS] of [Custom, Asset---4A6931EB, Instance]

- SM "TechnicalData" [IRI, https://example.com/ids/sm/5000_0213_0122_3703]
- SM "OperationalData" [IRI, https://example.com/ids/sm/8253_1201_1122_9978]
- SM "Documentation" [IRI, https://example.com/ids/sm/5491_1271_0122_6874]

AAS "Robotino_AAS" [Custom, Robotino_AAS] of [Custom, Asset---17F5613A, Instance]

- SM "TechnicalData" [IRI, https://example.com/ids/sm/2300_8180_0122_5359]
- SM "OperationalData" [IRI, https://example.com/ids/sm/7340_1101_1122_3922]
- SM "Documentation" [IRI, https://example.com/ids/sm/7264_1113_0122_9073]

AAS "FleetManager_AAS" [Custom, FleetManager_AAS] of [Custom, Asset---2EA06EF7, Instance]

- SM "Capabilities" [IRI, https://example.com/ids/sm/5150_1101_1122_6130]
- SM "Softwares" [IRI, https://example.com/ids/sm/6403_2103_1122_7108]

AAS "MES_AAS" [Custom, MES_AAS] of [Custom, Asset---3BA1D6EC, Instance]

- SM "Capabilities" [IRI, https://example.com/ids/sm/4350_1101_1122_5498]
- SM "Softwares" [IRI, https://example.com/ids/sm/0581_9110_2122_8181]

based on specifications of Platform Industrie 4.0

Element Content

Asset Administration Shell

Referable: idShort: CP_STORAGE_AAS

HasDataSpecification (Reference):

Identifiable: idType: Custom id: CP_STORAGE_AAS

Asset Reference assetRef: (Asset) (local) [Custom] Asset---194BA334 [Jump](#)

Asset

Referable: idShort: CP_STORAGE category:

HasDataSpecification (Reference):

Identifiable: idType: Custom id: Asset---194BA334

Kind (of Asset): kind: Instance

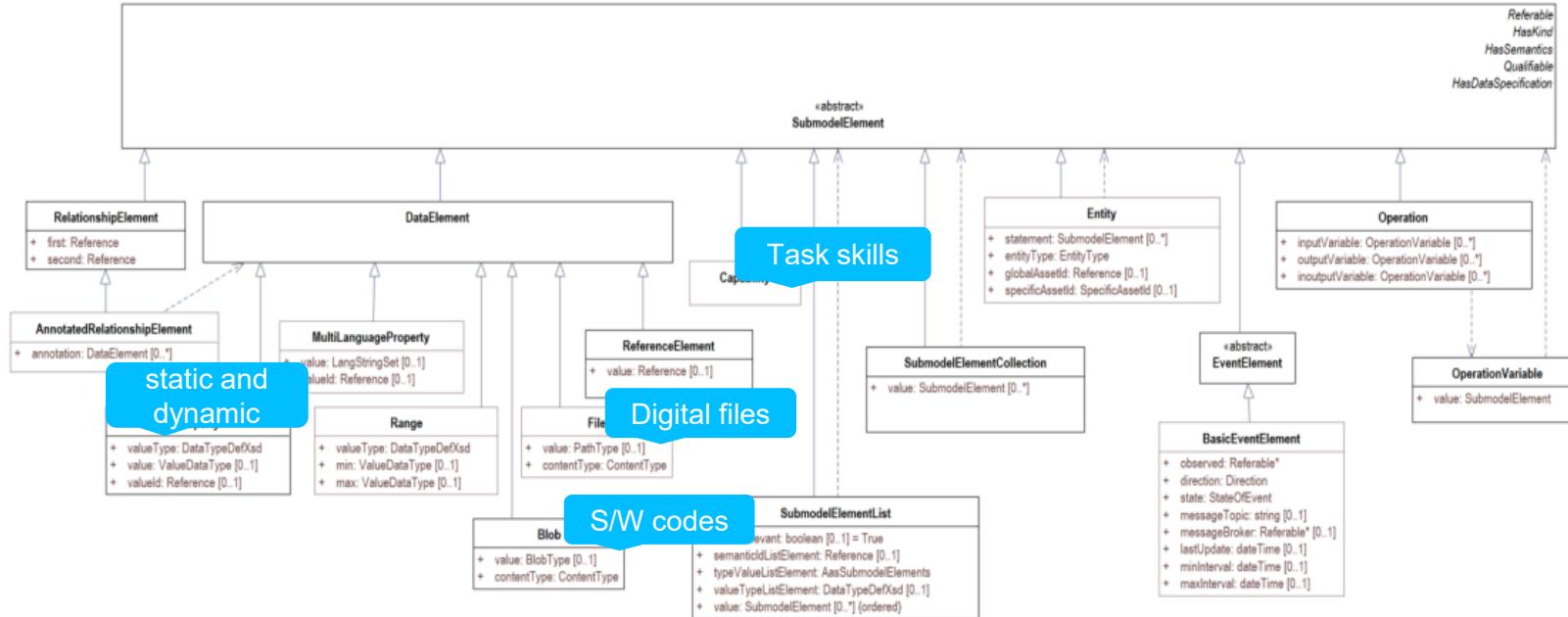
Submodel references with special meaning

Reload Drag from here! Show Content

0 bytes No errors Clear Report.. 2/12/2022 7

AAS Metamodel

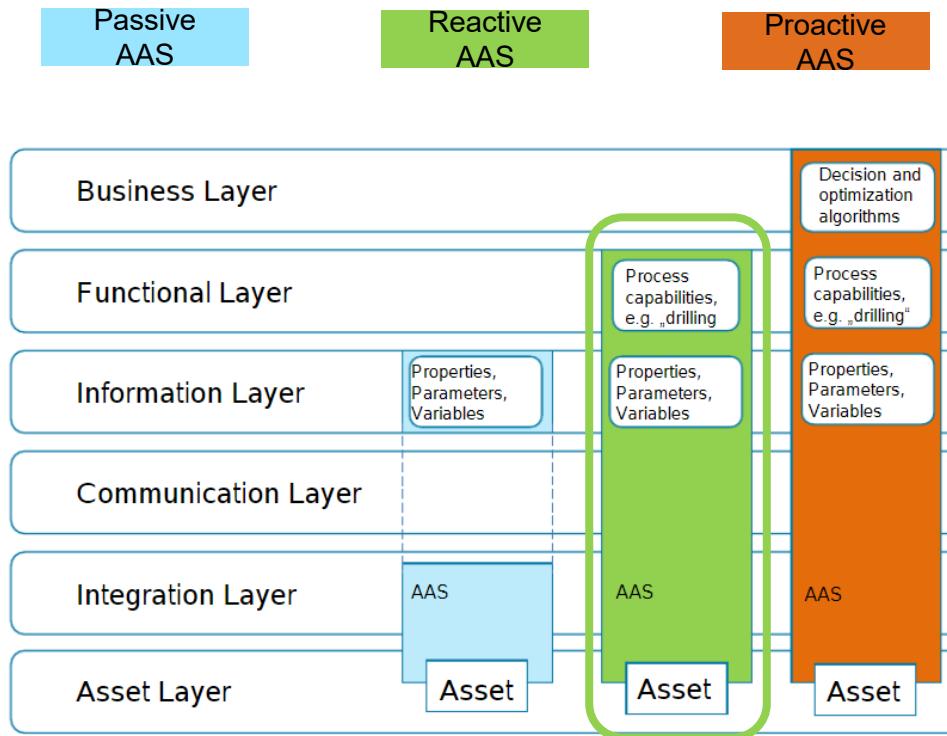
Submodel Element detailed view



[2]

Reference Architecture Model for Industry 4.0 (RAMI 4.0)

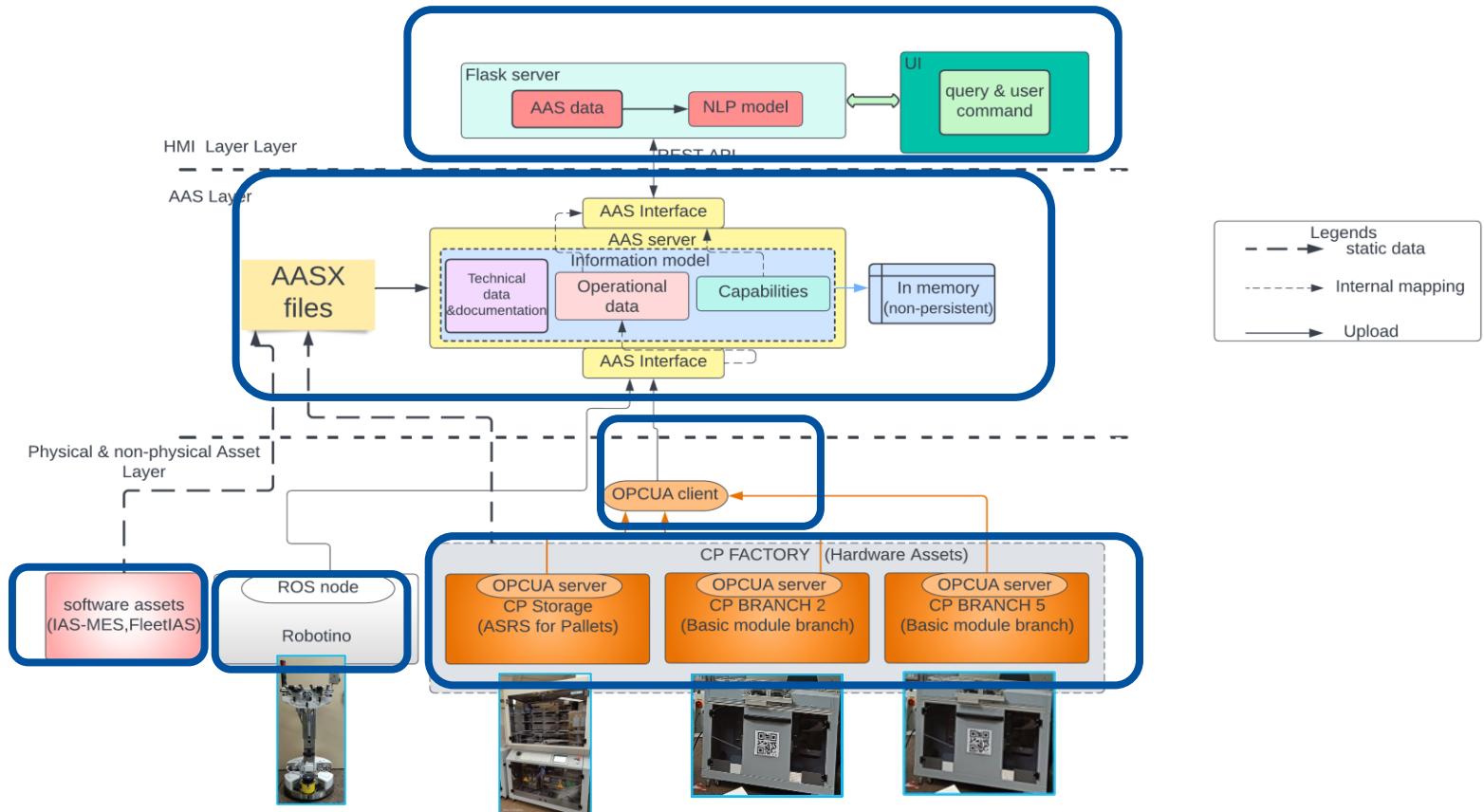
For Reactive AAS



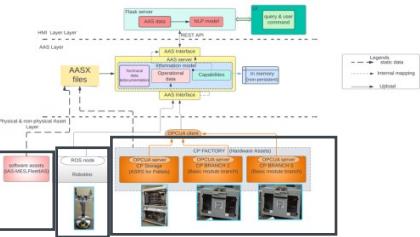
[4] adapted from

Architecture

Concept overview



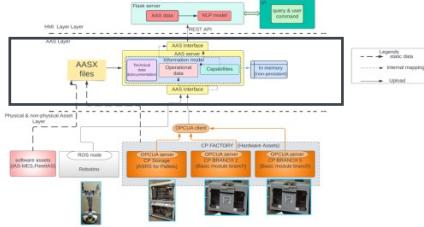
Asset connections



- Software assets IAS-MES, FleetIAS
 - Python applications added as blob file
- Command skills :
 - Modular encapsulated as task skills
 - e.g.: move_robotino_dock_2 REST API call
 - POST REST API request for assigning order
 - e.g.:

```
{  
    "name": "Test",  
    "description": "Test",  
    "assignedWorkingPlan": 2021,  
    "assignedWorkingPiece": 16,  
    "orderNo": 1,  
    "orderPos": 1,  
    "mainOrderPos": 1,  
    "customer": null,  
    "status": "[0,0,0,0,0,0]"  
}
```

AAS as the digital twin



- Contains information model
- AAS server provides harmonized interfaces
 - API (RESTful API)

PUT /shells/{aasId}/aas/submodels/{submodelIdShort}/submodel/submodelElements/{selIdShortPath}

GET /shells/{aasId}/aas/submodels/{submodelIdShort}/submodel/submodelElements/{selIdShortPath}

DELETE /shells/{aasId}/aas/submodels/{submodelIdShort}/submodel/submodelElements/{selIdShortPath}

GET /shells/{aasId}/aas/submodels/{submodelIdShort}/submodel/submodelElements/{selIdShortPath}/value

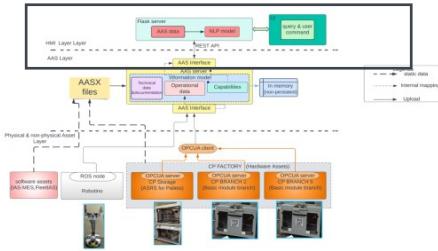
PUT /shells/{aasId}/aas/submodels/{submodelIdShort}/submodel/submodelElements/{selIdShortPath}/value

The screenshot shows a JSON object structure in a browser's developer tools:

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
0: ...
1: ...
2: ...
3: ...
4: ...
5: ...
  ▼ modelType:
    name: "AssetAdministrationShell"
    idShort: "FleetManager_AAS"
  ▼ identification:
    idType: "Custom"
    id: "FleetManager_AAS"
    dataSpecification: []
    embeddedDataSpecifications: []
  ▼ submodels:
    0:
      ▼ keys:
        0:
          type: "Submodel"
          local: true
          value: "https://example.com/ids/sm/5150_1101_1122_6130"
          idType: "IRI"
  ▼ asset:
    ▼ keys:
      0:
        type: "Asset"
        local: true
        value: ""
        idType: "IRDI"
  ▼ modelltype:
    name: "Asset"
    dataSpecification: []
    embeddedDataSpecifications: []
    idShort: ""
```

Better usability

Intuitive HMI



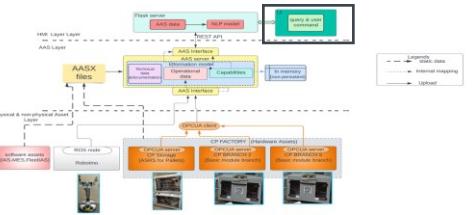
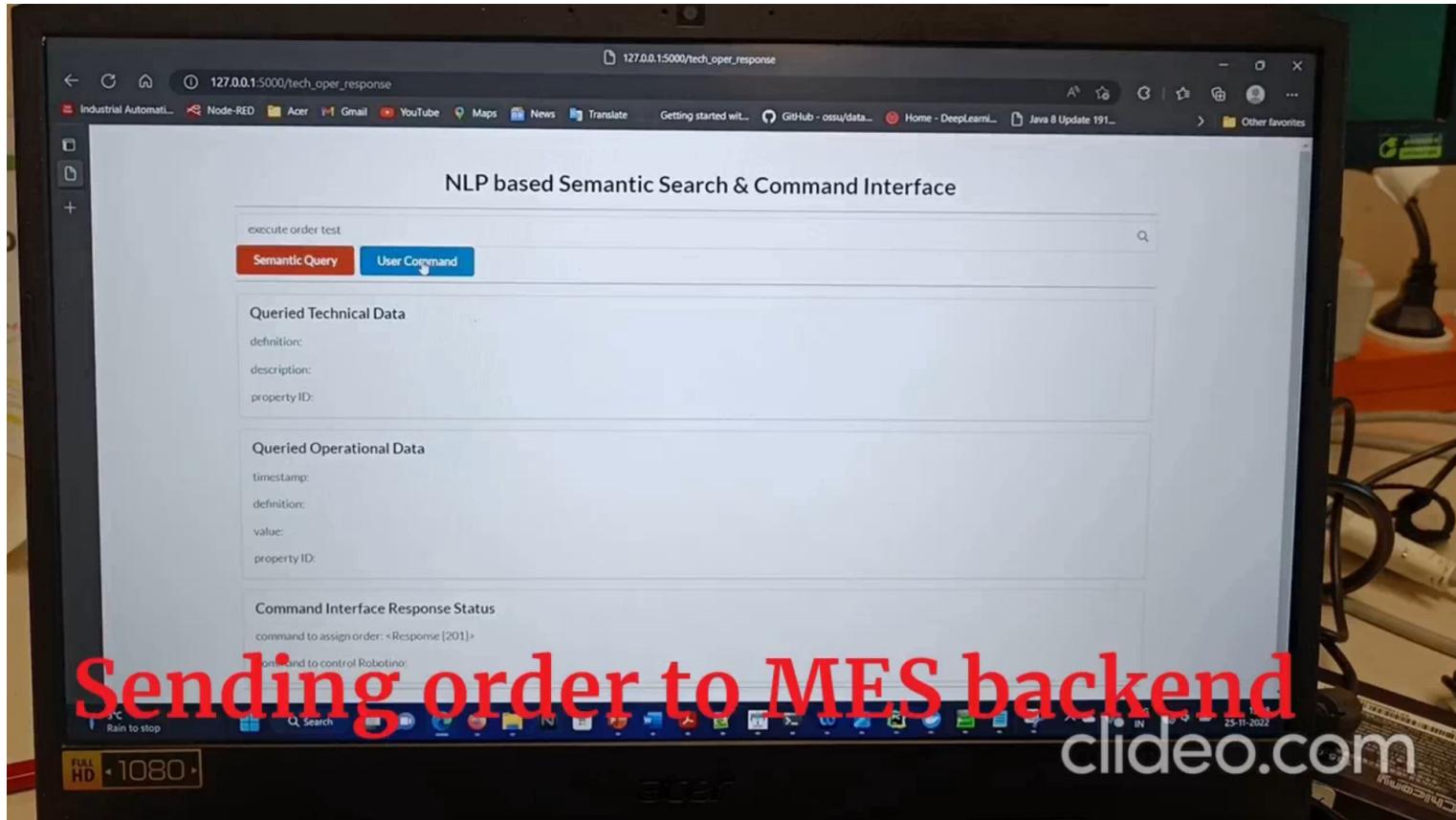
- Easy operation using Natural Language Processing (NLP)
- Helps in understanding human language to perform command and query
- NLP models are adapted from few foundational models like BERT, RoBERTa, etc.
- Sentence transformers (python based) library is used which provides pre-trained and finetuned models.
 - We make semantic similarity estimation/calculation.
- Two types of semantic search :
 - Symmetric Semantic Search →

Query input : *How to learn Python online?*
Corpus data : *How to learn Python on the web?*
 - Asymmetric Semantic Search →

Query input : *What is Python?*
Corpus data : *Python is an interpreted, high-level and general-purpose programming language*

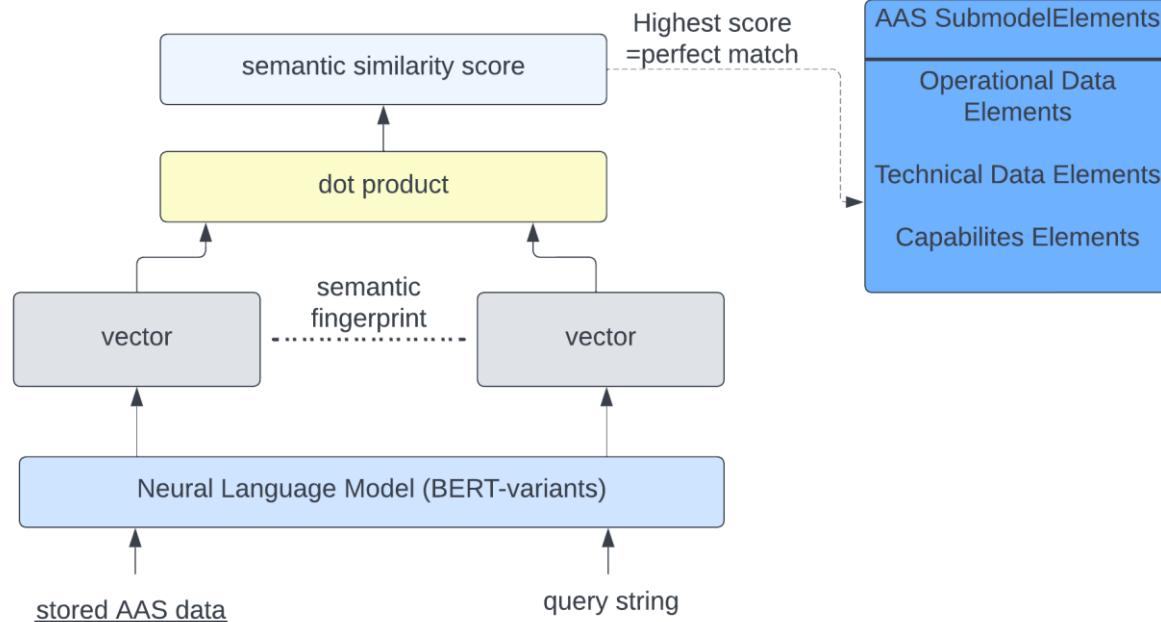
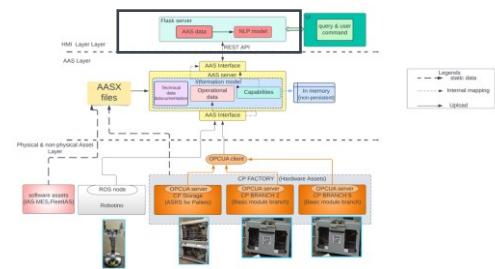
Results

NLP based Command Interface



Better Usability

Semantic score based on AAS data & query



Results

127.0.0.1:5000/tech_oper_response

NLP based Adaptive Semantic Query & Command Interface

module 2 junction

Semantic Query User Command

Queried Technical Data

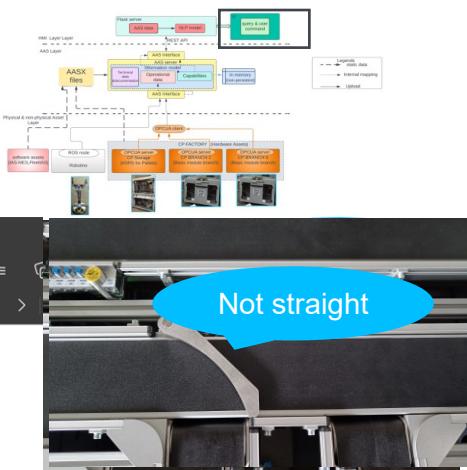
name: Junction_branch2
description: Pneumatic rotary cylinder with two end-position sensors
Semantic ID: <https://admin-shell.io/SemanticIdNotAvailable/1/1>

Queried Operational Data

timestamp: 11/30/2022, 14:49:11
name: Junction_branch2
value:[False] ← status
Semantic ID: https://example.com/ids/cd/2435_8142_1122_8850

Command Interface Response Status

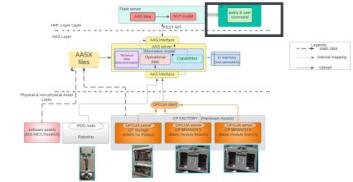
command to assign order:
command to control Robotino:



Not straight

Results

Experimental Evaluation



Adaptive query (user input)	Symmetric Semantic Search		Asymmetric Semantic Search	
	Technical data	Operational data	Technical data	Operational data
1. "module 2 junction status" 2. "Branch 5 junction"	✓	✓	✗	✓
1. "robotino lidar info" 2. "robotino laser"	✓	✓	✗	✓
"CP storage stopper information"	✓	✗	✓	✓
"stopper storage"	✓	✓	✗	✓
"Robotino odometry"	✓	✓	✗	✓

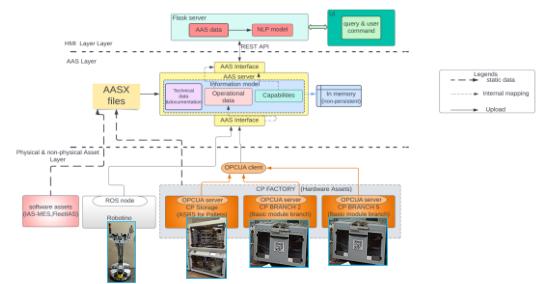
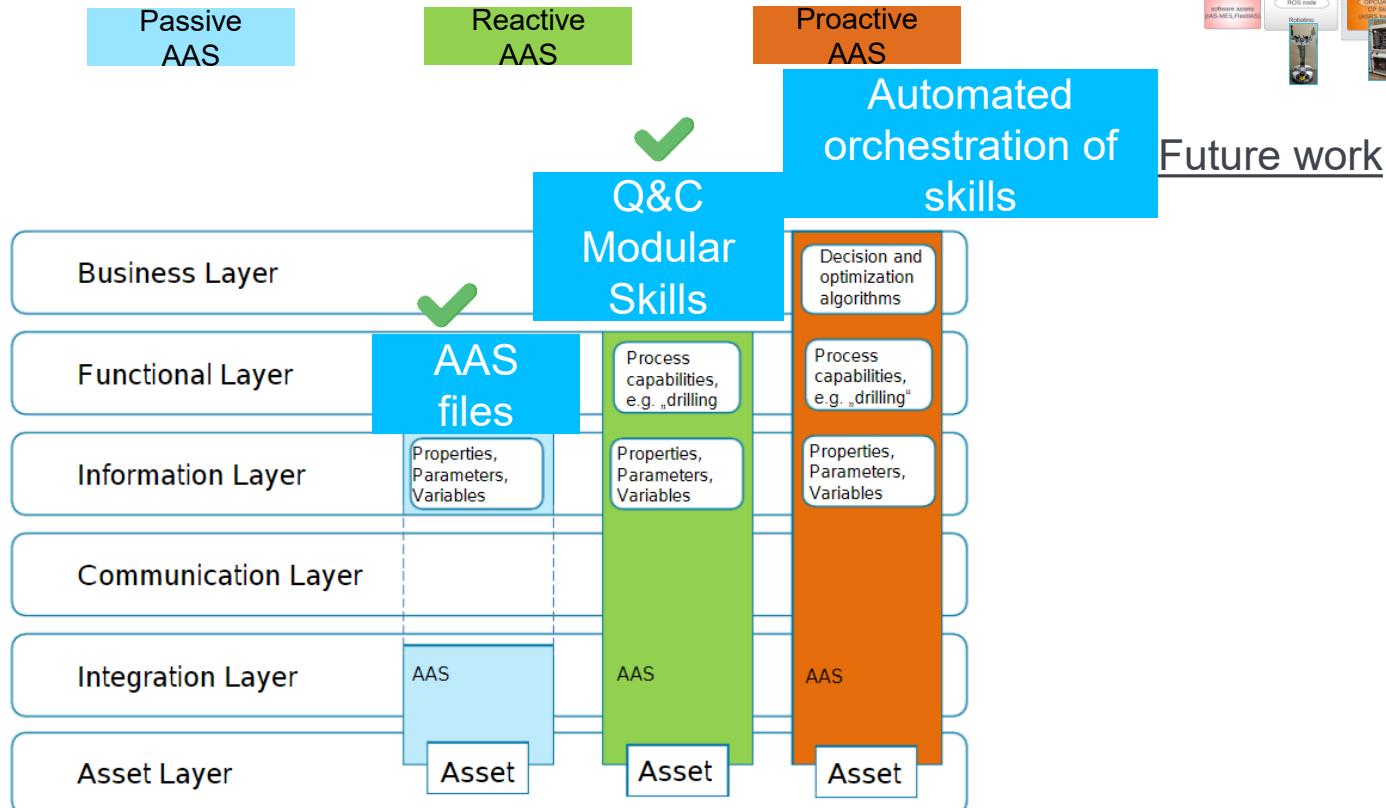
Selected NLM for test:

- Symmetric Semantic Search : all-distilroberta-v1
- Asymmetric Semantic Search : msmarco-distilbert-base-tas-b

Correct response:
Incorrect response:



Summary & Future work



[4] adapted from



University of Stuttgart
Institut of Industrial Automation
and Software Engineering

Thank you!



Manthan Venkataramana Shenoy

e-mail st175289@stud.uni-stuttgart.de

University of Stuttgart
Pfaffenwaldring 47, 70550



References

- [1] B. Ashtari Talkhestani, W. Schlägl, and M. Weyrich, "Synchronisierung von digitalen Modellen," atp Ed., vol. 59, no. 07–08, p. 62, Sep. 2017.
- [2] [BaSyx - Eclipsepedia](#)
- [3] [FA³ST Service — FA³ST Service documentation \(faaast-service.readthedocs.io\)](#)
- [4] Belyaev, Alexander & Diedrich, Christian. (2019). Specification "Demonstrator I4.0-Language" v3.0.
- [5] [Part 1 \(plattform-i40.de\)](#)