Technical Report

Formal Modeling of Manufacturing Process using mCRL2

Festo 1- Workstation & Input (in1/out1)

Master Project 1 (WS 2024-25)

Master Industrial Informatics Degree

Report By:

Student 1: Jasprit Dhall
Matriculation-No.: 7025692

Student 2: Preeti Preeti
Matriculation-No.: 7024464

Student 3: Dheeraj Pal
Matriculation-No.: 7025303

Under supervision:

Engr. Prof. Gert Veltink and Jeffrey Wermann

Hochschule Emden/Leer · Industrial Informatics
Constantiaplatz 4 · 26723 Emden · http://www.hs-emden-leer.de

Table of Contents

1. Introduction

In the modern era of industrial automation and smart manufacturing, the integration of advanced learning systems plays a pivotal role in developing future-ready professionals. Festo Didactic, a renowned provider of industrial education and training solutions, offers a comprehensive framework for enhancing technical skills and knowledge in automation, mechatronics, and Industry 4.0 concepts.

This report delves into the specifications and implementation of the Festo Didactic system, particularly focusing on the development of a shop floor with predefined resources. The study involves resource specification, control unit design, and process optimization through advanced technologies and standards such as the Capability-Skill-Service (CSS) model and Asset Administration Shell (AAS). By structuring and analyzing different shop floor resources—such as workstations, sorting units, and transport mechanisms—this report provides insights into effective industrial training methodologies.

This report presents the process and results of implementing a path selection system for the Festo Didactic shop floor using the mCRL2 tool. The goal of this project was to define a model that enables efficient selection of paths from the input station (in1) to the output station (out1) based on predefined shop floor specifications. This ensures seamless movement of workpieces while avoiding bottlenecks and deadlocks.

The course "Algebraic Formal methods and tools" focuses on key industrial automation objectives, including system behavior modeling and path verification, by providing a comprehensive understanding of industrial automation and smart manufacturing systems. The intention is for students to gain hands-on experience in modeling and verifying system behaviors using formal tools like mCRL2 while learning to optimize manufacturing processes through workflow analysis.

This project aligns with the objectives of the course, which include understanding industrial automation, modeling system behaviors, and verifying the correctness of specified paths. The Festo Didactic shop floor serves as a real-world example, allowing us to apply theoretical concepts in system verification and path optimization. By using formal methods, the project demonstrates the application of modeling tools in industrial automation.

Furthermore, the study aims to establish an efficient and structured approach for defining production workflows, ensuring seamless integration of different manufacturing components. The report also highlights the significance of defining distinct production paths to optimize workflow efficiency. Through this structured approach, Festo Didactic serves as a blueprint for fostering innovation and efficiency in industrial automation training.

1.1. Shop Floor Resources and Their Functions:

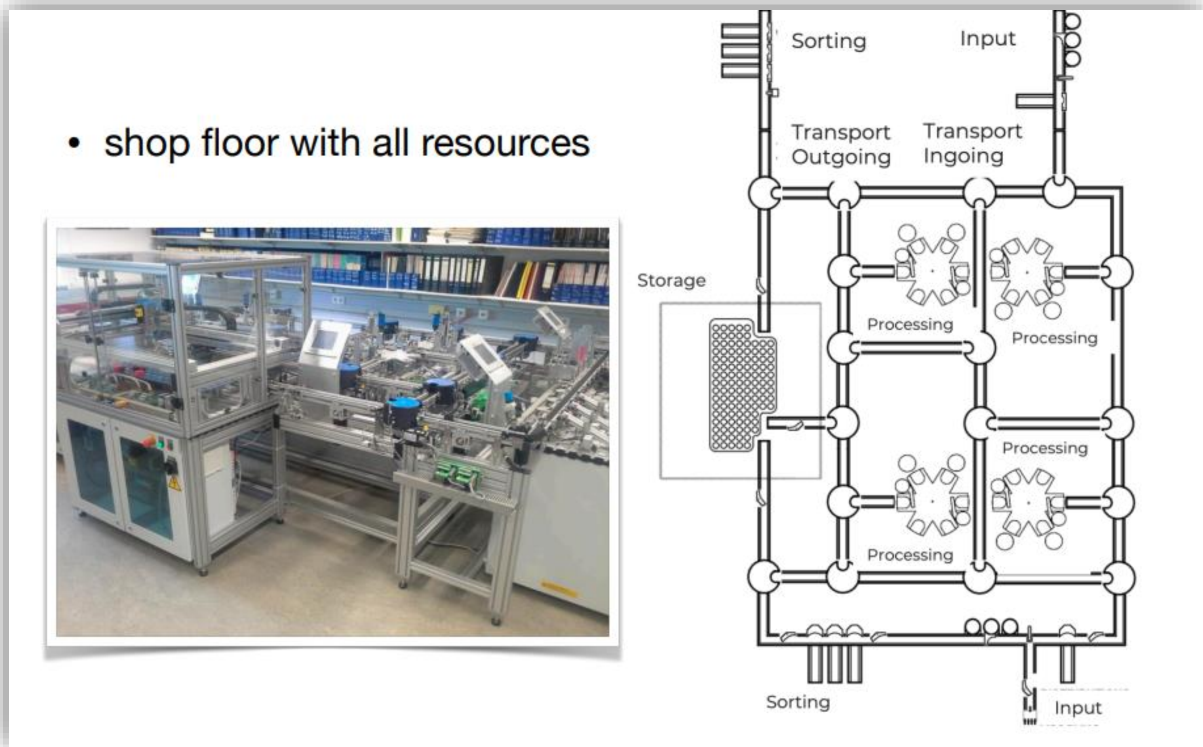

Image 01: The Manufacturing Model for Real-Time Data Processing

Festo Didactic is a Cyber Physical System training tool wherein using the pseudo form of shop floor and its resources, we could implement industry 4.0 technologies and observe real time changes. This tool can be used best for analysing and optimizing production flows.

1.2.1. Workstations

They Perform **processing tasks** such as assembly, inspection, or machining. Workpieces may need to pass through specific workstations based on their processing requirements.

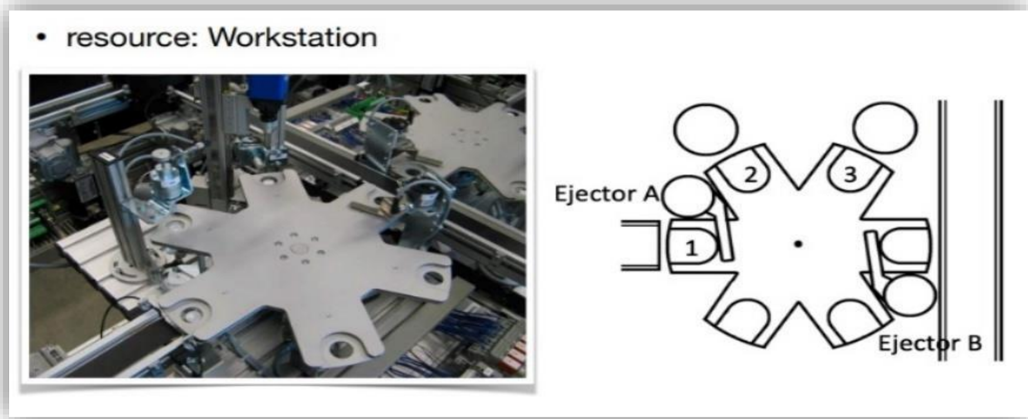

Image 02: Workstation of the Manufacturing Model

1.2.2. Conveyors

It **Transport workpieces** between different stations and determines **available routes** and ensures smooth movement between different locations. Each conveyor has two primary types of conveyor sensors used are **Start Sensors** and **End Sensors**. which detects the **presence and position of workpieces** on the shop floor and Provides **real-time data** for workpiece tracking and ensures correct path execution.

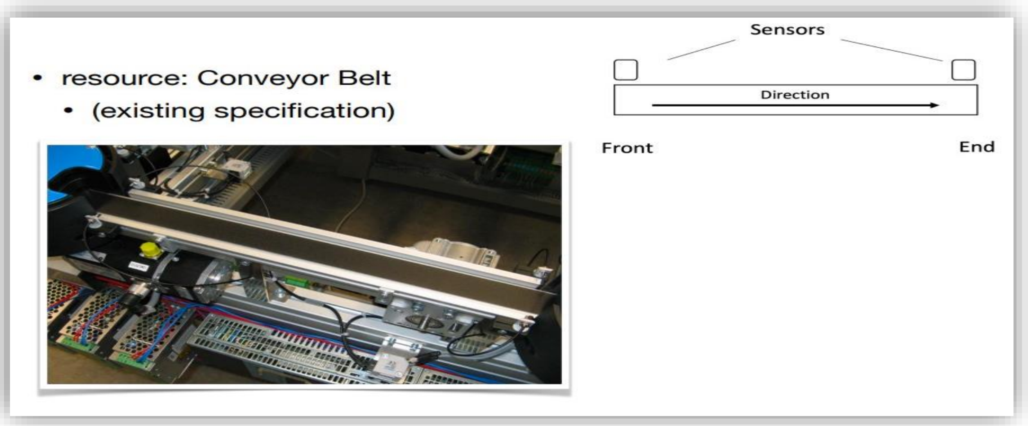

Image 03: Conveyor of the Manufacturing Model

1.2.3. Sorting Units

It Identify and **redirect workpieces** based on predefined conditions (e.g., type, size, or processing status). Also helps in **decision-making** for routing workpieces to the correct processing stations.

1.2.4. Switches

It helps in Changing the **direction of workpiece movement** to follow a specified path and allows for **dynamic rerouting** based on workflow requirements.

Image 04: Switch of the Manufacturing Model

2. Methodology

The methodology adopted in this study follows a structured approach to designing, modeling, and verifying the Festo Didactic shop floor system using mCRL2. The key steps include understanding the shop floor structure, modeling the system components, implementing path selection logic, and validating the model through formal verification techniques.

2.1. Understanding the Festo Didactic Shop Floor

The Festo Didactic system consists of multiple interconnected resources designed to facilitate efficient automation and material flow. Key components include workstations for assembly, processing, and quality control, conveyors for transporting workpieces, sorting units for directing workpieces based on predefined conditions, and switches for dynamically altering their paths. Additionally, sensors are integrated to detect the presence and position of workpieces, while storage zones temporarily hold materials during processing. Each of these elements plays a critical role in maintaining smooth production cycles. To accurately represent their interactions and behaviors in mCRL2, a detailed mapping of each component's function and workflow was developed.

2.2. Modeling in mCRL2

To formally specify the shop floor operations, mCRL2 was used to define the system's structure and behavior through a structured modeling approach. First, individual process components such as conveyors, workstations, and switches were specified, each with a set of valid transitions representing workpiece movement and execution. Next, key actions like move, detect, process, and change position were declared, with transition conditions ensuring a logical production flow. Synchronization and communication rules were then established to enable parallel execution of system components and ensure proper interaction between workpieces and resources. Finally, constraint enforcement techniques, including deadlock prevention and logical validation checks, were applied to maintain continuous workflow and ensure all workpieces reach their intended destinations efficiently.

2.3. Path Selection Logic Implementation

Path selection in the mCRL2 model was implemented using a decision-making algorithm to optimize workpiece movement. The process began by identifying all possible routes from in1 to out1, taking into account system constraints such as processing priorities and congestion avoidance. The optimal path was then determined by analyzing the transition graph to select the shortest and most efficient route. Additionally, alternative paths were defined to handle potential blockages or rework requirements, ensuring flexibility in the system. Finally, the model was rigorously tested under various conditions to verify that all workpieces followed the correct route, ensuring smooth and accurate execution of the manufacturing process.

2.4. Graph Generation and Analysis

The mCRL2 tool was used to generate a graph representation of the shop floor model, enabling comprehensive analysis and optimization. First, a state-space exploration was conducted to visualize all possible transitions, where nodes represented different system states and edges depicted valid workpiece movements. Graph analysis techniques were then applied to identify redundant transitions and inefficiencies, leading to optimizations that improved throughput and reduced processing time. Additionally, the system was examined for deadlocks and bottlenecks by detecting unreachable states or congestion points. Any inefficiencies found were addressed by modifying transition rules, ensuring a smooth and efficient manufacturing process.

2.5. Model Validation and Verification

To ensure the correctness of the Festo Didactic shop floor model, formal verification techniques were applied through simulation testing, property verification, and performance analysis. Multiple simulations were conducted to observe workpiece flow, with results compared against expected workflow patterns to validate system behavior. Logical properties, such as ensuring every workpiece reaches an output station, were rigorously verified, along with constraints preventing resource conflicts. Additionally, performance analysis was carried out by evaluating key parameters like processing time, transition efficiency, and deadlock resolution. These verification steps ensured that the model operated accurately, efficiently, and without unintended bottlenecks.

3. Knowledge about mCLR2 tool

3.1. Overview of the mCRL2 Tool

mCRL2 (Micro Common Representation Language 2) is a formal specification language used for modeling, analyzing, and verifying concurrent systems. It provides a robust mathematical framework to describe the behavior of distributed and parallel systems. mCRL2 is widely used in industrial automation, software verification, and communication protocols to ensure correctness, efficiency, and reliability of system models.

3.2. Key Features of mCRL2:

- Process Algebra-Based Modeling: Allows concise and structured system definitions.
- State Space Exploration: Generates graphs and state transitions to visualize workflows.
- Verification and Validation: Ensures systems conform to predefined constraints and correctness properties.
- Concurrency Handling: Models simultaneous actions efficiently using parallel composition.
- Deadlock Detection: Identifies blocking states in automated workflows.

mCRL2 helps formalize the shop floor operations of Festo Didactic by defining actions such as workpiece movement, transitions, and sensor interactions. This ensures that the selected control path (In1 to Out1) adheres to correctness principles and operates efficiently.

3.3. What is LPS and LTS?

3.3.1. LPS (Linear Process Specification)

LPS (Linear Process Specification) is an intermediate representation used in mCRL2 to describe the stepwise execution of systems through actions, conditions, and transitions. It encapsulates process definitions and data expressions that define the system's states, providing a structured format for modeling complex behaviors. By representing these behaviors in a formal way, LPS enables the conversion of system descriptions into verification models, which can then be used for further state-space analysis and exploration, aiding in the verification and validation of system properties.

3.3.2. LTS (Labeled Transition System)

LTS (Labeled Transition System) is a graphical representation of a system’s behavior, where nodes correspond to system states and edges depict transitions between states based on defined actions. It provides a comprehensive overview of all possible system executions, making it a vital tool for verifying correctness properties, detecting deadlocks, and optimizing workflows. Generated from the Linear Process Specification (LPS), LTS helps analyze system behavior by visualizing state transitions, ensuring that processes operate smoothly and efficiently within the defined constraints.

3.3.3. Role of LPS and LTS in This Project

The Festo Didactic shop floor model was initially specified in mCRL2, after which it was converted into an LPS representation to define all possible system transitions. This LPS representation was then used to generate an LTS (Labelled Transition System) graph, offering a visual depiction of how workpieces move and interact with various components within the system. The generated LTS was thoroughly analyzed to validate path selection, ensure operational efficiency, and optimize the overall performance of the system, contributing to the identification of potential improvements and areas for optimization.

By leveraging mCRL2, LPS, and LTS, this study successfully modeled and validated the Festo Didactic shop floor, ensuring that the control path from In1 to Out1 operates efficiently and without deadlocks.

4. Selected Control Path: In1 to Out1

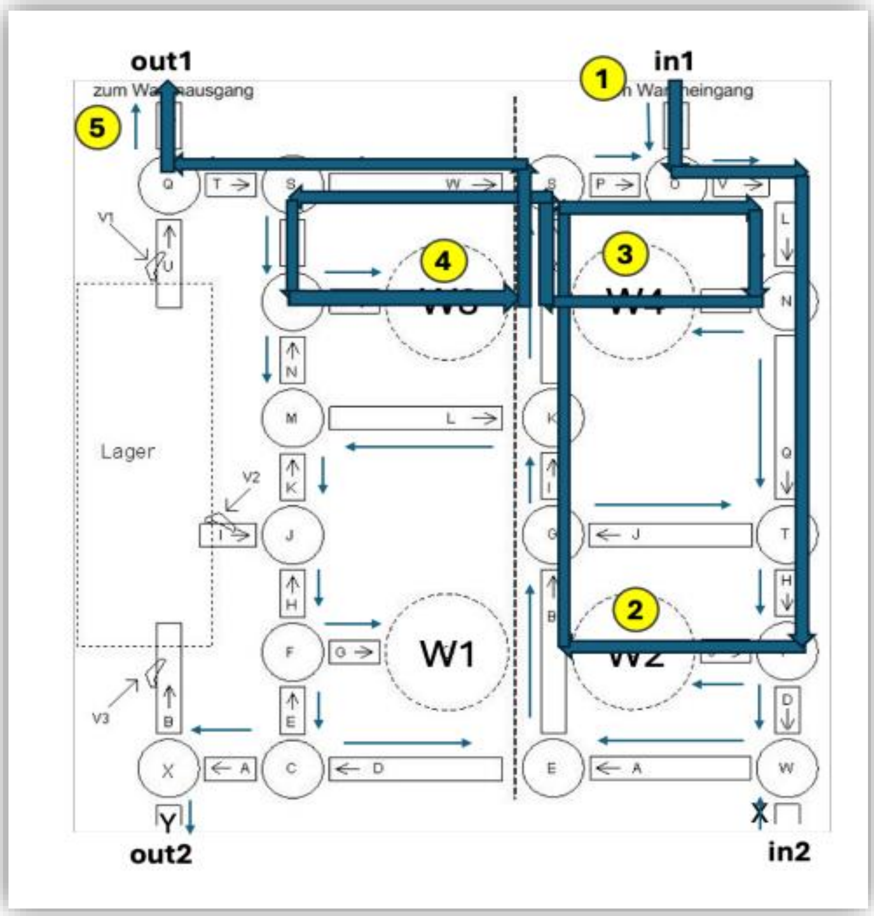

Image 05: Control Path

For this study, we have selected the first control path, which directs the workpieces from the input station (in1) to the output station (out1). This path represents the most fundamental and direct route within the Festo Didactic shop floor, ensuring an efficient and smooth workflow while preventing unnecessary detours and congestion. As part of the process, the workpieces pass through three key workstations: **W2 for positioning and drilling, W4 for mounting and fixing, and W3 for cleaning and polishing** before reaching the final output station (out1). This structured approach ensures that each workpiece undergoes the required processing steps in a sequential and optimized manner, contributing to an efficient and high-quality manufacturing workflow.

4.1. Path Flow and Key Stages

The system begins with the entry of a new workpiece at in1, where it is detected by sensors and assigned a processing sequence based on predefined rules. The conveyor belt system then activates, transporting the workpiece to the first switching station, where the system determines whether further processing is needed or if the workpiece can be routed directly. Switch positions are dynamically adjusted to guide the workpiece along the selected path, and it may be rerouted for additional quality control if required. At the workstations, processing units such as drilling, assembly, or inspection modules perform their functions on the workpiece. Once processing is complete, the workpiece is transported toward out1, with the system ensuring smooth transitions to prevent bottlenecks. Finally, the workpiece exits at out1, and the system logs its successful completion, updating records for quality assurance..

4.2. Justification for Path Selection

The system is designed with a focus on efficiency, ensuring that the shortest and most direct routes are selected to minimize both travel time and overall system load. Error prevention is built into the model through well-defined transitions and constraints, which help avoid incorrect routing and prevent potential deadlocks from occurring. Scalability is also a key consideration, as the control path serves as a baseline that can be easily expanded in future iterations, allowing for the addition of new routes and modifications as needed. Furthermore, the mCRL2 model plays a crucial role in the verification and validation process, successfully ensuring that the selected path adheres to the expected rules and constraints, operating without deviations. This combination of efficiency, error prevention, scalability, and rigorous verification guarantees a robust and adaptable system.

5. Asset Administration Shell (AASX) in the Project

The Asset Administration Shell (AASX) is a standardized digital representation of physical assets in Industry 4.0 environments. It provides a structured framework for managing, monitoring, and optimizing industrial processes by integrating digital twins, machine-readable data, and real-time communication protocols.

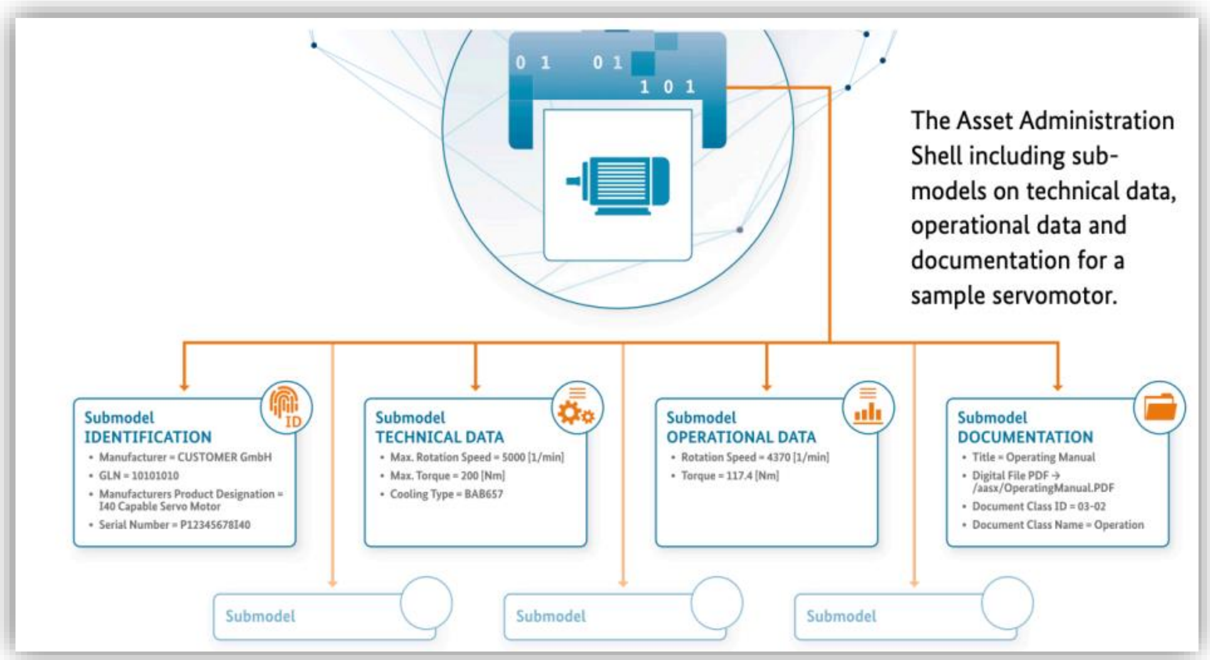

Image 06: AASx model and submodels, Basic representation

AASX (Asset Administration Shell eXchange) plays a vital role in the Festo Didactic shop floor by enabling a digital twin implementation that provides a real-time representation of physical resources, including workstations, conveyors, and sensors. It facilitates seamless data exchange and interoperability between various automation systems and controllers, ensuring smooth communication across the entire system. AASX also supports process monitoring and optimization, allowing for real-time tracking of workpiece movement and the detection of potential bottlenecks in the production workflow. Additionally, it enhances system scalability by allowing the shop floor model to expand in the future without requiring significant modifications, ensuring long-term adaptability and growth.

By leveraging AASX, the project benefits from greater automation, better decision-making, and enhanced efficiency in managing industrial resources and optimizing path selection within the system.

6. Code Explanation

The mCRL2 specification for the Festo Didactic shop floor outlines the states and transitions, modeling the interactions of various resources and the movement of workpieces throughout the system. The key components of the code include process definitions that specify behaviors for workstations, conveyors, and switches, actions and transitions that describe workpiece movement, detection, and processing, and synchronization rules that ensure cohesive interaction between system components.

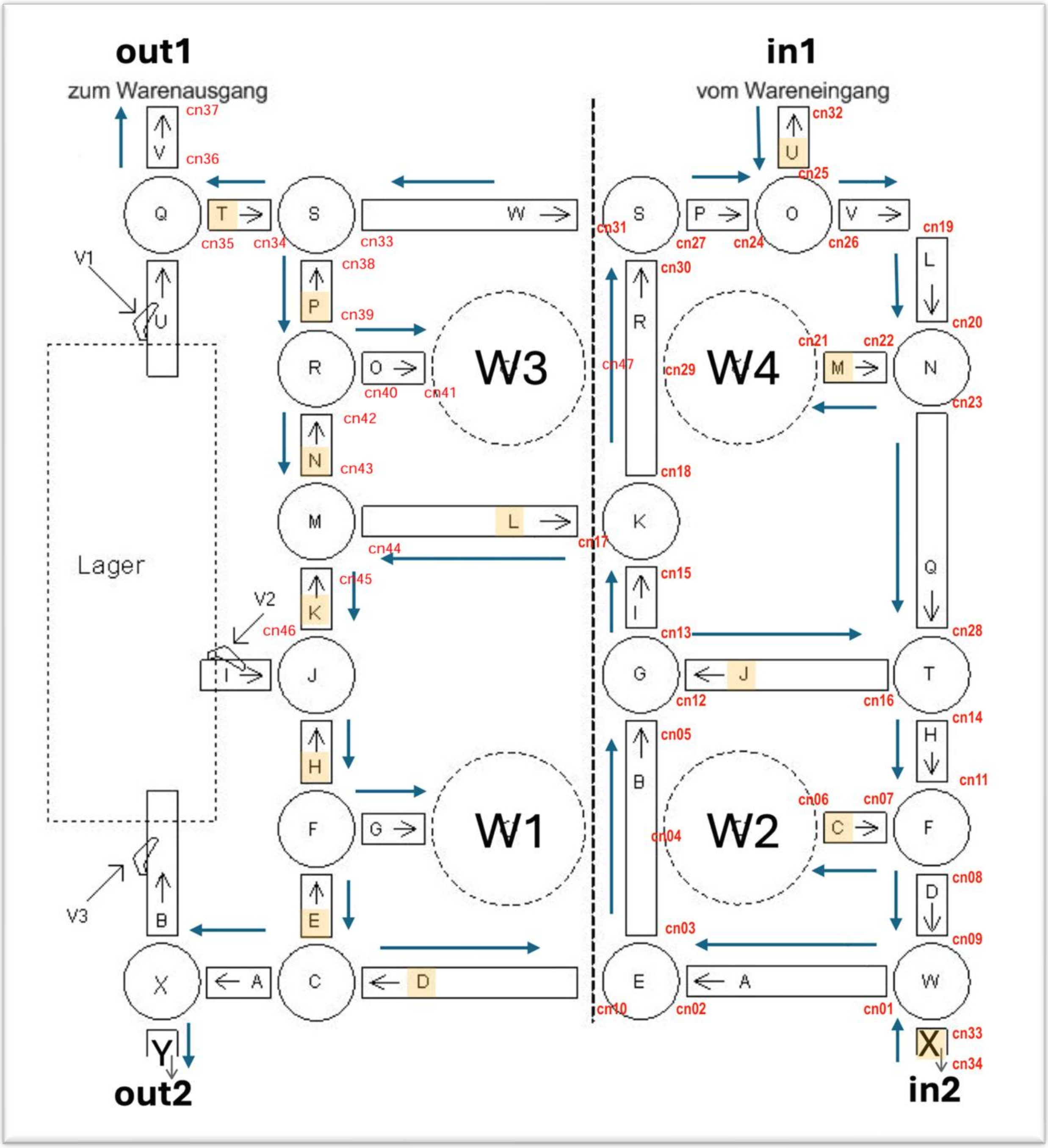

Image 07: Connections named by team

6.1. Sorts (Data Types)

6.1.1 Error:

This sort defines possible errors in the system, such as a workpiece being dropped (`ERR_WORKPIECE_DROPPED`) or multiple workpieces occupying the same cell (`ERR_CELL_OVERLOADED`).

6.1.2. ID, Connections and Directions:

```
sort ID = struct
    icbA | icbB | icbC | icbD | icbH | icbI | icbJ | icbL | icbM | icbP | icbQ | icbR | icbU | icbV
    | ocbA | ocbB | ocbD | ocbE | ocbG | ocbH | ocbI | ocbK | ocbL | ocbN | ocbO | ocbP | ocbT
| ocbU | ocbV | ocbW
    | iswE | iswF | iswG | iswK | iswN | iswO | iswS | iswT | iswW
    | oswC | oswF | oswJ | oswM | oswR | oswS | oswQ | oswX
;
sort Connection = struct
    cn00  % special unused default/undefined connection
    | cn01 | cn02 | cn03 | cn04 | cn05 | cn06 | cn07 | cn08 | cn09 | cn10
    | cn11 | cn12 | cn13 | cn14 | cn15 | cn16 | cn17 | cn18 | cn19 | cn20
    | cn21 | cn22 | cn23 | cn24 | cn25 | cn26 | cn27 | cn28 | cn29 | cn30
    | cn31 | cn32 | cn33 | cn34 | cn35 | cn36 | cn37 | cn38 | cn39 | cn40
    | cn41 | cn42 | cn43 | cn44 | cn45 | cn46 | cn47 | cn48 | cn49 | cn50
;
sort Direction = struct FORWARD?isFW | BACKWARD?isBW;
```

- `ID` defines unique identifiers for components like conveyor belts (`icb`, `ocb`) and switches (`isw`, `osw`).

- `Connection` defines possible connection points in the system (e.g., `cn01`, `cn02`).

- `Direction` defines the movement direction of conveyor belts (`FORWARD` or `BACKWARD`).

6.1.3. AAS (Asset Administration Shell)

```
sort AAS = struct
    _AAS(
        sm_id: SM_Identification,
        sm_td: SM_TechnicalData,
        sm_od: SM_OperationalData,
        sm_cf: SM_Configuration
    )?isDefinedAAS
    | undefAAS?isUndefinedAAS
;
```

- `AAS` represents the digital twin of a physical asset, containing identification (`sm_id`), technical data (`sm_td`), operational data (`sm_od`), and configuration (`sm_cf`).

6.1.4. WorkPiece, Cell, ConveyorBelt, Switch

```
sort WorkPiece = struct _Workpiece(wpid: WPId, colour: Colour, orientation: Orientation);
sort Cell = struct _Cell(workpiece: WorkPiece)?hasWorkPiece | emptyCell?isEmpty | errorCell?isError;
sort ConveyorBelt = List(Cell);
sort Switch = struct _Switch(position: SwitchPosition, cell: Cell);
```

- `WorkPiece` represents a workpiece with an ID, color, and orientation.

- `Cell` represents a cell on a conveyor belt, which can contain a workpiece, be empty, or be in an error state.

- `ConveyorBelt` is a list of cells.

- `Switch` represents a switch with a position and a cell.

6.2. Maps (Functions)

6.2.1. Conveyor Belt Operations

```
map
   moveForward: ConveyorBelt -> ConveyorBelt;
   moveBackward: ConveyorBelt -> ConveyorBelt;
   insertFront: ConveyorBelt # WorkPiece -> ConveyorBelt;
```

These functions define operations on conveyor belts, such as moving forward/backward, inserting/removing workpieces, and checking for workpieces.

6.2.2. Switch Operations

```
map
   changePosition: Switch # SwitchPosition -> Switch;
   load: Switch # WorkPiece -> Switch;
   unload: Switch -> Switch;
```

These functions define operations on switches, such as changing positions, loading/unloading workpieces, and setting content.

6.2.3. AASx files:

```
map
aas': AASMap;
eqn
aas' =
_AASMap(undefAAS) % default unused configuration!
[icbU ->
_AAS(
_SM_Identification(icbU, INTERNAL_ID, RESOURCE, [i,n,U], FESTO,
CONVEYOR_BELT),
_SM_TechnicalData([M,o,d,e,l,_1], [S,N,_0,_0,_1], CONVEYOR_BELT,
FESTO, 3),
_SM_OperationalData_ConveyorBelt(ACTIVE, FORWARD),
_SM_Configuration_ConveyorBelt(cn25, cn32, cn00)
)
]
[icbV ->
_AAS(
_SM_Identification(icbV, INTERNAL_ID, RESOURCE, [i,n,V], FESTO,
CONVEYOR_BELT),
_SM_TechnicalData([M,o,d,e,l,_1], [S,N,_0,_0,_1], CONVEYOR_BELT,
FESTO, 3),
_SM_OperationalData_ConveyorBelt(ACTIVE, FORWARD),
_SM_Configuration_ConveyorBelt(cn26, cn19, cn00)
……
```

Mapping AASx files for all relevant conveyors and switches.

6.3. Actions

```
dummyAct;                    % dummy action
```

Actions define observable events in the system, such as errors (`error`) or dummy actions (`dummyAct`).

6.4. Processes

6.3.1. Control Path Definition

```
proc Control =
manualLoad(icbU, cn32, _wp1)
. move(icbU, BACKWARD)
. detectWorkPieceAt(icbU, cn25)
. changePosReq(iswO, pos3)
. detectWorkPieceAt(icbV, cn26)
. changePosReq(iswO, pos2)
. move(icbV, FORWARD)
```

i. A workpiece is manually loaded onto a conveyor belt (manualLoad).
ii. The conveyor belt moves the workpiece forward or backward (move).
iii. Sensors detect the workpiece at specific locations (detectWorkPieceAt).
iv. Switches change positions (changePosReq) to redirect the workpiece.
v. The workpiece is processed at the next conveyor belt or workstation (process).
vi. Once the workpiece moves to next conveyor belt, the switch goes back to default receving postion
vii. Process goes on till workpiece reached out1 and is unloaded manually.

6.3.2. System Definition

```
proc System_ =
   % Conveyor Belts
   ConveyorBelt(aas'(icbU)) ||
   ConveyorBelt(aas'(icbV)) ||
   ...
   % Switches
   Switch(aas'(iswO)) ||
   Switch(aas'(iswN)) ||
```

- Calling for AAS files of all involved conveyor belts and switches.

6.3.3. Controlled System Definition

```
proc System =
  hide({
    transfer, dummyAct
  },
  allow({
    error, manualLoad, manualUnload, workPieceAt, transfer, changePos, move
  },
  comm({
    unload | load -> transfer,
    dummyAct | dummyAct -> dummyAct
  },
    System_
  )))
```

- `System` adds communication and hides internal actions.

```
proc ControlledSystem =
  hide({
    dummyAct
  },
  allow({
    error, manualTransfer, detectedWorkPieceAt, changePosAck, move
  },
  comm({
    workPieceAt | detectWorkPieceAt -> detectedWorkPieceAt,
    manualLoad | manualLoad -> manualTransfer,
    manualUnload | manualUnload -> manualTransfer,
    changePos | changePosReq -> changePosAck,
    dummyAct | dummyAct -> dummyAct
  },
    Control || System
  )))
```

- `ControlledSystem` combines the `Control` process with the `System` to create a fully controlled system. Calling this proc using init function creates LPS and LTS files.

7. Graph Analysis and Interpretation

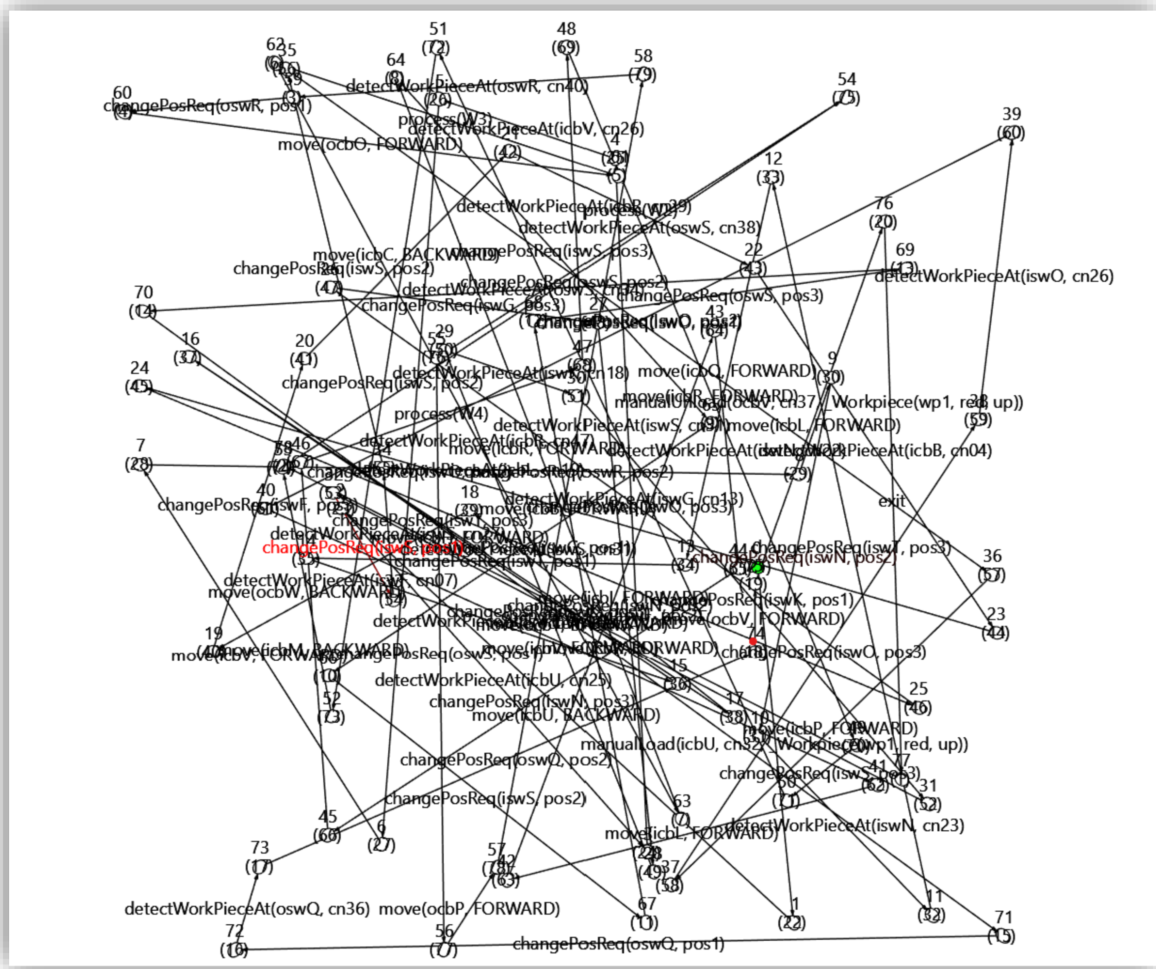

Image 08: LTS Graph of Control Path

7.1. Understanding the Graph Representation

The mCRL2-generated graph is a state-space representation of the shop floor model, providing a visual and mathematical depiction of all possible transitions a workpiece can take. This graph was derived from the **Labeled Transition System (LTS)**, which was generated by applying **linearization in stack** to the **Labeled Process Specification (LPS)**. The **linearization process** simplifies the original specification into a structured format, enabling efficient state-space exploration and graph construction.

Node and Edge Interpretations-

- Nodes: Represent distinct system states, including workpiece positions and control system conditions.
- Edges: Define valid transitions based on movement rules, conveyor interactions, and system constraints.
- Start Node: Represents the initial state when a workpiece enters the system at in1.
- End Node: Marks the final state when the workpiece successfully exits at out1.

7.2. Observations from the Graph Analysis

The graph effectively identifies multiple available paths, including both optimal and suboptimal routes, providing a comprehensive view of the system's behavior. Since the graph was generated using **LPS linearization**, it provides a structured representation of all possible system transitions, ensuring **formal verification** of the workflow.

Key insights from the analysis:

- **Bottlenecks** and inefficient paths were detected and mitigated by adjusting the code to optimize the workflow.
- **Deadlocks** were analyzed and prevented by refining transition rules and constraints, ensuring smooth operation throughout the system.
- The correctness of the system was validated, confirming that no workpiece becomes stuck or follows an unintended route, ensuring efficient and reliable execution.

7.3. Enhancing Workflow Through Graph Optimization

The system incorporates several strategies to enhance its performance and reliability. Deadlock resolution ensures that all transitions lead to an active system state, preventing workpieces from getting stuck. Load balancing was implemented to avoid over-reliance on specific workstations, thus boosting overall system throughput. Alternative routing strategies were identified to provide fallback routes in cases where a workstation becomes unavailable, ensuring continuous operation. Additionally, cycle time reduction was prioritized by analyzing and optimizing paths to minimize workpiece travel time, significantly improving system efficiency and productivity.

By generating the **LTS graph from LPS using linearization in stack**, we were able to **verify system behavior** using formal methods, ensuring correctness and efficiency in the Festo Didactic shop floor model.

7.4. Application of Graph Analysis for Future Improvements

The system integrates advanced technologies to improve performance and adaptability. Predictive maintenance uses real-time data to detect potential inefficiencies before they impact system operations, allowing for proactive interventions. AI-based optimization applies machine learning techniques to dynamically suggest the most efficient routing decisions, improving overall workflow and reducing delays. Additionally, advanced scalability measures have been implemented, expanding the graph model to accommodate additional resources and flexible manufacturing setups, ensuring the system remains adaptable and capable of growing with future demands.

By utilizing graph analysis techniques, this project ensures that the selected control path from In1 to Out1 operates with minimal delay, maximum efficiency, and optimal resource utilization.

8. Results

8.1. Path Visualization & Optimization

The graph visualizations effectively demonstrated how workpieces move through the system. It enabled real-time verification of different paths to identify optimal routes.

8.2. Verification and Accuracy Testing

The system underwent formal verification through state space exploration to validate all possible transitions, ensuring that every potential path was thoroughly checked for accuracy. Any inconsistencies between the intended and actual workpiece paths were addressed, eliminating errors and improving reliability. This rigorous validation process provided correctness assurance, confirming that the final implementation aligns with the expected production workflows and functions as intended, delivering a dependable and efficient system.

8.3. Performance & Efficiency Analysis

The system optimization focused on improving processing speed by identifying high-speed and slow-moving routes to ensure efficient workflow. Transition minimization was implemented to reduce unnecessary state changes, streamlining the overall process and enhancing system responsiveness. Additionally, load balancing was incorporated to prevent workstation overloads, utilizing smart routing logic that ensures a more even distribution of tasks across the system, ultimately improving performance and maintaining operational balance.

9. Conclusion

This project represents a comprehensive exploration of formal methods and advanced digital tools in the context of industrial automation, specifically focusing on the Festo Didactic shop floor system. By leveraging the mCRL2 toolset, the team successfully modeled, analyzed, and optimized the system's workflow, ensuring efficient and error-free movement of workpieces from the input station (In1) to the output station (Out1). The project's outcomes highlight the critical role of formal verification techniques in designing reliable, scalable, and adaptive manufacturing systems, aligning with the principles of Industry 4.0.

The detailed methodology adopted in this study involved the structured modeling of shop floor components—such as workstations, conveyors, switches, and sorting units—using mCRL2. This included defining data types, actions, and processes to represent the system's behavior formally. The implementation of a path selection algorithm ensured optimal workpiece movement, while the use of Linear Process Specification (LPS) and Labeled Transition System (LTS) provided a robust framework for visualizing and analyzing the system's state space. The generated graph representations allowed for the identification of bottlenecks, redundancies, and potential deadlocks, which were subsequently addressed to enhance workflow efficiency.

One of the key strengths of this project lies in its integration of the Asset Administration Shell (AASX), which enabled the creation of a digital twin for the shop floor. This facilitated real-time data exchange, process monitoring, and decision-making, thereby improving system interoperability and adaptability. The AASX framework not only enhanced the system's performance but also ensured scalability, allowing for future expansions and modifications without significant structural changes.

The results of the project underscore the effectiveness of formal verification techniques in ensuring system correctness and reliability. Through rigorous testing and validation, the team confirmed that all workpieces followed the intended paths without deviations, and potential issues such as deadlocks and bottlenecks were successfully mitigated. The optimization strategies implemented, including load balancing, transition minimization, and cycle time reduction, significantly improved the system's overall efficiency and throughput.

Looking ahead, there are several avenues for future exploration and enhancement. The integration of dynamic path selection algorithms, driven by real-time data and machine learning techniques, could further optimize workflow efficiency and adapt to changing production demands. Additionally, the development of physical prototypes and hardware integration would provide valuable insights into the system's behavior in real-world scenarios. The incorporation of predictive maintenance systems and AI-based decision-making tools could also enhance the system's resilience and adaptability, ensuring continuous operation even under unforeseen circumstances.

In summary, this project serves as a testament to the transformative potential of formal methods in industrial automation. By combining theoretical rigor with practical application, the team has demonstrated how advanced tools like mCRL2 and AASX can be used to design, verify, and optimize manufacturing workflows. The insights gained from this study provide a robust foundation for future research and development in smart manufacturing, paving the way for more efficient, reliable, and innovative industrial systems.

10. References

1. **Prof. Dr. G.J. Veltink,** *Lectures in Festo Didactic, Master in Industrial Informatics, Hochschule Emden Leer.*

2. **Jeffrey Wermann,** *Lectures in IDTT, , Master in Industrial Informatics, Hochschule Emden Leer.*

3. **Dr. Wilfried stoll**, *Festo Didactic,Inspiring Technical Education*

4. **Jan Friso Groote,** *The mCRL2 Toolset for Analysing Concurrent Systems, Eindhoven University of Technology*.

5. https://www.mcrl2.org/web

6. **Prof. Dr. Armando Walter Colombo. Lectures in Digitalization and Virtualization of ICPS.,** *Master in industrial informatics Course, Hochschule Emden Leer (Germany), 2024. DIN SPEC 91345*

7. **M.Eng. Mart´ın Alejandro Bar. Workshop on Asset Administration Shell**. *Master in Industrial Informatics ¨ Course, Hochschule Emden Leer(Germany), 2024. Asset Administration Shell- Practice Part*

8. **Colombo, A. W., Karnouskos, S., & Mendes, J. M**. (2015*). Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach. Springer.*

9. **Groote, J. F., & Mousavi, M. R.** (2014). *Modeling and Analysis of Communicating Systems*. MIT Press.

11. Contribution of Team members

Here's a structured table detailing each member's contributions to the project:

Group Member	**Research & Documentation**	**Coding in mCRL2**	**Results and Conclusions**	**Report Writing & Presentation**
Jasprit	Conducted literature review, studied Festo Didactic system	Planned control path and defined connection and logic	Verified correctness, optimized system behavior of LTS	Wrote methodology, and results. Also, improved final report.
Preeti	Researched Asset Administration Shell (AAS), compiled technical references	Assisted in AASx definitions	Worked on LPS and LTS generation.	Wrote introduction, and other details. Supported in formatting.
Dheeraj	Gathered references for industrial automation applications	Updated missing information and checked for errors.	Analyzed transition states and bottlenecks.	Worked on graph interpretations, conclusions and references. Also compiled and formatted report.