Skill-based Propagation of "Plug & Produce"-Devices
in Reconfigurable Production Systems by AML

3. AutomationML Anwenderkonferenz

Miriam Schleipen, Björn Hein, Julius Pfrommer, Kiril Aleksandrov,
Denis Stogl, Stefan Escaida, Jürgen Beyerer

www.skillpro-project.eu
FoF.NMP.2012-3 (314247)

General Information

SkillPro has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 314247.

- Duration: 10.2012 – 09.2015
- 12 Partners
  - 5 research institutes
  - 3 component providers
  - 2 software companies
  - 2 end users
- Coordinator: KIT-IPR

Clustered Projects

- I-Ramp^3, PRIME, CassaMobile

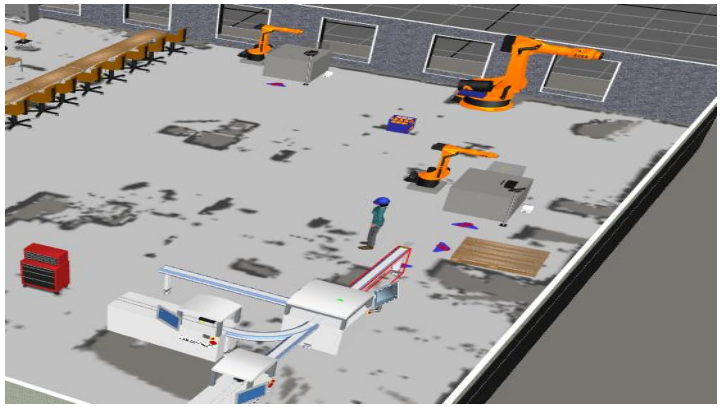
We have multi-use manufacturing hardware…

Industrial-grade 3D printer

Digitally programmed machine tool

Welding robot

…how can we build multi-use/flexible/adaptive manufacturing systems?

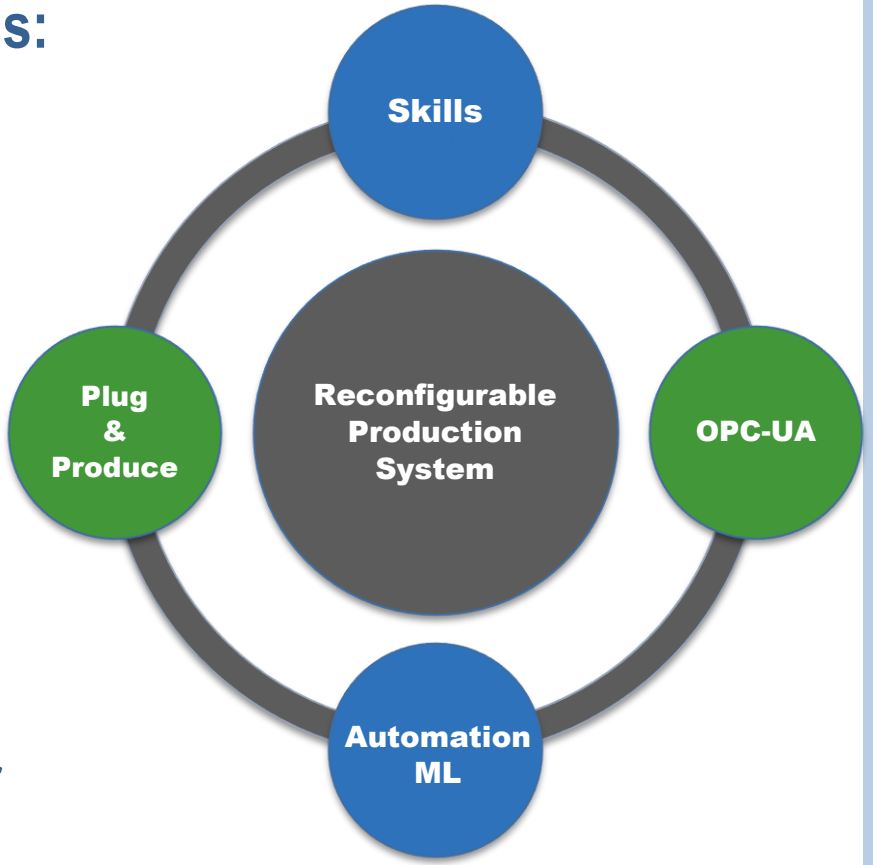


SkillPro-way to Plug‘n‘Produce

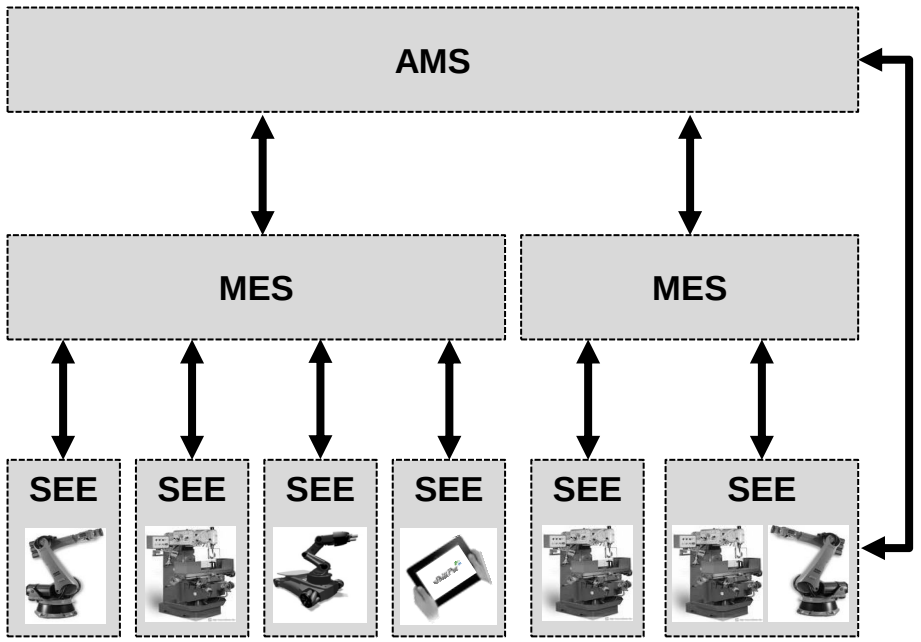
- **Standard interfaces and formats** to propagate the self-description of production resources
- **Deriving executable tasks** from **high-level descriptions** of the skills provided by the resources
- **Formal execution semantics** of these tasks that enable **reasoning about preconditions** and effects, and
- The **orchestration of production resources**, that provide access to task execution via services.

SkillPro Solution

- Utilization and extension of **standards:**
  - **AutomationML**
  - **OPC-UA**
- **Skill abstraction** as a common modelling, planning and execution entity expressed via **AutomationML**
- **Vertical integration** from the ERP layer towards physical resource layer via **OPC-UA**



SkillPro Architecture Overview



Asset Management System

- Long/Mid-term planning
- System reconfiguration
- Order mgmt, lifecycle analysis, …

Manufacturing Execution System

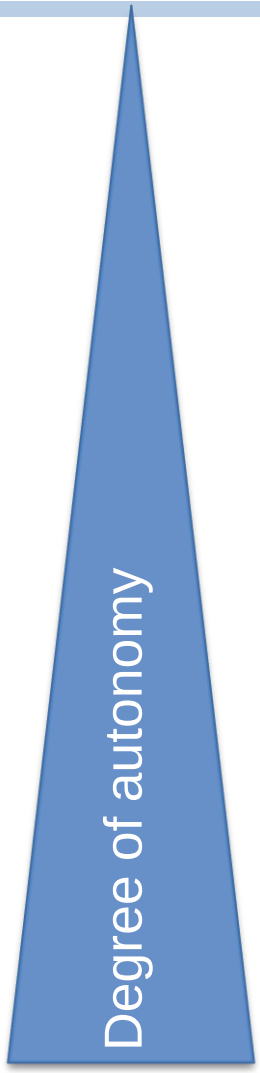
- Short-term planning for orchestration of skill execution
- Supervisory control

Skill Execution System

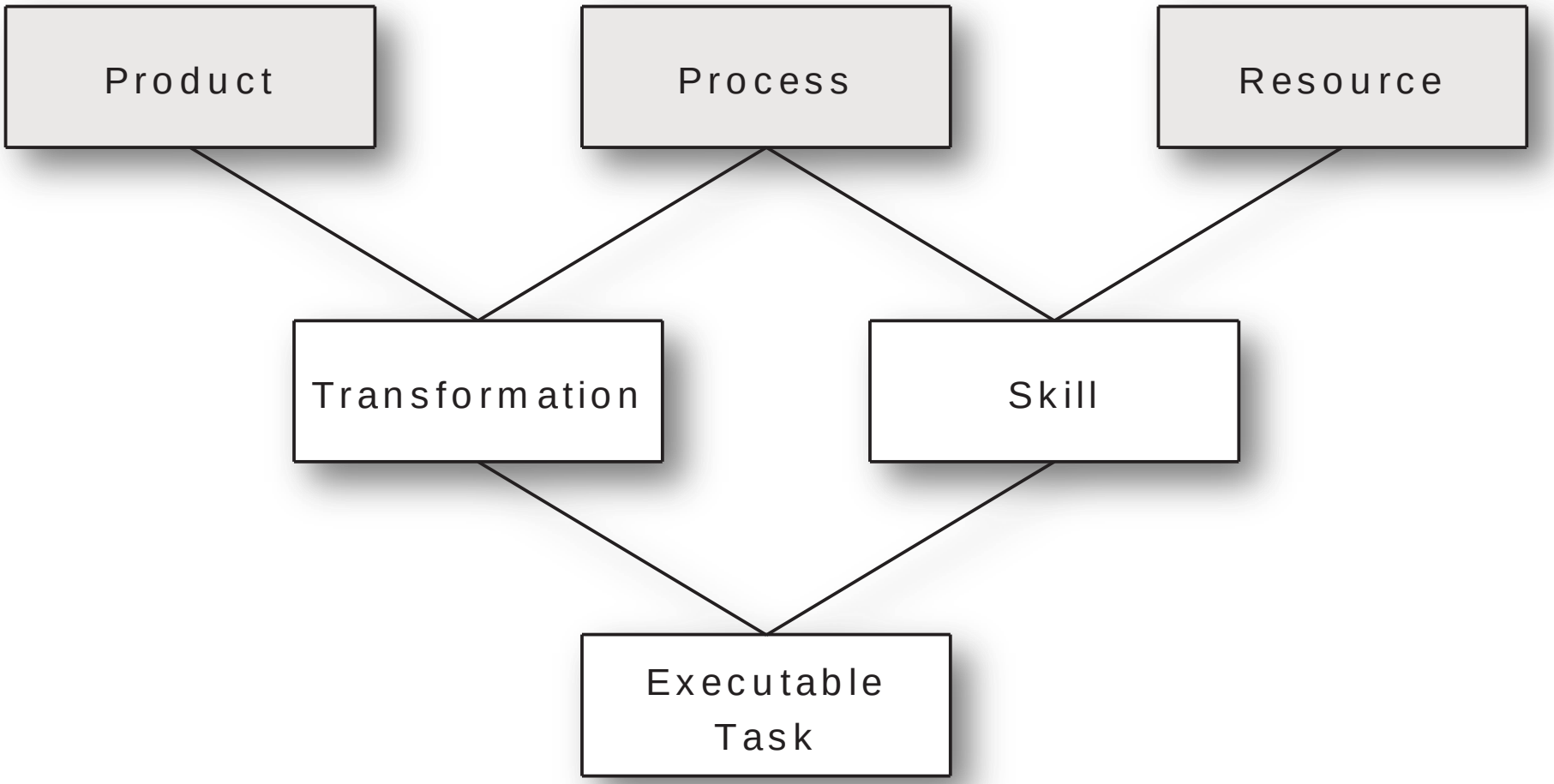
- Real-time execution
- Execution from high-level description
- Unified interface for different assets

Getting an „idea“ of skills (Examples)

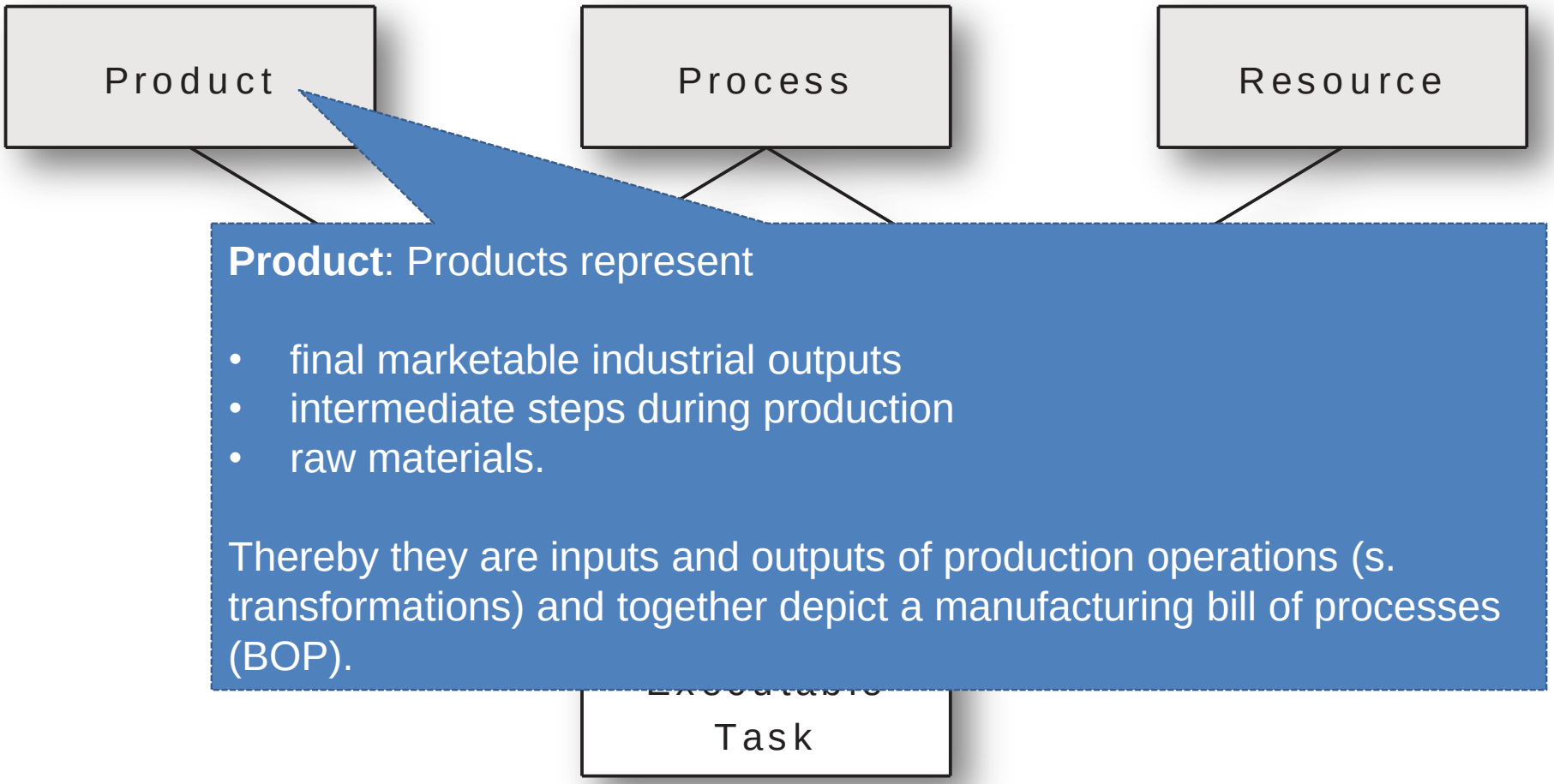
- Simple robot movement, e.g. PTP: Move from point A to point B
  - provided by the robot manufacturer
- Sensor based correction of the movement/position
  - Tasks
    - e.g. moving the TCP (Tool-Center-Point) along a line
    - e.g. gripping an object with non fixed end-position
    - e.g. gripping an object with moving end-position
  - Question
    - How to do this independently... ? (communication)
    - How to describe this independently from robot manufacturer and sensor system?
- Automatic path planning between two positions
  - Environment must be perceived
  - Collision free movement must be calculated
  - Question
    - How to do describe the task (obstacles, restriction, options)?
    - System architecture to allow the execution of the task?
- Automatic path planning handling an object
  - How to describe the task (object, start and goal position, boundary conditions)?



SkillPro – Matching PPR approach



SkillPro – Matching PPR approach

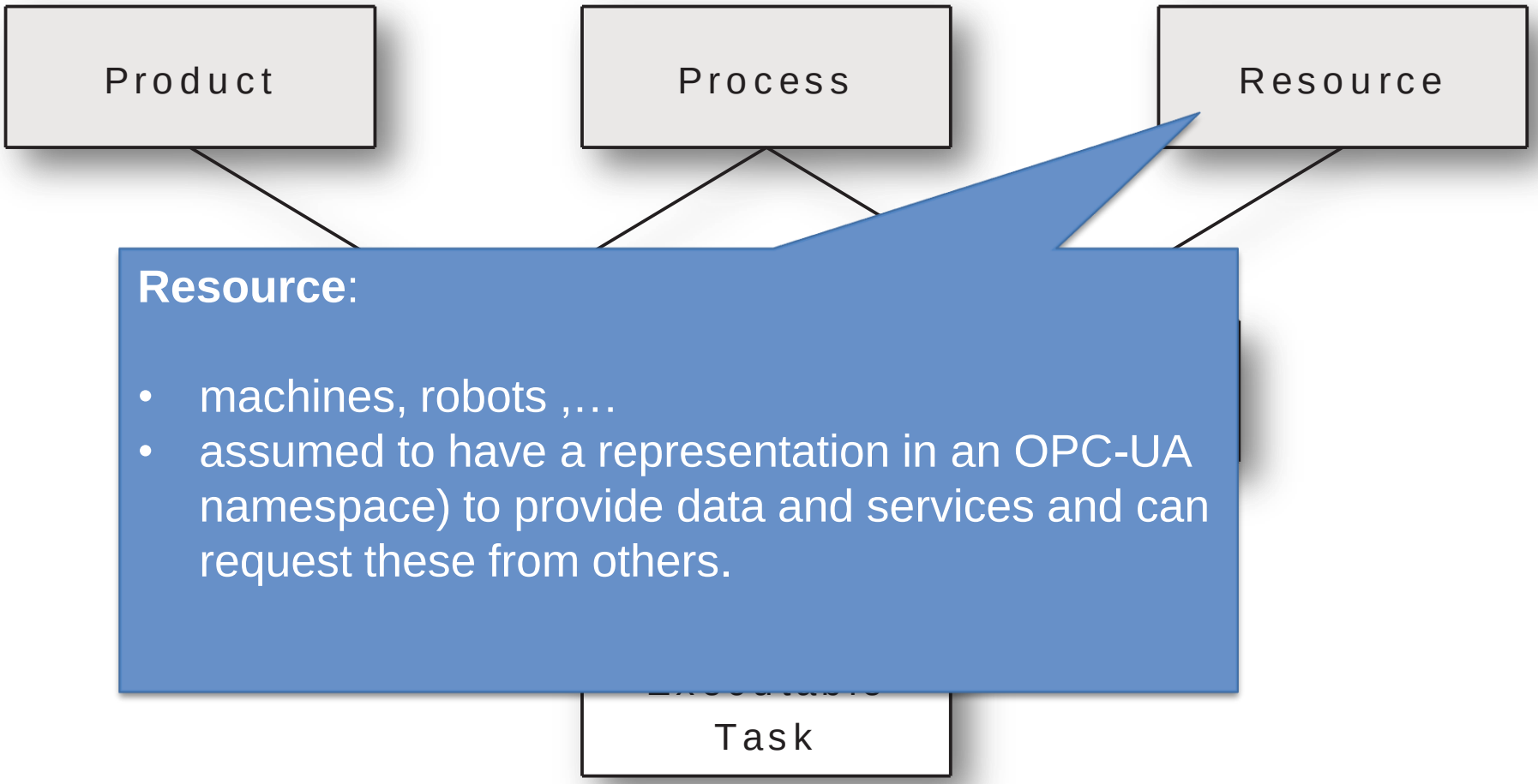


Product: Products represent

- final marketable industrial outputs
- intermediate steps during production
- raw materials.

Thereby they are inputs and outputs of production operations (s. transformations) and together depict a manufacturing bill of processes (BOP).

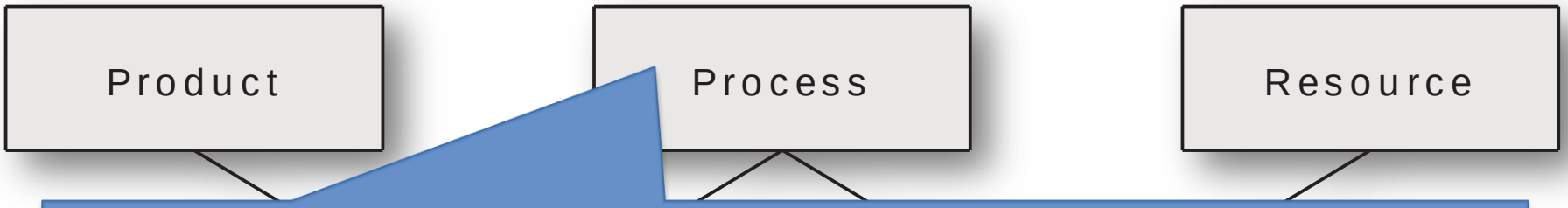
SkillPro – Matching PPR approach



Resource:

- machines, robots ,…
- assumed to have a representation in an OPC-UA namespace) to provide data and services and can request these from others.

SkillPro – Matching PPR approach

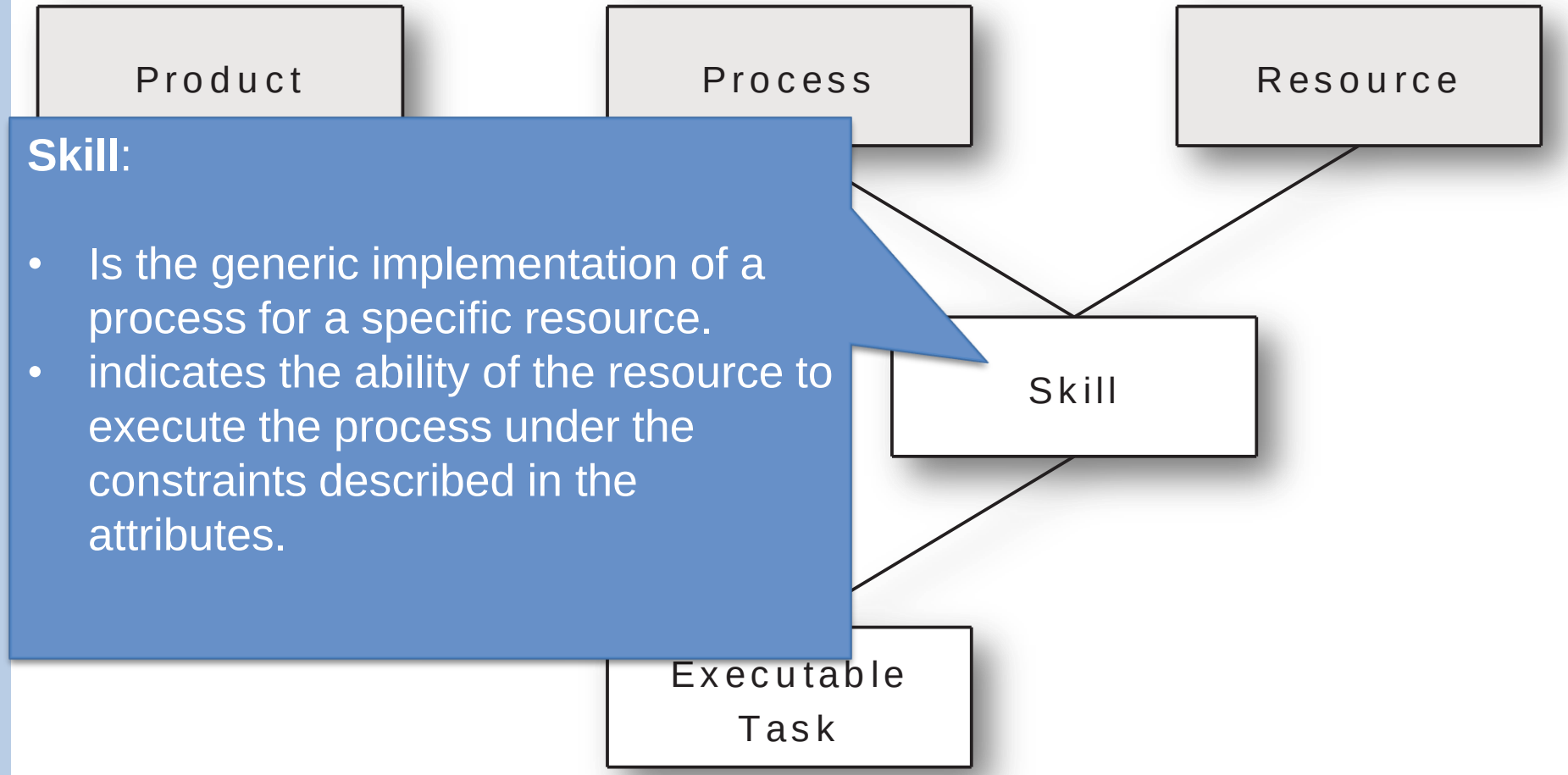


Process:

- Process refers to a type of manufacturing, logistics or other production related process
- provide the structure for describing these processes in their respective domain.
- Processes form a hierarchy where derived processes inherit the attributes of their parent.

These “process description templates” are instantiated by transformations that require a process with certain constraints on the process attributes, and from skills that represent the implementation of a process on a resource (but still independent of a specific product).

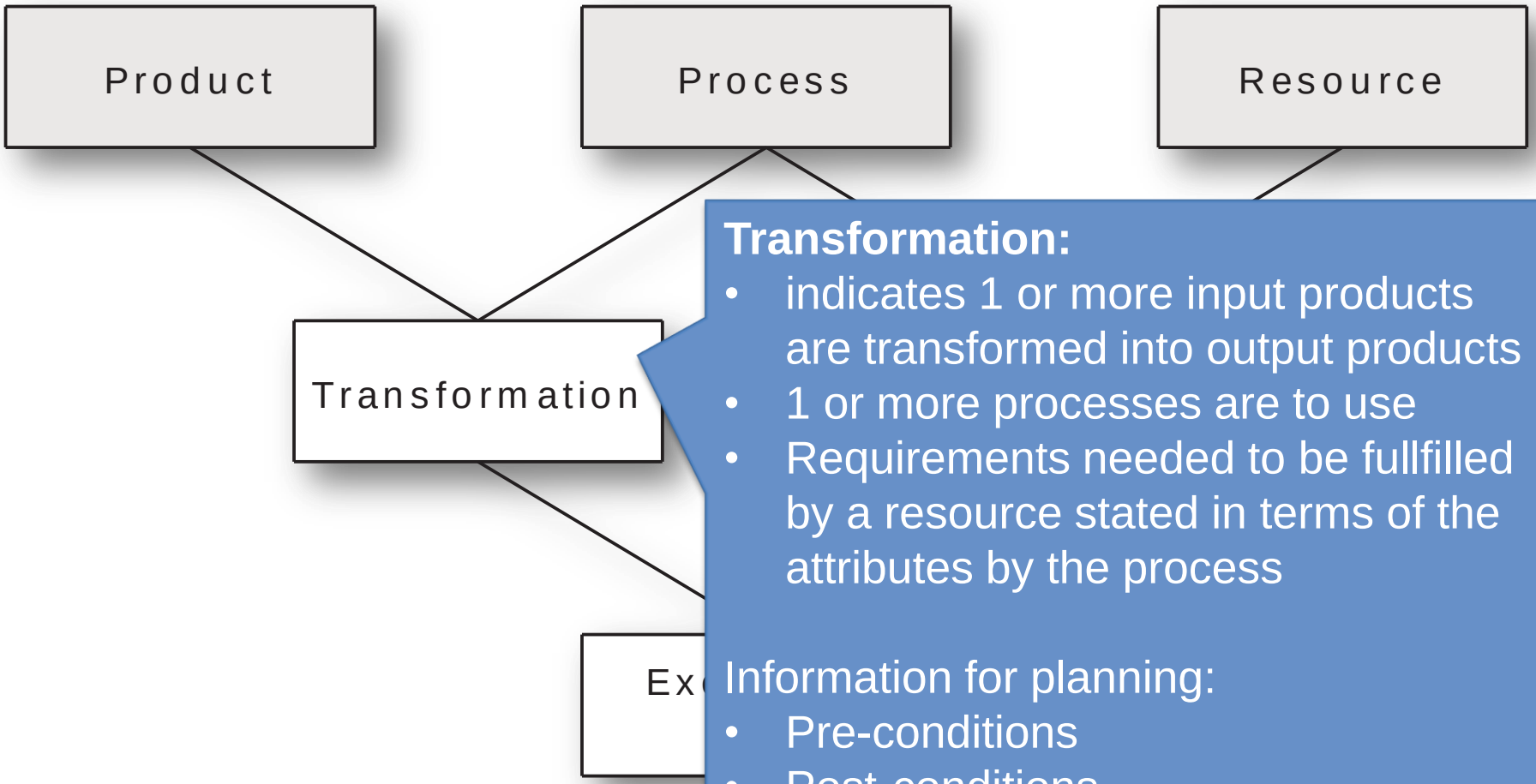
SkillPro – Matching PPR approach



Skill:

- Is the generic implementation of a process for a specific resource.
- indicates the ability of the resource to execute the process under the constraints described in the attributes.

SkillPro – Matching PPR approach



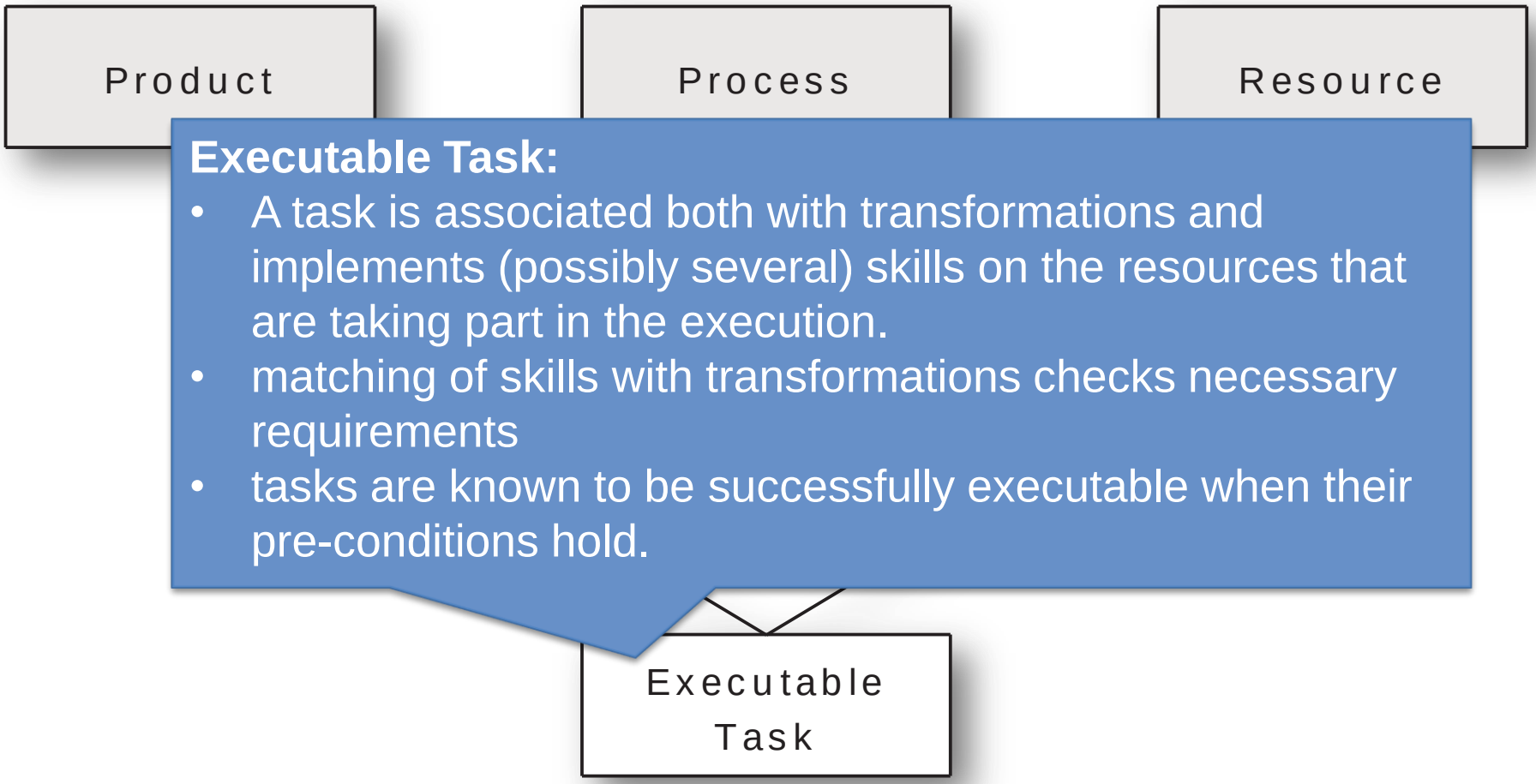
Transformation:

- indicates 1 or more input products are transformed into output products
- 1 or more processes are to use
- Requirements needed to be fullfilled by a resource stated in terms of the attributes by the process

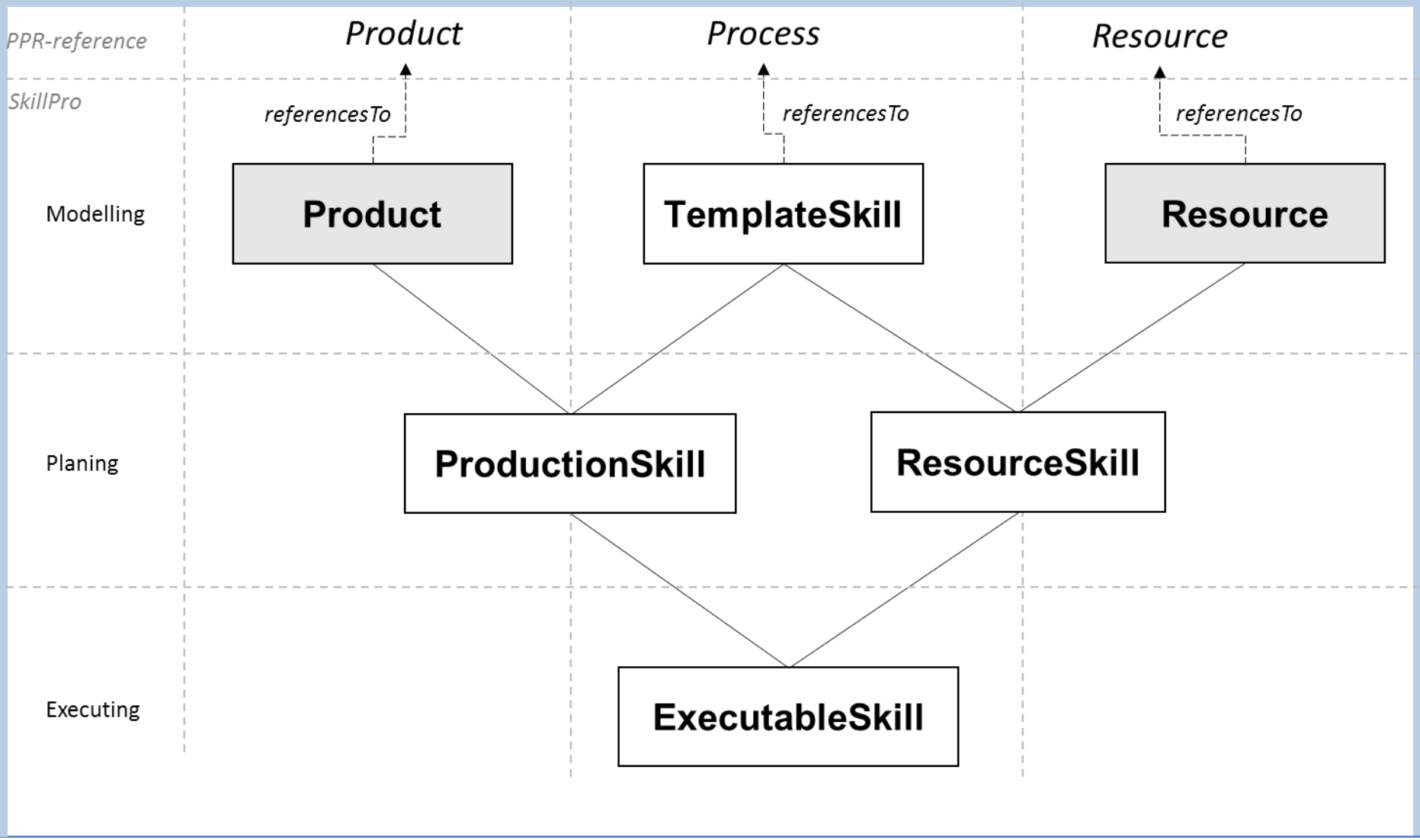
Information for planning:

- Pre-conditions
- Post-conditions
- Temporal constraints

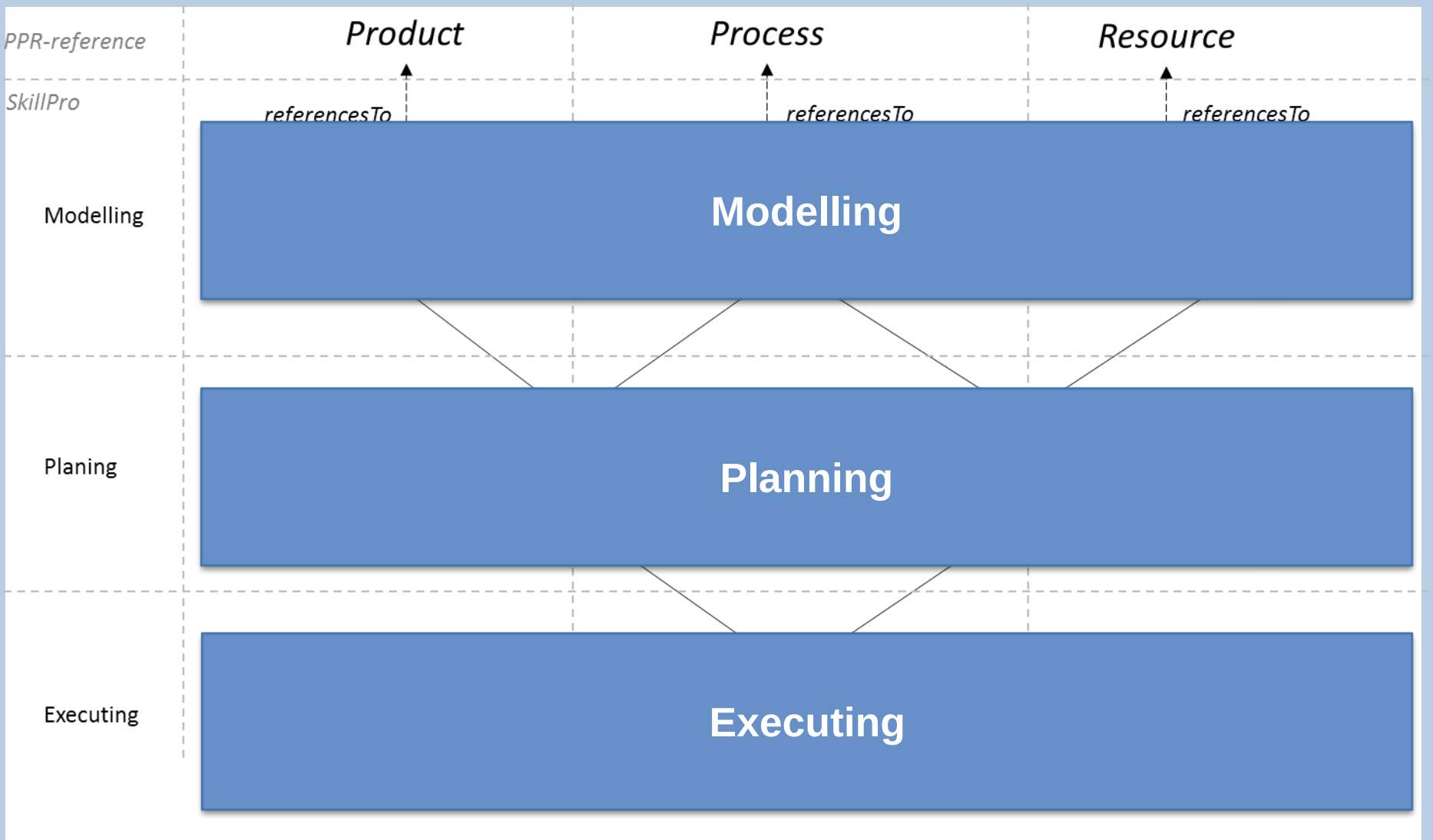
SkillPro – Matching PPR approach



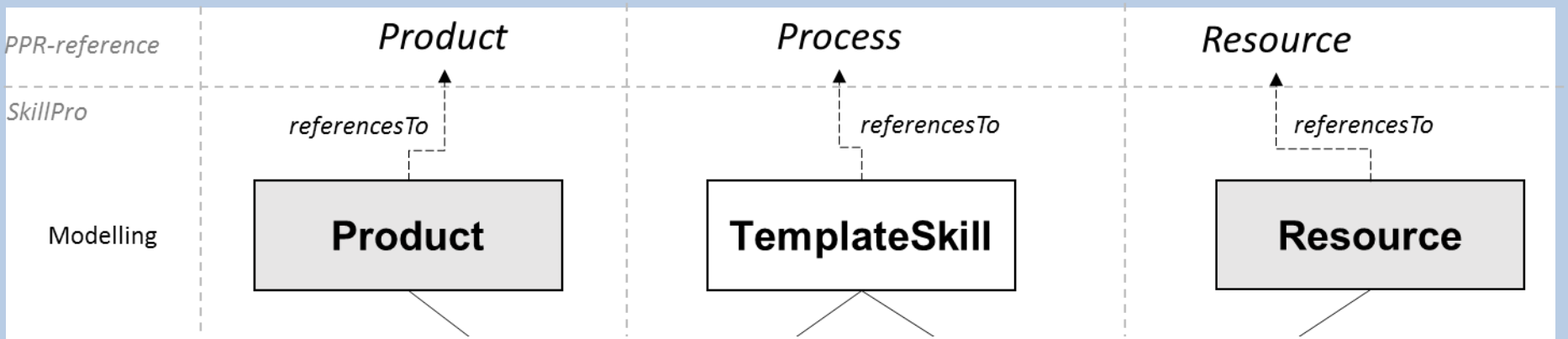
Derived Skill Definition in SkillPro



Skill domains



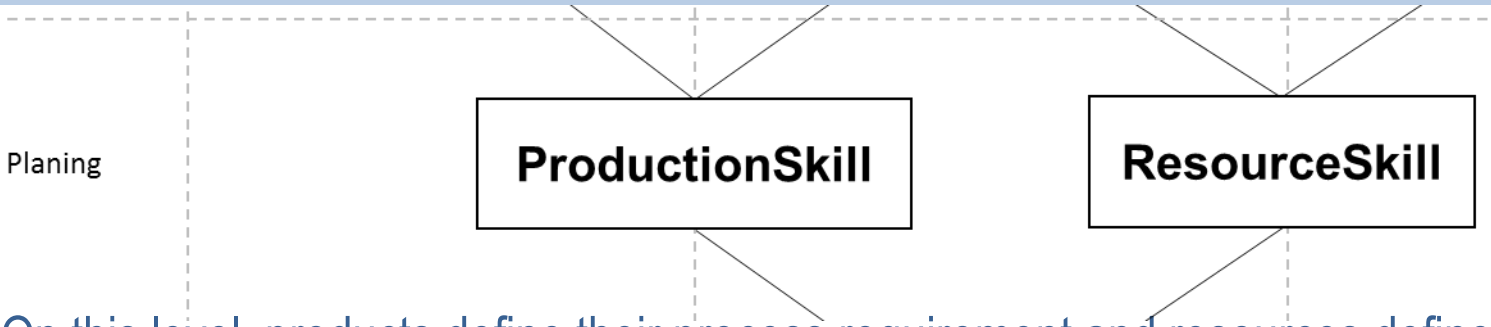
Modelling Level



This level is used for describing and managing elements of the 3 types of the product centric domain.

- Product keeps its original meaning.
  - represents a final product that contains a list of intermediate products with corresponding connections between them that represent transformation steps, building a graph.
- TemplateSkill refers to a process
  - Inheritance hierarchies can be used to build a taxonomical structures that describe production and logistic processes.
  - The level of abstraction depends on the modeler and the goals.
- Resource keeps ist original meaning

Planing Level: ProductionSkill



On this level, products define their process requirement and resources define the processes they provide.

- ProductionSkill corresponds to a product requirement regarding a transformation step during production (product1 becomes product2). It is therefore linked to one or several input and output products and references one TemplateSkill, one production process, that is involved in the transformation.
- The Product provides values for the parameters of the skill templates that are required for the transformation.

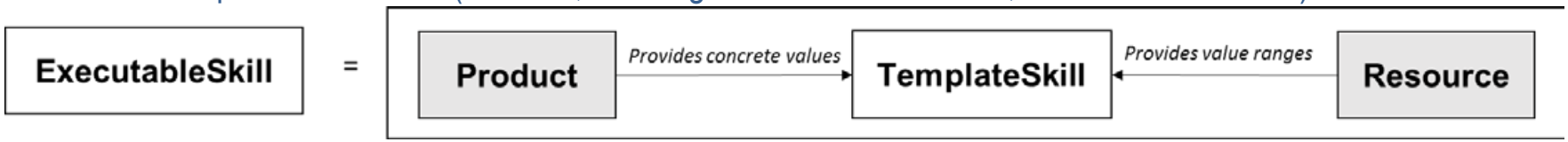
Planing Level: ResourceSkill

- ResourceSkill is the realisation of a **TemplateSkill** by a **Resource**. It defines the ability of a resource to perform a process (described by a **TemplateSkill**). It furthermore provides value ranges or restrictions for the metadata to be described in the implemented **TemplateSkill**. Those constraints restrict the parameters of **TemplateSkill** based on some resource specifics. As an example, in a CNC mill, the constraints of a *milling* process could be the metal type and size of the work piece.

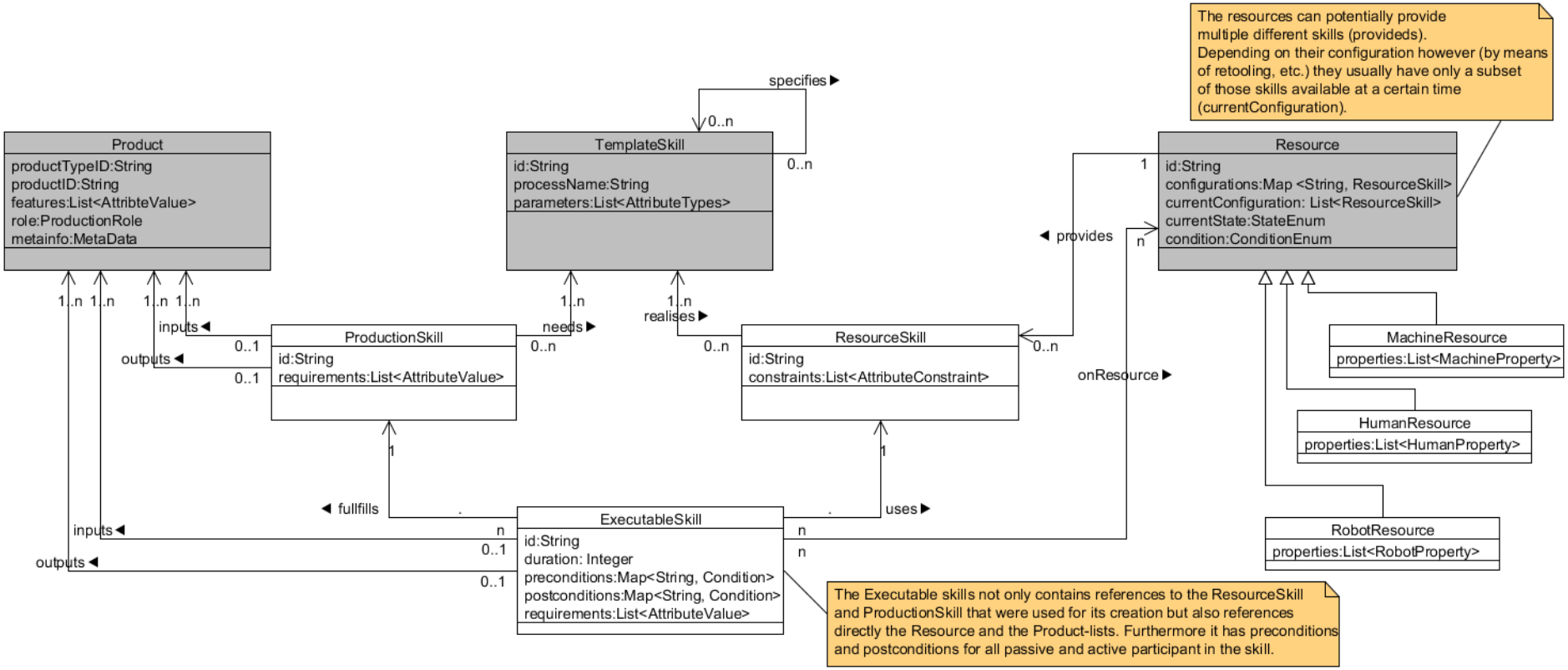
Execution Level

The entities on this level are used only for fine scheduling and triggering of executions.

- ExecutableSkill is a match of **ProductionSkill** and **ResourceSkill**.
  - It contains a process allocated to a certain resource and certain product.
  - It corresponds to an executing task triggerable by the MES
  - It contains the information needed by the SEE to execute it.
  - From the outside it is opaque (black boxes) with only the following information being publicly available:
    - Involved Resources
    - Pre-conditions (input product type, state and configuration of the resources)
    - Post-conditions (output product type, state and configuration of the resources)
    - Temporal constraints (duration, blocking of involved resources, not considered here)



Skill model


The resources can potentially provide multiple different skills (provideds). Depending on their configuration however (by means of retooling, etc.) they usually have only a subset of those skills available at a certain time (currentConfiguration).
Product
productTypeID:String
productID:String
features:List<AttribteValue>
role:ProductionRole
metainfo:MetaData
TemplateSkill
id:String
processName:String
parameters:List<AttributeTypes>
specifies
0..n
0..n
Resource
id:String
configurations:Map <String, ResourceSkill>
currentConfiguration: List<ResourceSkill>
currentState:StateEnum
condition:ConditionEnum
provides
1
n
1..n
inputs
outputs
0..1
ProductionSkill
id:String
requirements:List<AttributeValue>
needs
realises
ResourceSkill
id:String
constraints:List<AttributeConstraint>
onResource
MachineResource
properties:List<MachineProperty>
HumanResource
properties:List<HumanProperty>
RobotResource
properties:List<RobotProperty>
fullfills
uses
ExecutableSkill
id:String
duration: Integer
preconditions:Map<String, Condition>
postconditions:Map<String, Condition>
requirements:List<AttributeValue>
The Executable skills not only contains references to the ResourceSkill and ProductionSkill that were used for its creation but also references directly the Resource and the Product-lists. Furthermore it has preconditions and postconditions for all passive and active participant in the skill.

Examples in AML

AML: SkillParameter

- Several skills define parameters, their values or their value constraints. The Attribute defined in AutomationML fulfil all these requirements.

```
<Attribute Name="Height" AttributeDataType="xs:double" Unit="cm">
  <Description>Difference between ground level to highest point of the lamp.</Descrip
tion>
  <DefaultValue>0815</DefaultValue>
  <Value>525</Value>
  <Constraint Name="MinMaxHeight">
    <OrdinalScaledType>
      <RequiredMaxValue>900</RequiredMaxValue>
      <RequiredValue/>
      <RequiredMinValue>200</RequiredMinValue>
    </OrdinalScaledType>
  </Constraint>
</Attribute>
<Attribute Name="Shape" AttributeDataType="xs:string">
  <Value>E27</Value>
  <Constraint Name="ExceptedShapes">
    <NominalScaledType>
      <RequiredValue>E21</RequiredValue>
      <RequiredValue>E27</RequiredValue>
```

AML: Product

- Products should be modelled as InternalElements and should assign at least the RoleClass "SkillProRoleClassLib/SkillProProduct".
- The RoleClass assignment can be done via RoleRequirement or SupportedRoleClass.
- The SkillPro product role defines the attribute ProductTypeID and the ExternalInterface "ProductionSkillInterface", an interface to be linked to ProductionSkills.
- The ProductID can stored directly in the ID attribute of the InternalElement.

```
<RoleClass Name="SkillProProduct" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Product">
  <Attribute Name="ProductTypeID" />
  <ExternalInterface Name="ProductionSkillInterface" RefBaseClassPath="SkillProInterfaceClassLib/ProductionSkillConnector" />
</RoleClass>
```

AML: ProductionSkillConnector

- It is necessary to connect ProductionSkills with their incoming and outgoing products.
- Therefore the InterfaceClass "SkillProInterfaceClassLib/ProductionSkillConnector" is defined.
- All ExternalInterfaces from ProductionSkills and products should use this InterfaceClass for assigning each other.

```
<InterfaceClassLib Name="SkillProInterfaceClassLib">
  <Version>0.1</Version>
  <InterfaceClass Name="ProductionSkillConnector" RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/PPRConnector"/>
</InterfaceClassLib>
```

```
<RoleClass Name="SkillProProduct" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Product">
  <Attribute Name="ProductTypeID" />
  <ExternalInterface Name="ProductionSkillInterface" RefBaseClassPath="SkillProInterfaceClassLib/ProductionSkillConnector" />
</RoleClass>
```

AML: ProductionSkill

- ProductionSkills should be modelled as InternalElements and should assign the RoleClass "SkillProRoleClassLib/ProductionSkill" and the needed TemplateSkill role class.
- Parameters/attributes of the TemplateSkill can filled according to the production requirements. The product interfaces should connected to the appropriate products via InternalLinks.

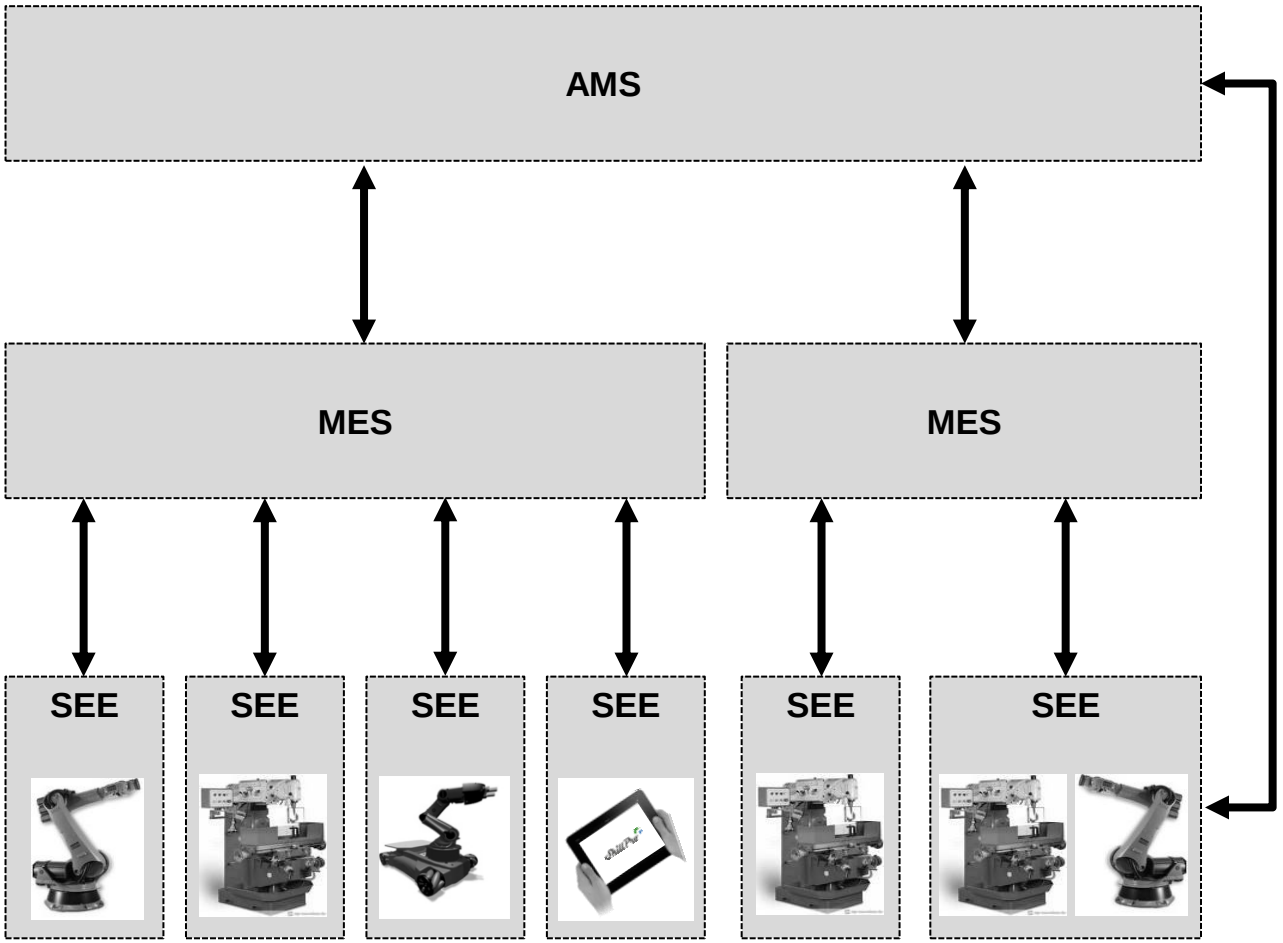
```
<RoleClass Name="ProductionSkill" RefBaseClassPath="AutomationMLBaseRoleClassLib/A
utomationMLBaseRole/Process">
  <Attribute Name="RefTemplateSkill">
    <Description>Path of the needed TemplateSkill</Description>
    <DefaultValue>AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Process</Defau
ltValue>
  </Attribute>
  <ExternalInterface Name="InputProductInterface" RefBaseClassPath="SkillProInterf
aceClassLib/ProductionSkillConnector"/>
  <ExternalInterface Name="OutputProductInterface" RefBaseClassPath="SkillProInter
faceClassLib/ProductionSkillConnector"/>
</RoleClass>
```

AML: ExecutableSkill

- ExecutableSkills are modelled as InternalElements and should assign the RoleClass "SkillProRoleClassLib/ExecutableSkill".
- ExecutableSkills includes a reference to the responsible SEE and a estimated duration for execution.
- An ExecutableSkill also contains a child InternalElement for each involved asset.
- These elements assign the role "SkillProRoleClassLib/ResourceExecutableSkill"

```
<RoleClass Name="ResourceExecutableSkill" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
  <Attribute Name="ResourceId"/>
  <Attribute Name="PreProductId"/>
  <Attribute Name="PreConfiguration"/>
  <Attribute Name="PostProductId"/>
  <Attribute Name="PostConfiguration"/>
  <Attribute Name="Slack"/>
  <Attribute Name="Duration" />
</RoleClass>
```

Planning and Executing



Planning and Executing

A central component of the MES is the planner that takes as input a description of the available executable skills and orders. It then generates a feasible action plan to (partially) achieve the stated goals using all the available resources.

- Planning is done using algorithms known from AI approaches
- The input is a description of the available executable skills and the goals that shall be achieved (final product).
- The planner then performs a forward-search in the state-space to find the best (according to some cost function) sequence of actions that achieves the goal.
- An existing planner is used → a PDDL description out of the AML is generated

Example (Domain Definition)


```
(define (domain skillpro)
   ; predicates are truth-"functions" returning true or false
   (:predicates (contains-product ?resource ?product)
                (has-configuration ?resource ?configuration))

   (:constants drilling_machine pcb pcb_drilled drill_nodrill drill_size8)

    (:action load_drill8
        :parameters (?resource)
        :precondition (and (equal ?resource drilling_machine)
                           (has-configuration ?resource drill_nodrill))
        :effect (and (not (has-configuration ?resource drill_nodrill))
                     (has-configuration ?resource drill_size8))
    )

    (:action drill_pcb
        :parameters (?resource)
        :precondition (and (equal ?resource drilling_machine)
                           (contains-product ?resource pcb)
                           (has-configuration ?resource drill_size8))
        :effect (and (not (contains-product ?resource pcb))
                     (contains-product ?resource pcb_drilled))
    )
)
```

Example (Problem Instance Definition)


```
(define (problem skillpro-problem-instance)
    (:domain skillpro)

    ; predicates set here are true. all others are false
    (:init (contains-product drilling_machine pcb)
           (has-configuration drilling_machine drill_nodrill)
    )

    (:goal (and (contains-product drilling_machine pcb_drilled)
                (has-configuration drilling_machine drill_size8))
    )
)
```

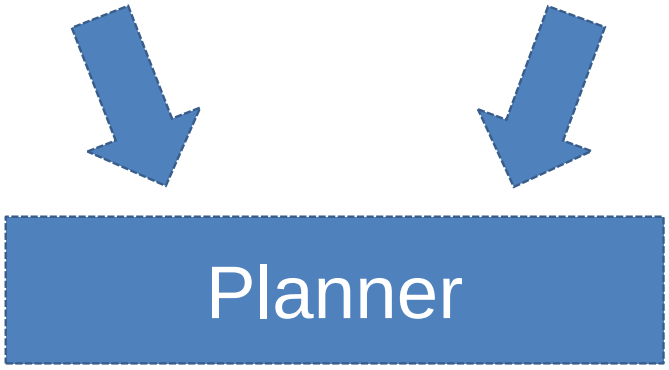
Result

```
(load_drill8 drilling_machine)
(drill_pcb drilling_machine)
```

Planning in the MES

Resources
- Current state
- Possible states
- Available ExecutableSkills

Orders
- Products and quantities
- Delivery date
- Per-order ExecutableSkills

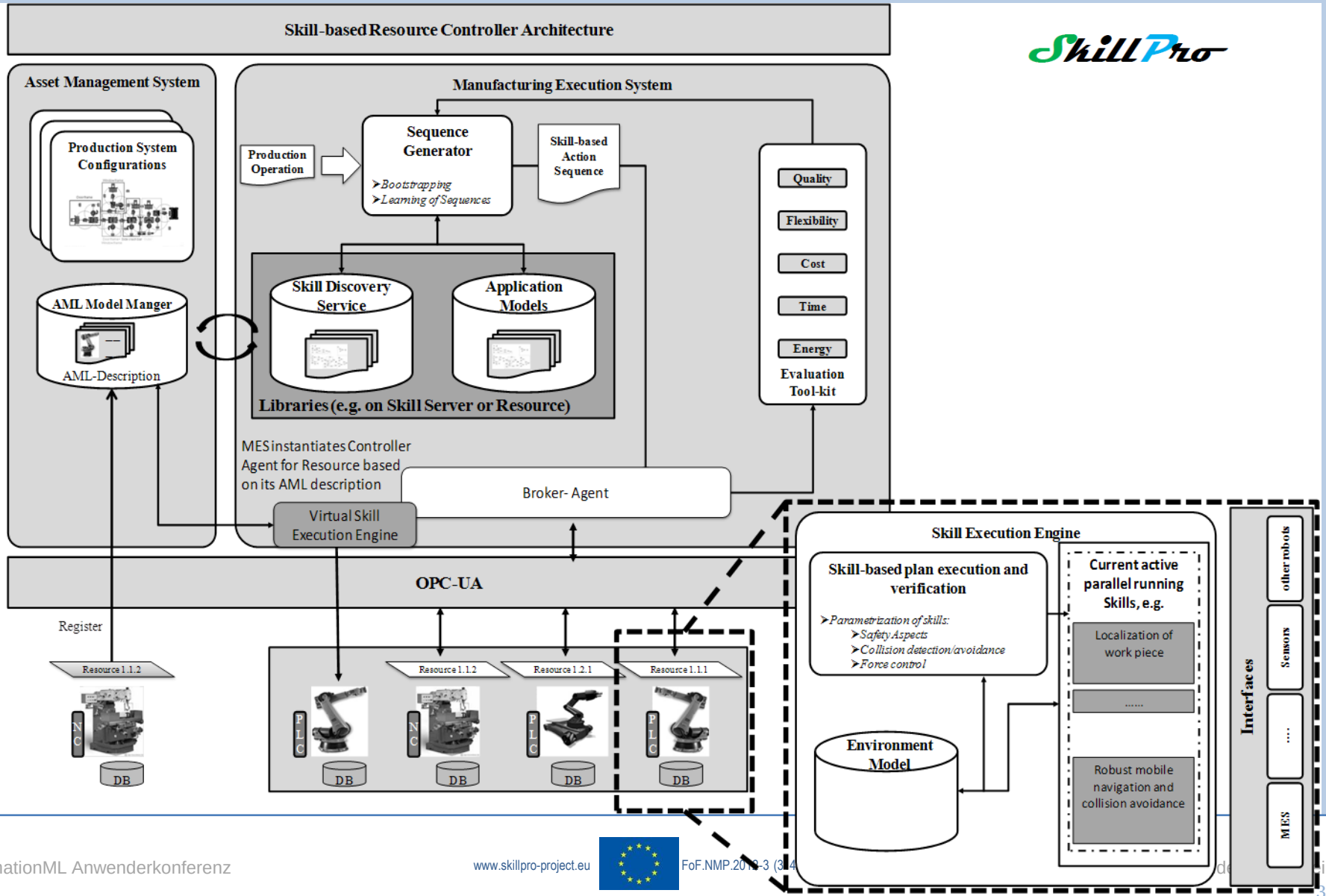


→ **Executable ActionPlan**

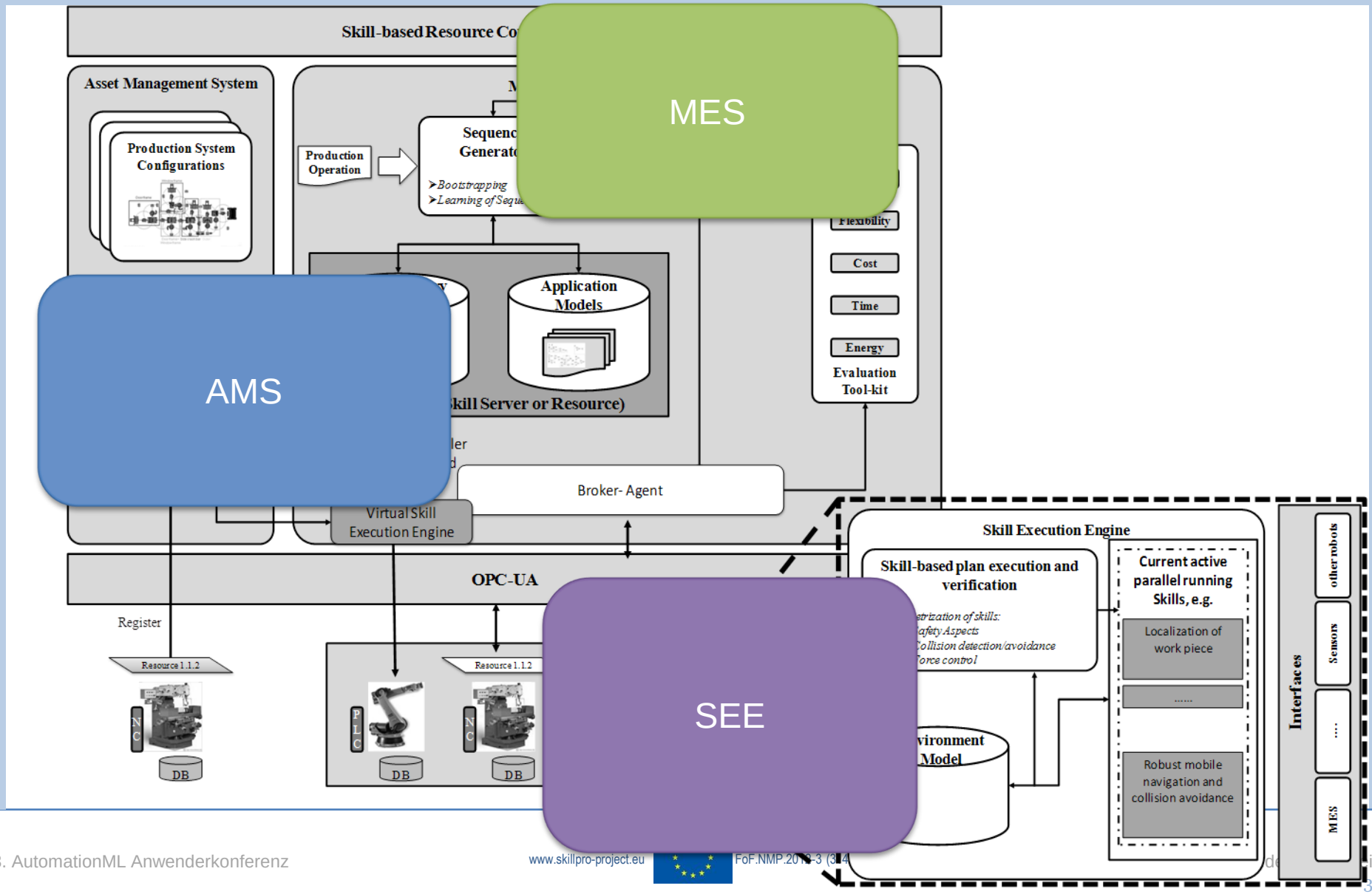
Video

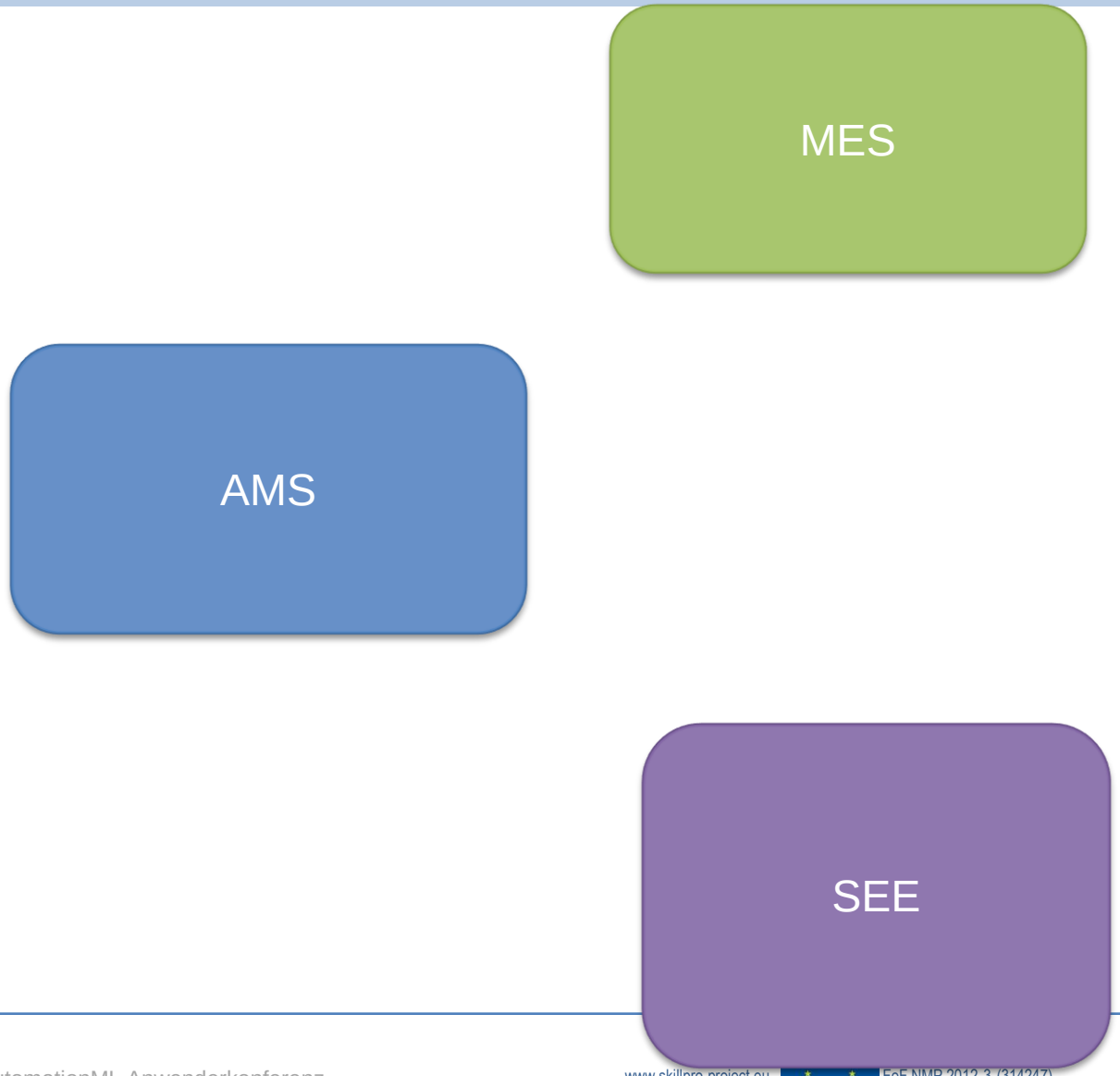
Getting further information….

Concept


Skill-based Resource Controller Architecture
Asset Management System
Production System Configurations
AML Model Manger
AML-Description
Manufacturing Execution System
Production Operation
Sequence Generator
➢Bootstrapping
➢Learning of Sequences
Skill-based Action Sequence
Skill Discovery Service
Application Models
Libraries (e.g. on Skill Server or Resource)
Quality
Flexibility
Cost
Time
Energy
Evaluation Tool-kit
MES instantiates Controller Agent for Resource based on its AML description
Broker- Agent
Virtual Skill Execution Engine
OPC-UA
Register
Resource 1.1.2
Resource 1.1.2
Resource 1.2.1
Resource 1.1.1
NC
PLC
DB
Skill Execution Engine
Skill-based plan execution and verification
➢Parametrization of skills:
➢Safety Aspects
➢Collision detection/avoidance
➢Force control
Current active parallel running Skills, e.g.
Localization of work piece
.....
Robust mobile navigation and collision avoidance
Environment Model
Interfaces
other robots
Sensors
....
MES

Concept


MES
AMS
SEE
Skill-based Resource Co
Asset Management System
Production System Configurations
Production Operation
Sequenc
Generat
➢Bootstrapping
➢Learning of Sequ
Application Models
Flexibility
Cost
Time
Energy
Evaluation Tool-kit
kill Server or Resource)
Broker- Agent
Virtual Skill Execution Engine
OPC-UA
Register
Resource 1.1.2
Resource 1.1.2
DB
DB
DB
Skill Execution Engine
Skill-based plan execution and verification
Current active parallel running Skills, e.g.
Localization of work piece
.....
Robust mobile navigation and collision avoidance
vironment Model
Interfaces
other robots
Sensors
....
MES


MES
AMS
SEE

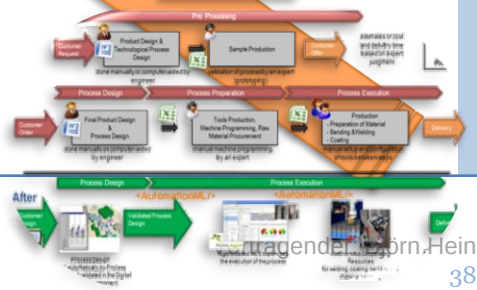
Concept

MES

AMS

SEE

D1.1.0
D1.2.0



Concept

MES

AMS

SEE

http://www.skillpro-project.eu


Version from May 16, 2013 15:24

Skill-based Propagation of "Plug&Produce"-Devices in Reconfigurable Production Systems by AML

Title: Use Case Definitions and Integrated Demo Scenarios

Deliverable: D1.1.0

Prepared by:

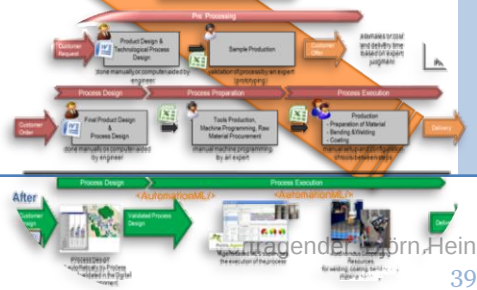
Name	Mike Ludwig
Organisation	DDE
Date	May 16, 2013

Approved by:

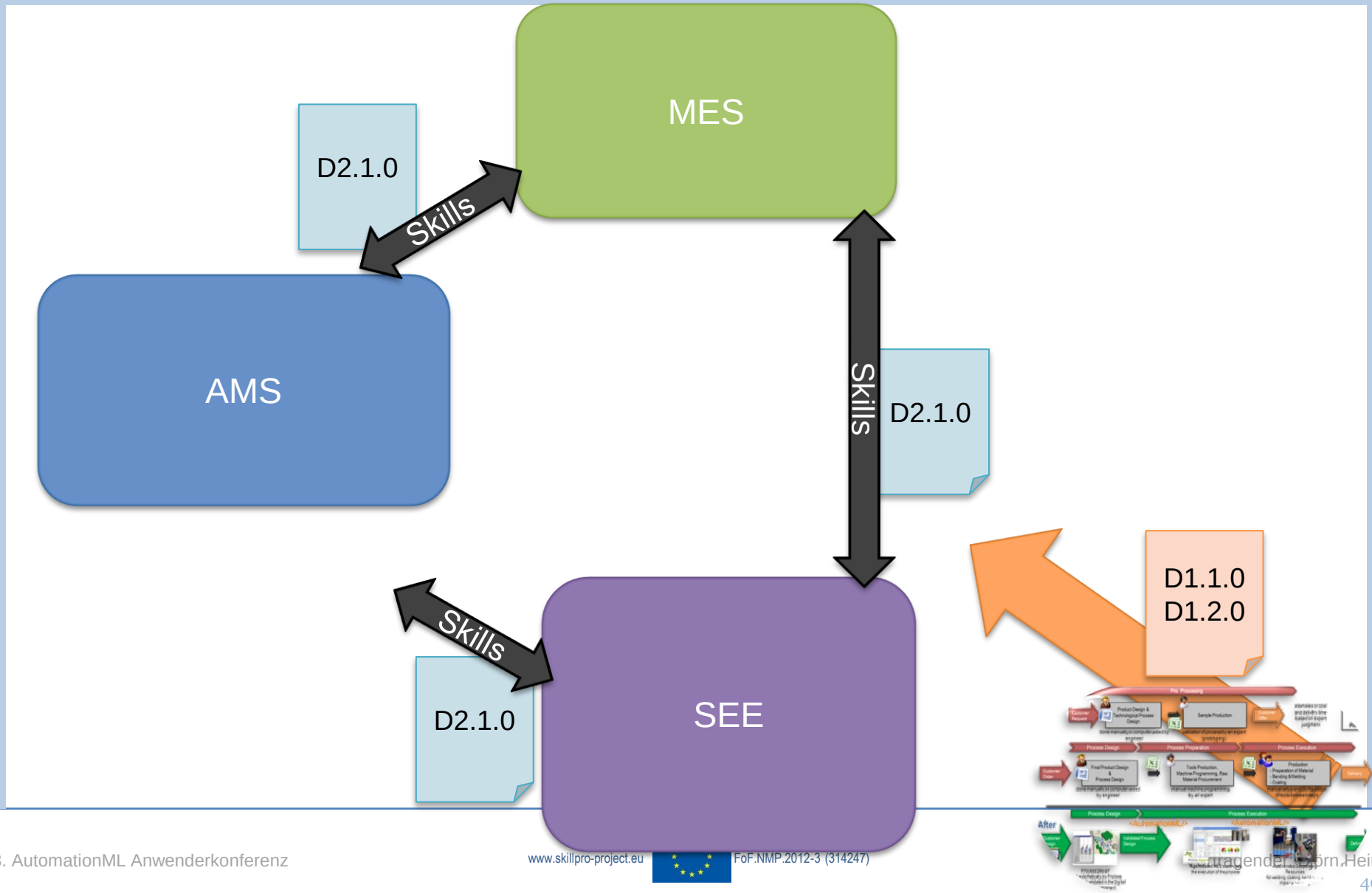
WP-Leader	Mike Ludwig
Signature	
TB-Member	Björn Hein
Signature	

SkillPro - D1.1.0 Page 1

D1.1.0
D1.2.0



Concept – where to get information



Concept – where to get informatio

MES

Evaluation Toolkit

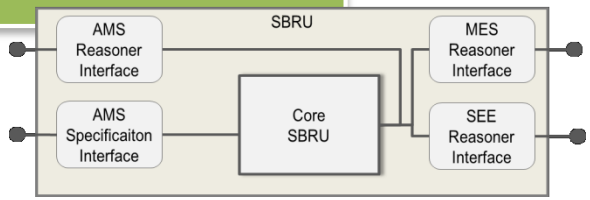
Reasoning Unit

D2.1.0

Skills

D2.3.0

OPC-UA



AMS

OPC-UA

OPC-UA Server

Skills

D2.1.0

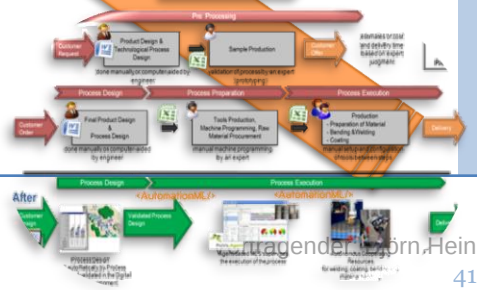
OPC-UA

D1.1.0
D1.2.0

Skills

D2.1.0

SEE



Concept – where to get informatio

MES

Evaluation Toolkit

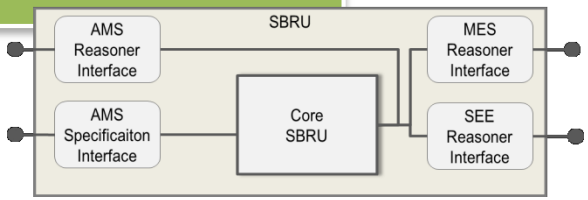
Reasoning Unit

D2.1.0

Skills

D2.3.0

OPC-UA



AMS

OPC-UA

OPC-UA Server

Skills

D2.1.0

AML Model Manager

D2.2.0

Skill Server

Production System Configuration Management

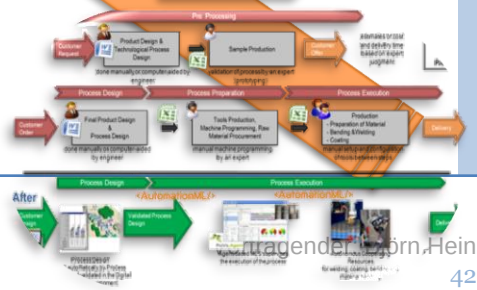
OPC-UA

Skills

D2.1.0

SEE

D1.1.0
D1.2.0



Concept – where to get informatio

MES

Evaluation Toolkit

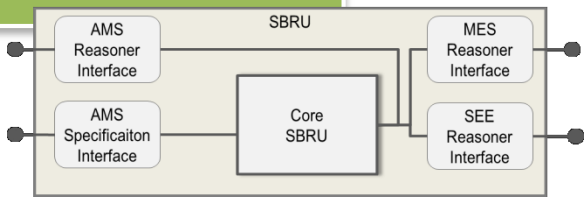
Reasoning Unit

D2.1.0

Skills

D2.3.0

OPC-UA



AMS

OPC-UA

OPC-UA Server

Skills

D2.1.0

AML Model Manager

D2.2.0

Skill Server

Production System Configuration Management

OPC-UA

D2.4.0

D1.1.0
D1.2.0

Skills

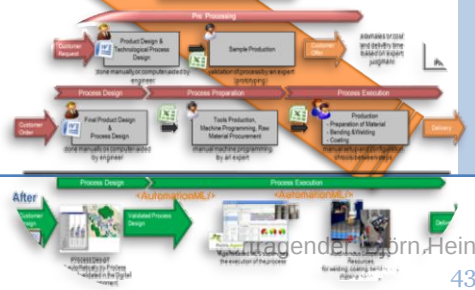
D2.1.0

SEE

SEE-Core (SBRC, Asset-Code-Gen, Skill-Repo,…)

Navigation

Object tracking



Skill-based Propagation of "Plug & Produce"-Devices
in Reconfigurable Production Systems by AML

3. AutomationML Anwenderkonferenz

Miriam Schleipen, Björn Hein, Julius Pfrommer, Kiril Aleksandrov,
Denis Stogl, Stefan Escaida, Jürgen Beyerer

www.skillpro-project.eu
FoF.NMP.2012-3 (314247)