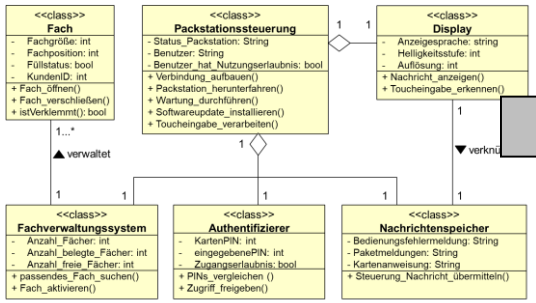
UML and RL on XPPU

Model = Code

Structural Model

Behavioral Model

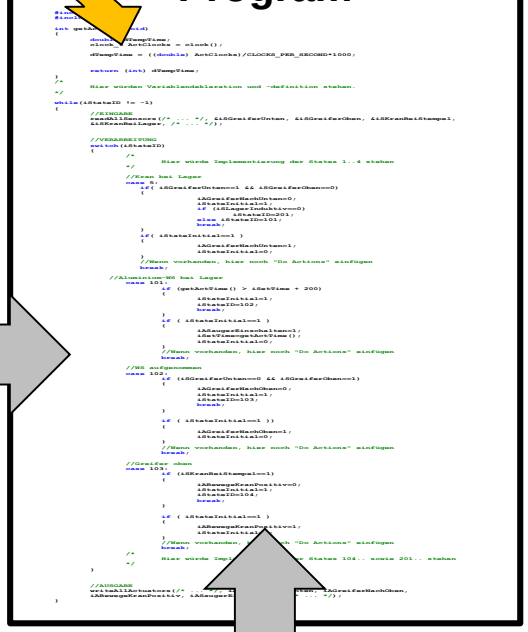
Class-Diagram



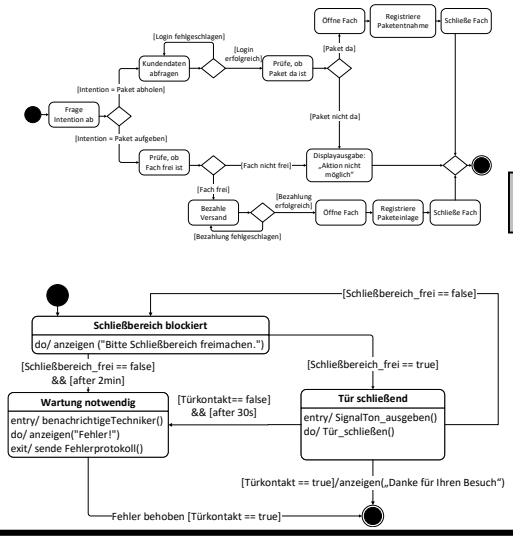
Declaration Section

```
typedef struct Lager_s {
      int WS_vorhanden;
      MATERIAL Material;
      HELLIGKEIT Helligkeit;
      Zylinder Schiebezylinder;
      Sensor Sensor_opt;
      Sensor Sensor_kap;
      Sensor Sensor_ind;
} Lager;

typedef enum {Aluminium, Kunststoff} MATERIAL;
typedef enum {hell, dunkel} HELLIGKEIT;

int Lager_WS_vereinzeln( void ) { /*...*/ }
MATERIAL Lager_WS_analysieren_Material( void ) { /*...*/ }
HELLIGKEIT Lager_WS_analysieren_Helligkeit( void ) { /*...*/ }
void Lager_Notaus( void ) {   /*...*/ }
int Lager_Init( void ) { /*...*/ }
```

Program



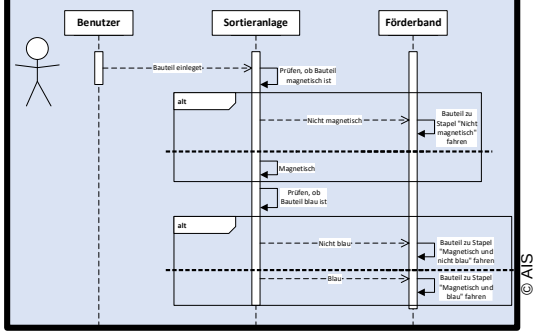
Activity- and State Diagramm



Implementation Section

```
//VERARBEITUNG
switch(iStateID)
{
  /*Hier würde Implementierung der ersten States
   stehen*/
  case 5: //Kran bei Lager
                                    //ausgehende Transition
     if( iSGreiferUnten == 1 && iSGreiferOben == 0)
     {
        iAGreiferNachUnten = 0;    //Exit Action
        iStateInitial = 1;         //für nächsten State
        if (iSLagerInduktiv == 0) //State wechseln
           iStateID = 201;
        else iStateID = 101;
        break;                     //Verlassen des States
      }
     if( iStateInitial == 1 )      //bei Betreten
     {
        iAGreiferNachUnten = 1;    //Entry Action
        iStateInitial = 0;
     }
     //Wenn vorhanden, hier noch "Do Actions" einfügen
     break;
  /*Hier würde Implementierung der nächsten States
   stehen*/
}
```

Testing via SD



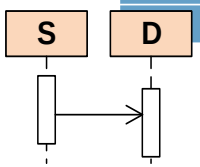
© AIS

Model-Based Product Design Approach

Requirements diagram

Identification of the requirements and the actors involved

Sequence diagram

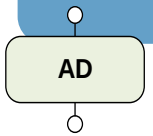


Description of the messages between the communication partners (actors)

Block Definition Diagram & Internal Block Diagram

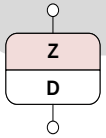
Description of the structure of the classes and their interfaces

Activity Diagram



Description of the behavior from the point of view of the mechanical engineer

State diagram



Description of the behavior from the point of view of the computer scientist

© AIS

Block Definition Diagram (BDD)

- How to structure the system?
- What components or subparts can be identified? Relations?
- Structuring with **BDD**

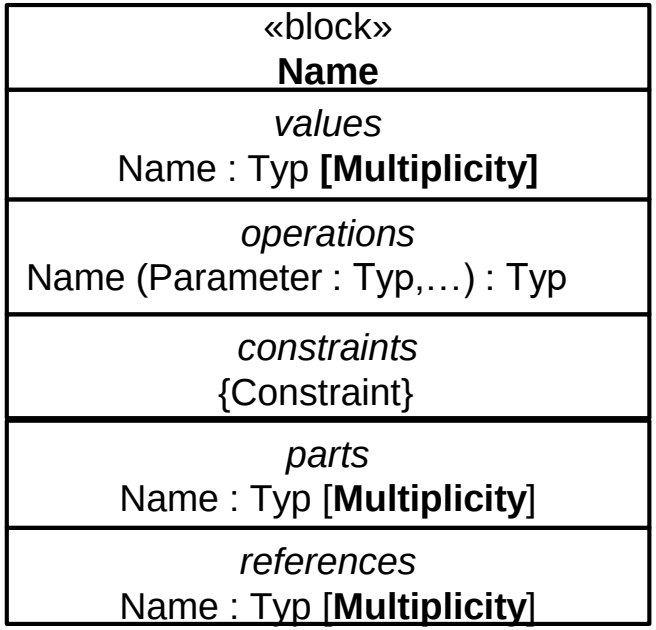
The structure of a system block is described in the block definition diagram.

- The Block Definition Diagram (BDD) provides a "black box" representation of a system block. In addition, it describes the hierarchy of the sub-blocks of the system block.
- The BDD can be compared to the first page of a construction manual of a furniture. It shows the type and number of all parts in the package.

rough design

System modules can be described optionally in more detail through additional properties:

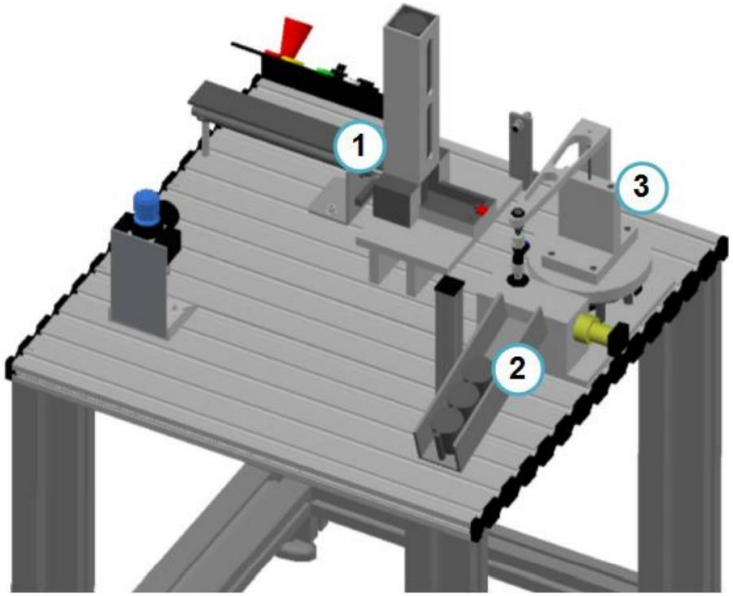
- **values**: Values describe specific physical, performance or other properties of a system block (e.g., weight, speed).
- **operations**: opportunities to influence the behavior of the system block (e.g. activities)
- **constraints**: Externally predetermined conditions, which are satisfied by the block
- **parts**: Describe the composition hierarchy of the system block
- **references**: References to other parts of the system; can be referenced by several system components simultaneously.

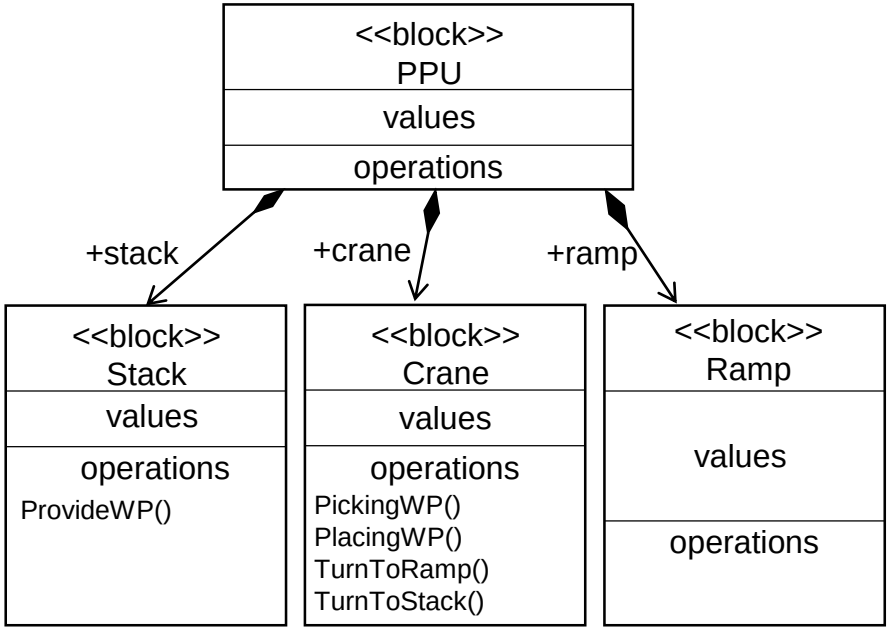


© AIS

The system modules (block) for more information include – optional properties

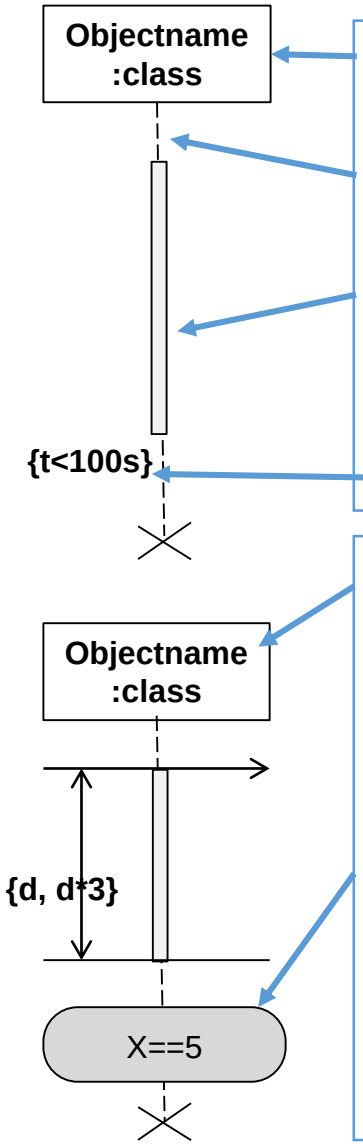
As an example scenario the PPU consists of a stack (no. 1) working as a work piece input storage could provide WPs, a ramp (no. 2) working as a work piece output storage and a crane (no. 3) for transporting work pieces by picking and placing them between these two working positions.





© AIS

Sequence Chart : notation elements



Object: the object that interacts with other objects

Lifeline: life line of the object

Execution / Activity: is the time when the object is instantiated and executes interactions

Time constraints: specify the system time requirement

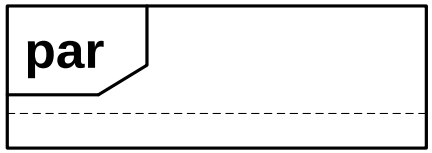
Component: is modular system part with transparent encapsulation of the content. It consists of elements with clear defined functionality, can be a standalone application an is describable as interface.

State Invariant: specifies a Boolean condition. It must be fulfilled, before the next event is reached.

Implicitly execution are the constructor and destructor of a class and changes of get- and set- attributes.

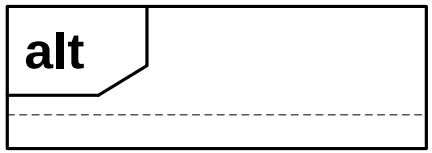
© AIS

Sequence Chart : Combined fragments and communication



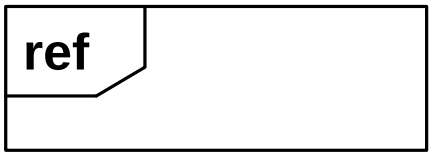
Parallel Processing

- Enables the modeling of fragments running in parallel



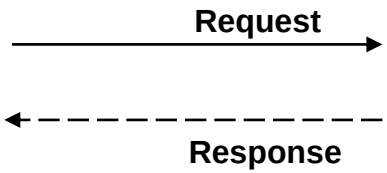
Alternative Processing

- Enables the modeling of fragments running alternatively



Interaction reference

- Describes a subsequence, defined by another sequence diagram (→"Black Box")



Synchronous communication *(closed arrowhead)*

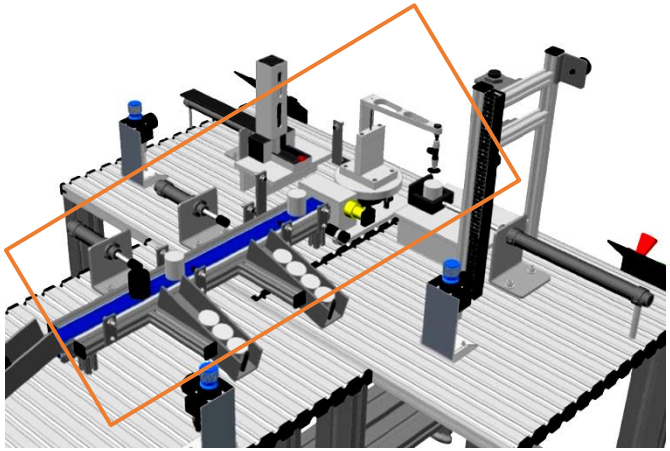
- The sender expects a response and cannot execute any processes until the response is received

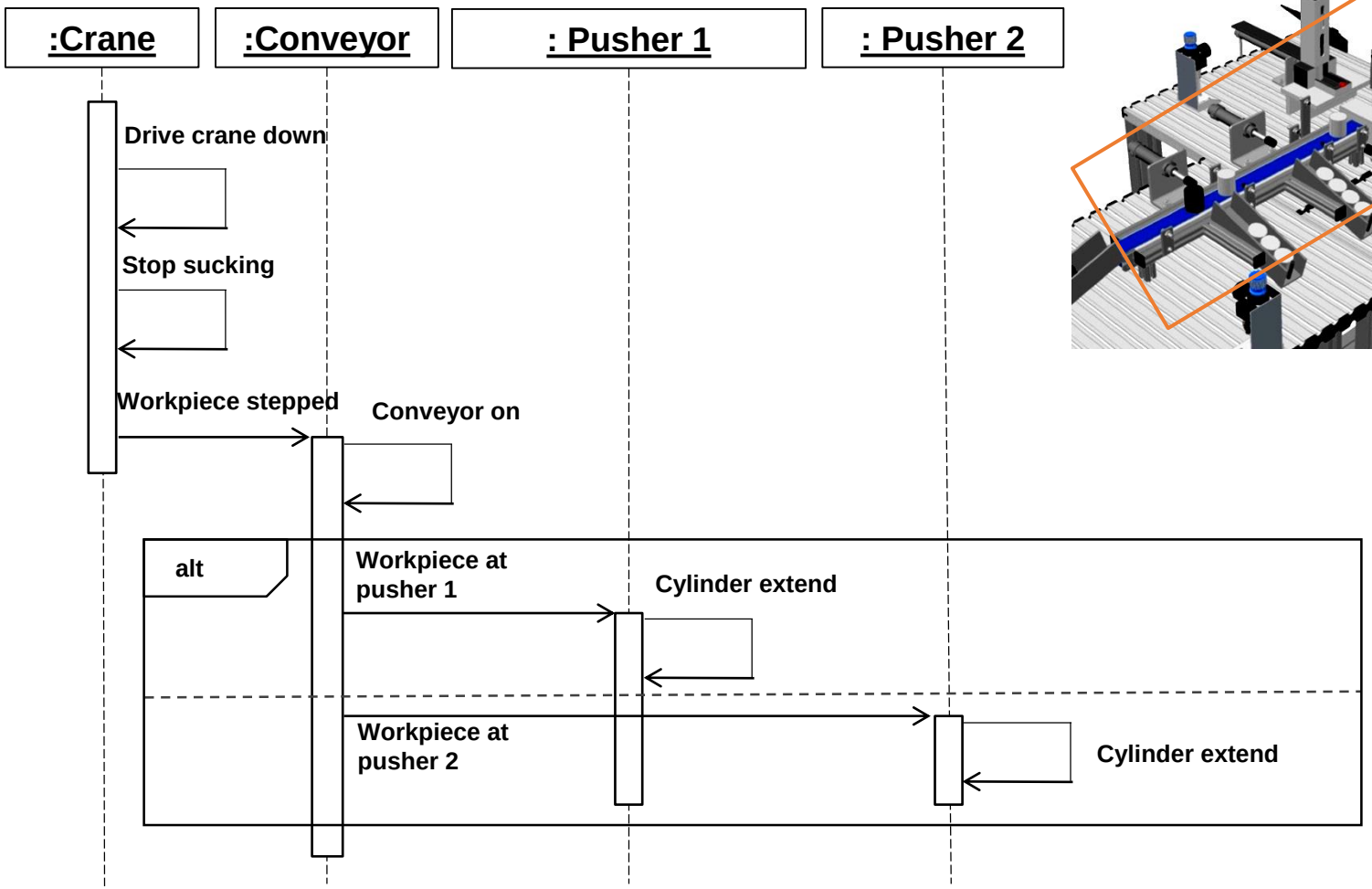
Asynchronous communication *(open arrowhead)*

- The sender does not expect a response. The sending and receiving of data is staggered and does not block any processes

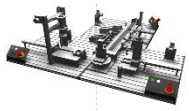
© AIS

Sequence chart: communication between crane and conveyor belt





© AIS

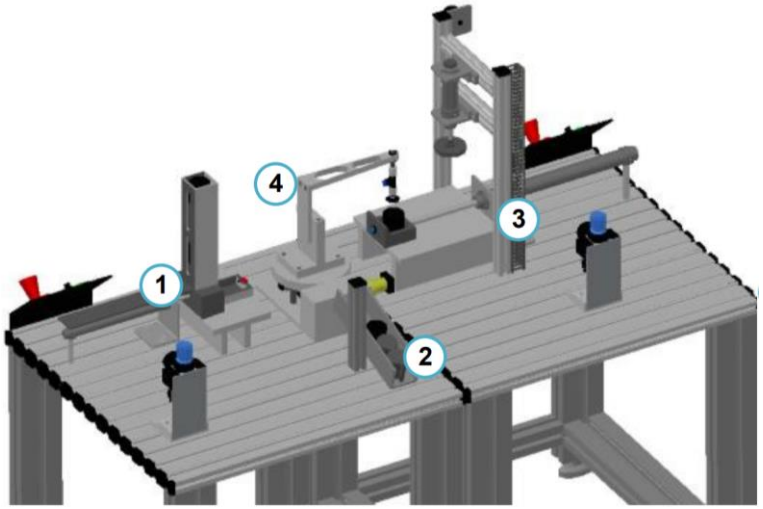


Exercise 1: Scenario Description on xPPU

Scenario description:

As shown in the figure below, the PPU consists of a stack (no. 1) working as a work piece input storage, a ramp (no. 2) working as a work piece output storage, a stamp (no. 3) for stamping work pieces and a crane (no. 4) for transporting work pieces by picking and placing them between these three working positions.

The PPU processes black plastic WPs as well as metallic WPs in this Scenario. Black WPs are separated at the stack and subsequently transported directly to the ramp. In contrast, metallic WPs are transported from the stack to the stamp, processed there and then transported to the ramp. Please make both BDD and sequence chart for this scenario.



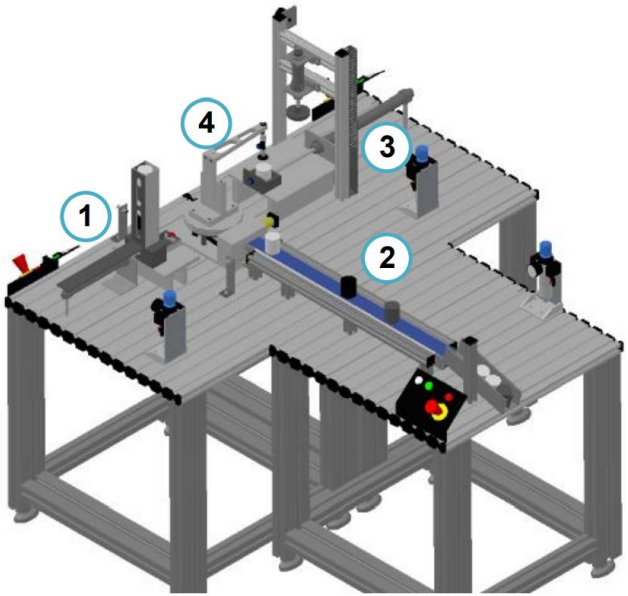
© AIS

Exercise 2: Scenario Description on xPPU

Scenario Description

As the figure below shows, the PPU consists of a stack (no.1) working as a work piece input storage, a conveyor (no. 2) working as a work piece output storage, a stamp (no. 3) for stamping work pieces and a crane (no. 4) for transporting work pieces by picking and placing them between these three working positions.

The PPU processes black plastic work pieces, white plastic work pieces as well as metallic work pieces in this Scenario. Black work pieces are separated at the stack and subsequently transported directly to the conveyor. In contrast, white plastic work pieces and metallic work pieces are transported from the stack to the stamp, processed there and then transported to the conveyor. The number of processed work pieces is less than 3. Please make both BDD and sequence chart for this scenario.



© AIS