

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**

# **ISA 95 compatible MES**

**Gonalo Rafael Barreto de Magalhães**



Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Mário Jorge Rodrigues de Sousa

July 16, 2026



# **ISA 95 compatible MES**

**Gonçalo Rafael Barreto de Magalhães**

Mestrado em Engenharia Eletrotécnica e de Computadores

Approved in oral examination by the committee:

President: Prof. Armando Jorge Miranda de Sousa

Referee: Prof. Jaime Francisco Cruz Fonseca

Referee: Prof. Mário Jorge Rodrigues de Sousa

July 16, 2026

# Resumo

Esta dissertação aborda o desenvolvimento e a implementação de um Manufacturing Execution System (MES) genérico, baseado na norma ISA-95, enquadrado no contexto Industry 4.0. Na indústria moderna, o MES assume um papel central como camada de integração entre os sistemas de planeamento ao nível empresarial e os sistemas de controlo do shop-floor. Todavia, uma parte significativa das soluções atualmente disponíveis apresenta limitações decorrentes da sua forte orientação para domínios específicos, da sua integração em frameworks extensas de Enterprise Resource Planning (ERP) ou da sua focalização em funcionalidades isoladas, tais como o planeamento ou a monitorização. Neste contexto, torna-se evidente a necessidade de soluções MES mais leves, baseadas na norma ISA-95, que aliem modularidade a um elevado grau de flexibilidade, permitindo a sua adaptação a diferentes ambientes produtivos.

O principal objetivo deste trabalho consiste no desenvolvimento de um MES compatível com a norma ISA-95, com capacidade para realizar o escalonamento da produção, bem como o controlo e a monitorização em tempo real dos processos produtivos. O sistema é concebido para funcionar de forma genérica em múltiplos cenários de shop-floor, privilegiando uma abordagem baseada na configuração através de uma base de dados, em detrimento da utilização de lógica de controlo rigidamente codificada. A validação do sistema proposto será efetuada recorrendo ao simulador de shop-floor disponibilizado pelo Departamento de Engenharia Eletrotécnica e de Computadores (DEEC).

Com base nos modelos de referência, frameworks e fluxos de informação definidos pela norma ISA-95, este trabalho propõe uma arquitetura de natureza modular, na qual um algoritmo de escalonamento genérico processa informação diretamente a partir de uma base de dados normalizada. Paralelamente, será desenvolvida uma interface gráfica de utilizador que possibilite a configuração do sistema, bem como a visualização, em tempo real, do estado da produção.

Ao disponibilizar uma aplicação MES adaptável, modular e em conformidade com a norma ISA-95, esta dissertação propõe uma plataforma reutilizável para fins de experimentação e de apoio ao ensino de MES e de informática industrial. Adicionalmente, contribui para a discussão científica em curso sobre o papel das normas, nomeadamente a ISA-95, no suporte a arquiteturas de MES de nova geração, concebidas para responder aos desafios colocados pela Industrial Internet of Things (IIoT).

# Abstract

This dissertation explores the design and implementation of a generic Manufacturing Execution System (MES) that follows the ISA-95 standard in the context of Industry 4.0. Modern manufacturing relies on MES as the bridge between enterprise planning and shop-floor control systems. However, many existing solutions are either highly oriented for a specific domain, embedded in heavy Enterprise Resource Planning (ERP) frameworks, or focus on isolated tasks such as planning or monitoring. Therefore, there is a lack of lightweight, standards-based MES cores that are both modular and flexible enough to be adapted to different production environments.

The proposed work has the goal of developing an ISA-95 compatible MES capable of production scheduling, control, and real-time monitoring. The system is designed to operate generically across multiple shop-floor scenarios by relying on the configuration of data models rather than hard-coded logic. The system will be tested and validated using the Departamento de Engenharia Eletrotécnica e Computadores (DEEC) shop-floor simulator.

Building on the frameworks and data flows defined by ISA-95, this project defines a modular architecture where a generic scheduling algorithm processes information directly from the standardized database. A graphical interface will be developed to allow for the system configuration and real-time production status visualization.

By delivering an adaptable and modular ISA-95 compatible MES application, this work provides a reusable platform for experimentation, supports MES and industrial informatics teaching, and contributes to the ongoing discussion on how standards like ISA-95 can help the next-generation, Industrial Internet of Things (IIoT) ready MES architectures.

# UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework aimed at achieving a better and more sustainable future for all. This initiative encompasses 17 goals<sup>1</sup> designed to address the most pressing global challenges, calling for action to protect the planet while promoting shared prosperity.

This dissertation proposes the development of a generic and modular Manufacturing Execution System (MES) based on the ISA-95 standard and designed for the Industry 4.0 paradigm. By promoting the modernization of shop-floor infrastructures and production efficiency, this project directly contributes to the following goals:

**SDG 4 - Quality Education** Ensure inclusive and equitable quality education and promote life-long learning opportunities for all

**SDG 9 - Industry, Innovation and Infrastructure** Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

**SDG 12 - Responsible Consumption and Production** Ensure sustainable consumption and production patterns

**SDG 4 - Quality Education:** Beyond its industrial applications, the developed platform assumes a significant academic commitment. Designed in based on the DEEC laboratory simulator, this system functions as a reusable educational platform. The project supports the teaching of industrial informatics, equipping students and researchers with practical tools to understand new-generation MES systems and their associated international standards.

**SDG 9 - Industry, Innovation and Infrastructure:** The widespread adoption of Industry 4.0 requires agile and innovative information systems. By developing a flexible MES architecture that bridges the gap between enterprise planning and shop-floor automation through protocols like OPC UA and database-driven configurations, this work supports the development of resilient technological infrastructures. This approach contributes to industrial innovation and promotes the adoption of the Industrial Internet of Things (IIoT), providing tools that can be adapted by companies of different scales.

**SDG 12 - Responsible Consumption and Production:** The core of an MES relies on continuous improvement and efficiency. Through scheduling algorithms and strict real-time control of production processes, the developed system helps optimize the allocation of equipment and raw materials. Efficient digital coordination minimizes waste, reduces shop-floor resource downtime, and fosters industrial management practices focused on minimizing production losses.

---

<sup>1</sup><https://sdgs.un.org/goals>

| <b>SGD</b> | <b>Target</b> | <b>Contribution</b>                                                                                                                            | <b>Performance Indicators and Metrics</b>                                                       |
|------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| 4          | 4.4           | Validation and integration with laboratory simulators result in an open-source resource that qualifies future engineers in these technologies. | Successful deployment of the MES in academic scenarios (DEEC) for teaching and experimentation. |
| 9          | 9.4           | The development of a generic, modular MES based on the ISA-95 standard enables a seamless and integrated transition to Industry 4.0.           | System integration via OPC UA; Support for standard-based, variable shop-floor scenarios.       |
| 12         | 12.2          | Automated dispatching and control ensure a continuous flow of materials and reduce equipment downtime.                                         | Workflow optimization; Accurate mapping of tools and processes within ISA-95 models.            |

# Acknowledgements

This project marks the conclusion of one of the most important stages of my life, and it would not have been possible without the support, encouragement, and guidance of the people who have accompanied me throughout this journey. I would like to express my sincere gratitude to all of them.

First and foremost, I would like to thank my supervisor, Professor Mário Jorge Rodrigues de Sousa, for his patience, guidance, and invaluable knowledge throughout the development of this dissertation. His continuous support and insightful advice were essential to the successful completion of this project.

I would also like to thank my friends and family for their constant encouragement, understanding, and support. Their motivation and the many moments we shared helped me overcome the challenges and celebrate the achievements along the way.

Finally, I would like to express my deepest gratitude to my parents, Mónica and Miguel, for their unconditional love and support, and endless encouragement throughout my life. Their belief in me, during both the good times and the difficult ones, has shaped the person I am today. I am forever grateful for everything they have done for me.

To everyone who, in one way or another, contributed to this important stage of my life, thank you. Your support has meant more than words can express.

Gonçalo Rafael Barreto de Magalhães

# Contents

|          |                                                                 |           |
|----------|-----------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                             | <b>1</b>  |
| 1.1      | Context . . . . .                                               | 1         |
| 1.2      | Motivation . . . . .                                            | 1         |
| 1.3      | Problem Statement . . . . .                                     | 2         |
| 1.4      | Objectives . . . . .                                            | 2         |
| 1.5      | Dissertation Structure . . . . .                                | 3         |
| <b>2</b> | <b>Background</b>                                               | <b>4</b>  |
| 2.1      | Industry 4.0 . . . . .                                          | 4         |
| 2.1.1    | Reference Architectural Model Industry 4.0 (RAMI 4.0) . . . . . | 4         |
| 2.2      | ISA-95 Standard Overview . . . . .                              | 5         |
| 2.3      | ISA-95 Object and Activity Models . . . . .                     | 8         |
| 2.4      | Manufacturing Execution Systems . . . . .                       | 10        |
| 2.5      | Production Scheduling Fundamentals . . . . .                    | 10        |
| <b>3</b> | <b>State-of-the-art</b>                                         | <b>12</b> |
| 3.1      | Commercial MES Solutions . . . . .                              | 12        |
| 3.2      | Free and Open-Source MES Tools . . . . .                        | 13        |
| 3.3      | Academic MES and MOM Architectures . . . . .                    | 14        |
| 3.4      | ISA-95 Based MES . . . . .                                      | 15        |
| 3.5      | Critical Synthesis and Research Gap . . . . .                   | 16        |
| <b>4</b> | <b>ISA-95 Data Models</b>                                       | <b>17</b> |
| 4.1      | Introduction . . . . .                                          | 17        |
| 4.2      | ISA-95 Information Models Adaptation . . . . .                  | 17        |
| 4.3      | Equipment Model . . . . .                                       | 17        |
| 4.3.1    | Equipment Class . . . . .                                       | 18        |
| 4.3.2    | Equipment Class Property . . . . .                              | 19        |
| 4.3.3    | Equipment . . . . .                                             | 19        |
| 4.3.4    | Equipment Property . . . . .                                    | 19        |
| 4.4      | Physical Asset Model . . . . .                                  | 20        |
| 4.4.1    | Physical Asset . . . . .                                        | 21        |
| 4.4.2    | Physical Asset Property . . . . .                               | 21        |
| 4.4.3    | Physical Asset Class . . . . .                                  | 22        |
| 4.4.4    | Physical Asset Class Property . . . . .                         | 22        |
| 4.5      | Material Model . . . . .                                        | 22        |
| 4.5.1    | Material Class . . . . .                                        | 23        |
| 4.5.2    | Material Class Property . . . . .                               | 24        |

|          |                                                                  |           |
|----------|------------------------------------------------------------------|-----------|
| 4.5.3    | Material Definition . . . . .                                    | 24        |
| 4.5.4    | Material Definition Property . . . . .                           | 25        |
| 4.5.5    | Material Lot . . . . .                                           | 25        |
| 4.5.6    | Material Lot Property . . . . .                                  | 26        |
| 4.6      | Process Segment Model . . . . .                                  | 27        |
| 4.7      | Operations Definition Model . . . . .                            | 27        |
| 4.8      | Operations Schedule Model . . . . .                              | 28        |
| <b>5</b> | <b>System Design and Implementation</b>                          | <b>30</b> |
| 5.1      | Overall View of the System . . . . .                             | 30        |
| 5.2      | Database Implementation . . . . .                                | 32        |
| 5.2.1    | Equipment Model . . . . .                                        | 33        |
| 5.2.2    | Dynamic Scheduling and Production Inventory . . . . .            | 35        |
| 5.3      | Graphical User Interface . . . . .                               | 36        |
| 5.4      | MES Core Program (Program MES) . . . . .                         | 39        |
| 5.4.1    | Package Structure and Modular Organization . . . . .             | 40        |
| 5.4.2    | Sequential Workflow and Component Interaction . . . . .          | 40        |
| 5.4.3    | OPC UA Communication and Four-Stage Handshake Protocol . . . . . | 47        |
| <b>6</b> | <b>Validation and Results</b>                                    | <b>48</b> |
| 6.1      | Experimental Setup . . . . .                                     | 48        |
| 6.2      | Case Study and Validation Dataset . . . . .                      | 48        |
| 6.2.1    | Simulator Layout . . . . .                                       | 49        |
| 6.2.2    | Simulator Information Data . . . . .                             | 50        |
| 6.3      | Data Pipeline and XML Ingestion Validation . . . . .             | 55        |
| 6.4      | Dispatch Plan Generation and Scheduling . . . . .                | 57        |
| 6.5      | Sequential PLC Control and Handshake Robustness . . . . .        | 58        |
| 6.6      | Discussion and Limitations . . . . .                             | 59        |
| <b>7</b> | <b>Conclusion and Future Work</b>                                | <b>61</b> |
| 7.1      | Conclusion . . . . .                                             | 61        |
| 7.2      | Future Work . . . . .                                            | 62        |
|          | <b>References</b>                                                | <b>65</b> |
| <b>A</b> | <b>ISA-95 Data Models</b>                                        | <b>68</b> |
| A.1      | Process Segment . . . . .                                        | 68        |
| A.1.1    | Process Segment . . . . .                                        | 68        |
| A.1.2    | Equipment Segment Specification . . . . .                        | 68        |
| A.1.3    | Equipment Segment Specification Property . . . . .               | 69        |
| A.1.4    | Physical Asset Segment Specification . . . . .                   | 69        |
| A.1.5    | Physical Asset Segment Specification Property . . . . .          | 70        |
| A.1.6    | Material Segment Specification . . . . .                         | 70        |
| A.1.7    | Material Segment Specification Property . . . . .                | 71        |
| A.1.8    | Process Segment Parameter . . . . .                              | 72        |
| A.1.9    | Process Segment Dependency . . . . .                             | 72        |
| A.2      | Operations Definition . . . . .                                  | 73        |
| A.2.1    | Operations Definition . . . . .                                  | 73        |
| A.2.2    | Operations Segment . . . . .                                     | 74        |

|          |                                                 |           |
|----------|-------------------------------------------------|-----------|
| A.2.3    | Parameter Specification . . . . .               | 74        |
| A.2.4    | Equipment Specification . . . . .               | 75        |
| A.2.5    | Equipment Specification Property . . . . .      | 75        |
| A.2.6    | Physical Asset Specification . . . . .          | 75        |
| A.2.7    | Physical Asset Specification Property . . . . . | 76        |
| A.2.8    | Material Specification . . . . .                | 76        |
| A.2.9    | Material Specification Property . . . . .       | 77        |
| A.2.10   | Operations Segment Dependency . . . . .         | 78        |
| A.3      | Operations Schedule . . . . .                   | 78        |
| A.3.1    | Operations Schedule . . . . .                   | 78        |
| A.3.2    | Operations Request . . . . .                    | 79        |
| A.3.3    | Segment Requirement . . . . .                   | 80        |
| A.3.4    | Segment Parameter . . . . .                     | 81        |
| A.3.5    | Equipment Requirement . . . . .                 | 82        |
| A.3.6    | Equipment Requirement Property . . . . .        | 82        |
| A.3.7    | Physical Asset Requirement . . . . .            | 82        |
| A.3.8    | Physical Asset Requirement Property . . . . .   | 83        |
| A.3.9    | Material Requirement . . . . .                  | 83        |
| A.3.10   | Material Requirement Property . . . . .         | 84        |
| <b>B</b> | <b>Database Table Structure</b>                 | <b>86</b> |
| B.1      | Equipment . . . . .                             | 86        |
| B.2      | Physical Asset . . . . .                        | 87        |
| B.3      | Material . . . . .                              | 88        |
| B.4      | Process Segment . . . . .                       | 89        |
| B.5      | Operations Definition . . . . .                 | 90        |
| B.6      | Operations Schedule . . . . .                   | 91        |
| <b>C</b> | <b>User Interface</b>                           | <b>94</b> |
| C.1      | Navigation Flow . . . . .                       | 94        |
| C.1.1    | Navigation . . . . .                            | 94        |
| C.1.2    | Equipments Tab . . . . .                        | 95        |
| C.1.3    | Physical Asset Tab . . . . .                    | 96        |
| C.1.4    | Material Tab . . . . .                          | 98        |
| C.1.5    | Process Segment Tab . . . . .                   | 99        |
| C.1.6    | Operations Definition Tab . . . . .             | 101       |
| C.1.7    | Operations Schedule Tab . . . . .               | 103       |
| C.2      | Index - Home Page . . . . .                     | 105       |
| C.3      | Equipment . . . . .                             | 106       |
| C.3.1    | Equipment Class Create Form . . . . .           | 107       |
| C.3.2    | Equipment Create Form . . . . .                 | 108       |
| C.3.3    | Equipment Detail Page . . . . .                 | 108       |
| C.3.4    | Equipment Property Form . . . . .               | 108       |
| C.4      | Process Segment . . . . .                       | 109       |
| C.4.1    | Process Segment Form . . . . .                  | 110       |
| C.4.2    | Process Segment Detail Page . . . . .           | 110       |
| C.4.3    | Equipment Segment Specification Form . . . . .  | 111       |
| C.4.4    | Material Segment Specification Form . . . . .   | 111       |

|          |                                                      |            |
|----------|------------------------------------------------------|------------|
| <b>D</b> | <b>MES Core Program</b>                              | <b>113</b> |
| D.1      | Package Structure and Modular Organization . . . . . | 113        |
| <b>E</b> | <b>Validation and Results</b>                        | <b>114</b> |
| E.1      | Transformations Diagrams . . . . .                   | 114        |

# List of Figures

|      |                                                                     |    |
|------|---------------------------------------------------------------------|----|
| 2.1  | RAMI 4.0, from [3]                                                  | 5  |
| 2.2  | ISA-95 model, from [1]                                              | 5  |
| 2.3  | Data Flow - Functional Model, from [4]                              | 7  |
| 2.4  | Activity Relationship, from [5]                                     | 8  |
| 2.5  | Role based equipment model, from [6]                                | 9  |
| 2.6  | Material model, from [6]                                            | 9  |
| 4.1  | Role based equipment model, from [6]                                | 18 |
| 4.2  | Physical Asset model, from [6]                                      | 20 |
| 4.3  | Physical Asset - Equipment relation, from [6]                       | 21 |
| 4.4  | Material Model, from [6]                                            | 23 |
| 4.5  | Process Segment Model, from [6]                                     | 27 |
| 4.6  | Operations Definition Model, from [6]                               | 28 |
| 4.7  | Operations Schedule Model, from [6]                                 | 29 |
| 5.1  | Project Overall View                                                | 31 |
| 5.2  | Role Based Equipment Model, from [6]                                | 33 |
| 5.3  | Equipment Database Table Structure                                  | 34 |
| 5.4  | Process Segment Database Structure                                  | 35 |
| 5.5  | Equipment Form                                                      | 36 |
| 5.6  | Equipment Class Form                                                | 37 |
| 5.7  | Index Page - Example                                                | 37 |
| 5.8  | Equipment Main Page - Example                                       | 38 |
| 5.9  | Operations Schedule Main Page - Example                             | 38 |
| 5.10 | Operations Schedule Page - Example                                  | 39 |
| 5.11 | Workflow Diagram                                                    | 41 |
| 5.12 | Material Planner Logic                                              | 44 |
| 6.1  | Shop-floor Layout                                                   | 49 |
| 6.2  | Equipment Table - Tools Entry                                       | 51 |
| 6.3  | Equipment Table - Machine M1a entry                                 | 52 |
| 6.4  | Equipment Made Up Of Table                                          | 52 |
| 6.5  | Machine M1a Page                                                    | 52 |
| 6.6  | Transformation - Wood to RtopW                                      | 54 |
| 6.7  | Process Segment Table - Transformation Wood - RtopW                 | 54 |
| 6.8  | Equipment Segment Specification Table - Transformation Wood - RtopW | 54 |
| 6.9  | Material Segment Specification Table - Transformation Wood - RtopW  | 54 |
| 6.10 | Process Segment Wood - RtopW Page                                   | 54 |
| 6.11 | XML File Used for Test                                              | 55 |

|      |                                                          |     |
|------|----------------------------------------------------------|-----|
| 6.12 | Operations Schedule Main Page . . . . .                  | 56  |
| 6.13 | Operations Schedule Detail Page . . . . .                | 56  |
| 6.14 | Operations Request Page . . . . .                        | 57  |
| 6.15 | Segment Request Page . . . . .                           | 57  |
| 6.16 | Dispatch Plan . . . . .                                  | 58  |
|      |                                                          |     |
| B.1  | Equipment Database Table Structure . . . . .             | 87  |
| B.2  | Physical Asset Database Table Structure . . . . .        | 88  |
| B.3  | Material Database Table Structure . . . . .              | 89  |
| B.4  | Process Segment Database Table Structure . . . . .       | 90  |
| B.5  | Operations Definition Database Table Structure . . . . . | 91  |
| B.6  | Operations Schedule Database Table Structure . . . . .   | 93  |
|      |                                                          |     |
| C.1  | Navbar Pages Diagram . . . . .                           | 94  |
| C.2  | Equipment Pages Diagram . . . . .                        | 96  |
| C.3  | Physical Asset Pages Diagram . . . . .                   | 97  |
| C.4  | Material Pages Diagram . . . . .                         | 98  |
| C.5  | Process Segment Pages Diagram . . . . .                  | 100 |
| C.6  | Operations Definition Pages Diagram . . . . .            | 102 |
| C.7  | Operations Schedule Pages Diagram . . . . .              | 104 |
| C.8  | Index Page - Equipment . . . . .                         | 105 |
| C.9  | Index Page - Material . . . . .                          | 105 |
| C.10 | Index Page - Process Segment . . . . .                   | 106 |
| C.11 | Equipment Main Page 1 . . . . .                          | 106 |
| C.12 | Equipment Main Page 2 . . . . .                          | 107 |
| C.13 | Equipment Class Form . . . . .                           | 107 |
| C.14 | Equipment Form . . . . .                                 | 108 |
| C.15 | Equipment Detail Page . . . . .                          | 108 |
| C.16 | Equipment Property Form . . . . .                        | 109 |
| C.17 | Process Segment Main Page . . . . .                      | 109 |
| C.18 | Process Segment Form . . . . .                           | 110 |
| C.19 | Process Segment Main Page . . . . .                      | 111 |
| C.20 | Equipment Segment Specification Form . . . . .           | 111 |
| C.21 | Material Segment Specification Form . . . . .            | 112 |
|      |                                                          |     |
| E.1  | Transformation - Wood to RtopW . . . . .                 | 114 |
| E.2  | Transformation - Wood to StopW . . . . .                 | 114 |
| E.3  | Transformation - Wood to LegW . . . . .                  | 115 |
| E.4  | Transformation - Wood to LegW . . . . .                  | 115 |
| E.5  | Transformation - Metal to RtopM . . . . .                | 115 |
| E.6  | Transformation - Metal to StopM . . . . .                | 116 |
| E.7  | Transformation - Metal to LegM . . . . .                 | 116 |
| E.8  | Transformation - Metal to LegM . . . . .                 | 116 |
| E.9  | Transformation - Assemble to RWW . . . . .               | 117 |
| E.10 | Transformation - Assemble to SWW . . . . .               | 117 |
| E.11 | Transformation - Assemble to RWM . . . . .               | 118 |
| E.12 | Transformation - Assemble to SWM . . . . .               | 118 |
| E.13 | Transformation - Assemble to RMM . . . . .               | 119 |
| E.14 | Transformation - Assemble to SMM . . . . .               | 119 |

# List of Tables

|      |                                                                          |    |
|------|--------------------------------------------------------------------------|----|
| 3.1  | Overview of selected commercial and open MES-related solutions . . . . . | 14 |
| 4.1  | Equipment Class Attributes [6] . . . . .                                 | 18 |
| 4.2  | Equipment Class Property Attributes [6] . . . . .                        | 19 |
| 4.3  | Equipment Attributes [6] . . . . .                                       | 19 |
| 4.4  | Equipment Property Attributes [6] . . . . .                              | 19 |
| 4.5  | Physical Asset Attributes [6] . . . . .                                  | 21 |
| 4.6  | Physical Asset Property Attributes [6] . . . . .                         | 22 |
| 4.7  | Physical Asset Class Attributes [6] . . . . .                            | 22 |
| 4.8  | Physical Asset Class Property Attributes [6] . . . . .                   | 22 |
| 4.9  | Material Class Attributes [6] . . . . .                                  | 24 |
| 4.10 | Material Class Property Attributes [6] . . . . .                         | 24 |
| 4.11 | Material Definition Attributes [6] . . . . .                             | 25 |
| 4.12 | Material Definition Property Attributes [6] . . . . .                    | 25 |
| 4.13 | Material Lot Attributes [6] . . . . .                                    | 26 |
| 4.14 | Material Lot Property Attributes [6] . . . . .                           | 26 |
| 6.1  | Equipment - Tools available on each machine . . . . .                    | 51 |
| 6.2  | Work-Piece Transformations . . . . .                                     | 53 |
| A.1  | Process Segment Attributes [6] . . . . .                                 | 68 |
| A.2  | Equipment Segment Specification Segment Attributes [6] . . . . .         | 69 |
| A.3  | Equipment Segment Specification Property Attributes [6] . . . . .        | 69 |
| A.4  | Physical Asset Segment Specification Attributes [6] . . . . .            | 70 |
| A.5  | Physical Asset Segment Specification Property Attributes [6] . . . . .   | 70 |
| A.6  | Material Segment Specification Attributes [6] . . . . .                  | 71 |
| A.7  | Material Segment Specification Property Attributes [6] . . . . .         | 72 |
| A.8  | Process Segment Parameter Attributes [6] . . . . .                       | 72 |
| A.9  | Process Segment Dependency Attributes [6] . . . . .                      | 73 |
| A.10 | Operations Definition Attributes [6] . . . . .                           | 73 |
| A.11 | Operations Segment Attributes [6] . . . . .                              | 74 |
| A.12 | Parameter Specification Attributes [6] . . . . .                         | 74 |
| A.13 | Equipment Specification Attributes [6] . . . . .                         | 75 |
| A.14 | Equipment Specification Property Attributes [6] . . . . .                | 75 |
| A.15 | Physical Asset Specification Attributes [6] . . . . .                    | 76 |
| A.16 | Physical Asset Specification Property Attributes [6] . . . . .           | 76 |
| A.17 | Material Specification Attributes [6] . . . . .                          | 77 |
| A.18 | Material Specification Property Attributes [6] . . . . .                 | 78 |
| A.19 | Operations Segment Dependency Attributes [6] . . . . .                   | 78 |

|                                                                     |    |
|---------------------------------------------------------------------|----|
| A.20 Operations Schedule Attributes [6] . . . . .                   | 79 |
| A.21 Operations Request Attributes [6] . . . . .                    | 80 |
| A.22 Segment Requirement Attributes [6] . . . . .                   | 81 |
| A.23 Segment Parameter Attributes [6] . . . . .                     | 81 |
| A.24 Equipment Requirement Attributes [6] . . . . .                 | 82 |
| A.25 Equipment Requirement Property Attributes [6] . . . . .        | 82 |
| A.26 Physical Asset Requirement Attributes [6] . . . . .            | 83 |
| A.27 Physical Asset Specification Property Attributes [6] . . . . . | 83 |
| A.28 Material Requirement Attributes [6] . . . . .                  | 84 |
| A.29 Material Requirement Property Attributes [6] . . . . .         | 85 |

# List of Acronyms

|       |                                                         |
|-------|---------------------------------------------------------|
| AAU   | Aalborg University                                      |
| AI    | Artificial Intelligence                                 |
| B2MML | Business To Manufacturing Markup Language               |
| CPM   | Collaborative Production Management                     |
| DEEC  | Departamento de Engenharia Eletrotécnica e Computadores |
| ERP   | Enterprise Resource Planning                            |
| FEUP  | Faculdade de Engenharia da Universidade do Porto        |
| GUI   | Graphical User Interface                                |
| IIoT  | Industrial Internet of Things                           |
| IoT   | Internet of Things                                      |
| ISA   | International Society of Automation                     |
| MES   | Manufacturing Execution System                          |
| MO&C  | Manufacturing Operations and Control                    |
| MOM   | Manufacturing Operations Management                     |
| OEE   | Overall Equipment Effectiveness                         |
| RAMI  | Reference Architectural Model Industry                  |
| RDBMS | Relational Database Management System                   |
| SCADA | Supervisory Control And Data Acquisition                |
| UI    | User Interface                                          |
| URS   | User Requirement Specifications                         |
| XML   | Extended Markup Language                                |

# Chapter 1

## Introduction

### 1.1 Context

Industry 4.0 is pushing manufacturing towards highly connected, data-driven environments where information must flow reliably between business planning and the shop-floor. With this, a MES occupies a central position in the automation hierarchy, serving as the bridge between high-level planning systems, such as ERP systems, and low-level operating systems with real-time production operations, providing visibility and control over work-in-progress. The ISA-95 standard formalizes this integration by defining functional levels, models, and terminology for enterprise and control systems integration, placing MES between them with responsibilities such as production scheduling, dispatching, resource coordination, and performance tracking.

Although ISA-95 was originally developed in the mid-1990s, the standard was designed as a technology-agnostic framework that focuses on operations and information exchanges instead of specific implementation technologies. This abstraction allows the ISA-95 models to remain applicable in the current era of Industry 4.0, Internet of Things (IoT), and smart manufacturing, where data flows are increasingly distributed, and systems are more interconnected than just hierarchical. The concept of levels still provides a useful way to define clear boundaries and responsibilities between systems, which is important for designing and executing integration projects. The standard is widely used as a reference architecture for modern manufacturing information exchange, and many manufacturers rely on it to specify, develop, and integrate complex systems and processes in the Industry 4.0 and beyond.[1]

### 1.2 Motivation

Many implemented MES solutions are bound to specific vendors, technology stacks, or industrial sectors, making it hard to adapt, extend, or reuse in different manufacturing environments. Free and open-source options, while attractive, typically cover only parts of the required functionality, such as isolated planning, execution, or performance monitoring, and rarely provide a complete, standard-based, generic MES that can be configured through data and models.

This gap is particularly evident in the existing solutions and research available. While it is currently possible to model a production line using static ISA-95 objects (defining equipment, recipes, and materials), there is no lightweight, adaptable application capable of scheduling and controlling shop-floors simulators, such as the DEEC. This creates a disconnect between the theoretical study of standards and the practical application of control algorithms. Therefore, the motivation for this work is to bridge this gap by developing a system that is not only ISA-95 compliant but also capable of being generic for different situations, effectively turning the standard from a passive data model into an active data model.

### 1.3 Problem Statement

While the ISA-95 standard is widely recognized for integrating enterprise and shop-floor systems, there is still a lack of practical implementations that demonstrate how it can be applied to design flexible and generic MES solutions. Many existing MES platforms are built for specific industrial areas or technology stacks, making them difficult to adapt to different manufacturing environments through configuration alone. The central problem addressed in this work is how to design and implement a generic, scalable MES that complies with ISA-95 and can be validated in a simulated shop-floor environment. More specifically, the challenge is to create a MES program that is not tied to a single industrial domain, but can support different production scenarios via appropriate configuration and data models, while following the framework and information flows developed by the standard.

This study aims to address this problem by developing a generic MES application based on the principles and structure defined by ISA-95, focusing on core functionalities such as scheduling, monitoring, and production control and validating them through integration with a simulated shop-floor environment. By following ISA-95 guidelines, the resulting solution is intended to be modular, scalable, and operational in different shop-floor scenarios, serving as a reusable foundation for future MES developments and research in Industry 4.0 contexts.

### 1.4 Objectives

The general objective of the dissertation is to design and implement a modular, ISA-95 compatible MES application that can be configured for different shop-floor scenarios and validated using a shop-floor simulator. This general goal can be decomposed into specific objectives:

- Study the ISA-95 standard and existing MES solutions, and identify requirements and modeling guidelines for a generic, standards-based MES.
- Develop a database aligned with ISA-95 information models to represent equipment, materials, processes, and activities relevant to the simulated production environment.
- Design and implement core MES modules for order management, production scheduling, and dispatching, and real-time status tracking.

- Implement a generic production scheduling algorithm that uses standardized information stored in the MES database, enabling different configurations to be handled through data and parameters.
- Create a Graphical User Interface (GUI) for system configuration, order entry, and real-time process monitoring.
- Integrate the MES prototype with a shop-floor simulator and evaluate its accuracy, flexibility, and limitations under different simulated operating conditions.

## 1.5 Dissertation Structure

The rest of this document is organized as follows:

Chapter 2, Background, provides the essential theoretical basis for understanding the project. It summarizes the ISA-95 standard, key MES functions, and the main concepts and terminology used throughout the report, helping that the reader follow the subsequent state-of-the-art discussion and design choices.

Chapter 3, State-of-the-art, reviews existing MES solutions and related work. It examines commercial MES platforms and free/open-source options, as well as standards and research on ISA-95 based modeling, and production scheduling, highlighting their main features, limitations, and the gaps that motivate the proposed work.

Chapter 4, ISA-95 Data Models, analyzes the data structures defined by the ISA-95 standard that are relevant to the scope of this project. It examines the main object models and information exchanges related to production operations, resources, product definitions, production scheduling, and production performance, identifying the elements required for the development of the proposed MES application.

Chapter 5, System Design and Implementation, presents the architecture and implementation of the proposed MES application. It describes the adopted technologies, database design, software modules, scheduling functionality, and integration with the DEEC factory simulator.

Chapter 6, Validation and Results, evaluates the developed system through representative use cases and test scenarios. It demonstrates the implemented functionalities and discusses the obtained results regarding ISA-95 compliance, production scheduling, and system integration.

Chapter 7, Conclusion and Future Work, summarizes the main contributions of the dissertation, discusses the achieved objectives and limitations, and proposes directions for future developments of the MES platform.

## Chapter 2

# Background

### 2.1 Industry 4.0

Industry 4.0, often referred as smart manufacturing, represents the digital transformation of industrial production through the integration of IoT, cloud computing, data analytics, and artificial intelligence (AI) into factories and supply chains. Smart factories use networks of sensors, embedded software, and robotic systems to collect and analyze data in real-time, combining it with information from ERP, supply chain, and customer systems to achieve new levels of visibility and insight. These technologies enable increased automation, predictive maintenance, self-optimization, and greater flexibility, allowing manufacturers to respond faster to changing customer demands while improving quality and productivity.[2]

As manufacturing systems become more connected and data-driven, they require information platforms capable of coordinating business planning, production execution, and analysis across different devices and software systems. This reinforces the need for structured approaches to system integration, where clearly defined roles, responsibilities, and information flows help ensure that data moves consistently between the business and shop-floor domains.

#### 2.1.1 Reference Architectural Model Industry 4.0 (RAMI 4.0)

A widely used reference for structuring Industry 4.0 systems is the Reference Architectural Model Industry 4.0 (RAMI 4.0). RAMI 4.0 is a three-dimensional model that describes Industry 4.0 solutions along three axes: hierarchy levels ranging from field device up to connected world, the product life cycle from development to disposal, and architecture layers from physical assets up to business processes. This model helps break down complex smart factory scenarios into manageable views and provides a common language for different stakeholders involved in system design and integration.

RAMI 4.0 complements standards such as ISA-95 by emphasizing that integration must be considered in both technical layers and organizational levels. In this work, RAMI 4.0 is used

mainly as a conceptual reference to situate MES at the operations level and to highlight the importance of clearly defined system boundaries when designing an ISA-95 compatible MES for Industry 4.0.[3]

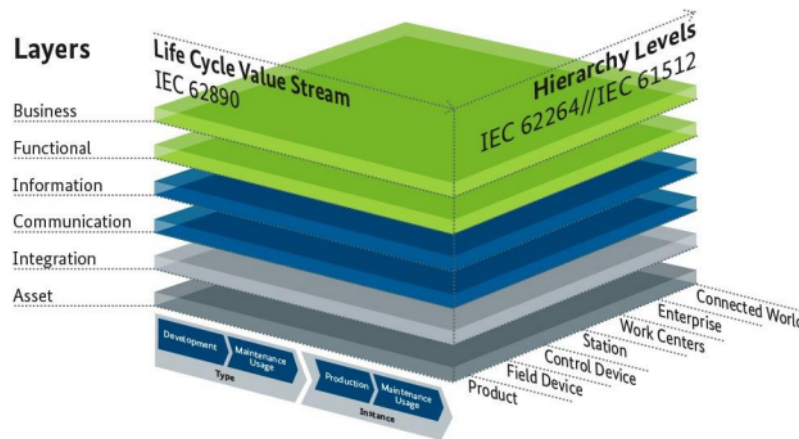


Figure 2.1: RAMI 4.0, from [3]

## 2.2 ISA-95 Standard Overview

The ISA-95 standard, also known as IEC 62264, developed by the International Society of Automation, defines a framework for integrating enterprise and control systems in manufacturing environments. Its main goal is to make it easy for business systems, such as ERP, and shop-floor systems, including MES and control systems, by providing standardized models and terminology[1]

ISA-95 introduces a functional hierarchy with levels ranging from the physical process to business planning and logistics, as shown in Figure 2.2.

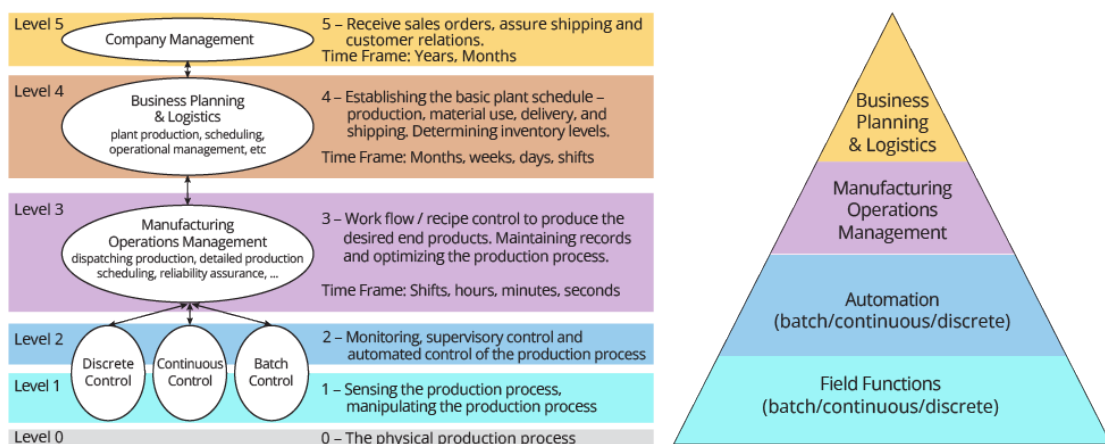


Figure 2.2: ISA-95 model, from [1]

The ISA-95 model consists of five levels:

- Level 0: The physical production process - machinery and other assets in the field.
- Level 1: Sensing and manipulation the product process - sensors, smart devices, valves and other devices.
- Level 2: Monitoring and supervising control - programmable logic controllers (PLCs), distributed control systems and other control devices or systems such as supervisory control and data acquisition (SCADA).
- Level 3: Manufacturing Operations Management - describes manufacturing execution systems (MES).
- Level 4: Business Logistics - includes enterprise resource planning (ERP) systems which establish the basic plant schedule - production, material use, procurement, delivery and shipping.

The ISA-95 hierarchy differentiates between different levels of activities in a manufacturing system. At the bottom, Level 0 corresponds to the physical manufacturing or production process itself. Level 1 covers sensing and actuation, including manual sensing, sensors, and actuators used to monitor and manipulate the process. Level 2 comprises the control activities, whether manual or automated, that keep the process stable and under control. Together, Levels 1 and 2 represent process control and cell or line supervision functions, which are not the main focus of this standard.[4]

ISA-95 uses the term Manufacturing Operations Management (MOM) to describe the activities and information flows at Level 3, while Manufacturing Operations and Control (MO&C) involves the activities and information flows at Levels 1, 2, and 3. All activities outside the MO&C domain are considered to belong to the enterprise domain, corresponding to Levels 4 and above.[4]

In addition to the hierarchy, ISA-95 provides a functional model for manufacturing operations, describing how Level 3 interacts with Level 4 and with Levels 1–2 through standardized information flows. In this model, Level 4 provides production plans and schedules down to Level 3, which refines them into executable instructions and dispatches them to the shop floor. Execution results and performance information are then reported back up the hierarchy. This functional view is central to MES design, as it identifies the key types of information and messages that a Level-3 system must handle.

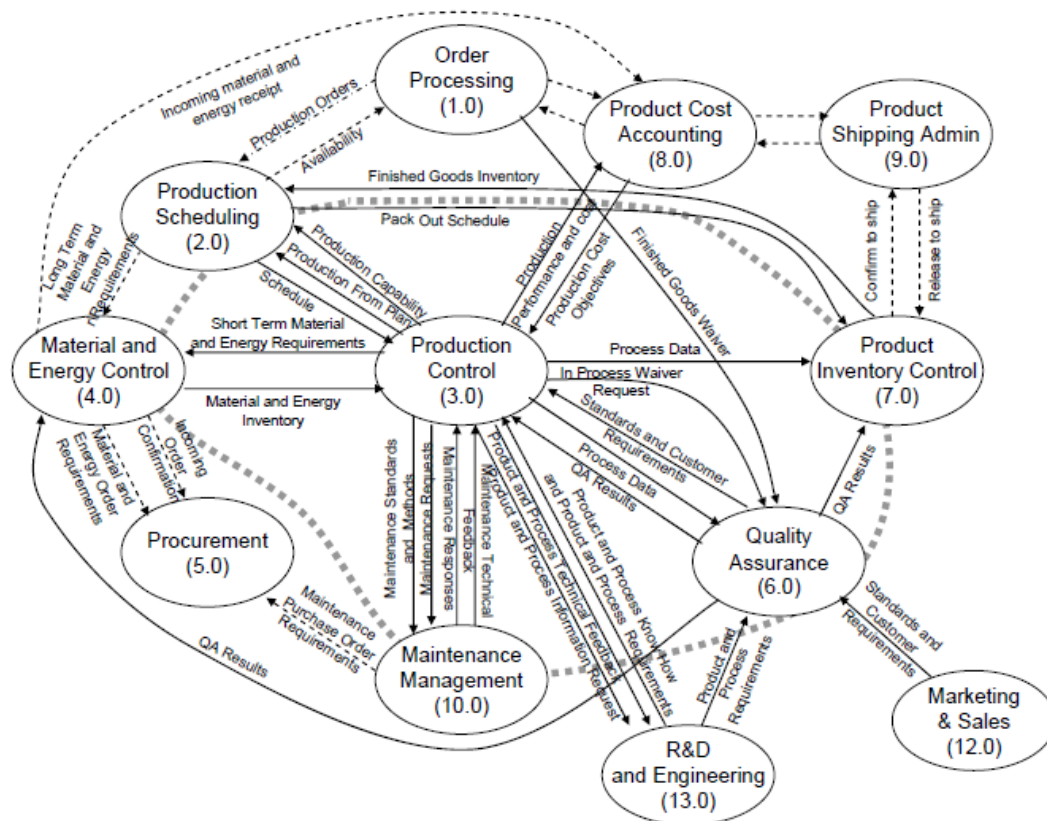


Figure 2.3: Data Flow - Functional Model, from [4]

The wide dotted boundary in Figure 2.3 represents the interface between Level 3 (manufacturing operations and control, including MES) and Level 4 (enterprise planning and business systems, such as ERP). The functions on the manufacturing side include production scheduling, production control, quality, and material management, while the functions on the enterprise side focus on order processing, cost accounting, and shipping administration. The labeled arrows illustrate the main information exchanges required to coordinate these functions, emphasizing how standardized data flows enable integration across the enterprise and shop-floor domains.

In addition to the functional model, ISA-95 also defines an activity-relationship view that shows how Level 3 activities interact with activities at Levels 4 and 2. This view emphasizes the MES role as an intermediate layer that receives planning information from enterprise systems, coordinates manufacturing operations, and exchanges data with process-control activities on the shop-floor.

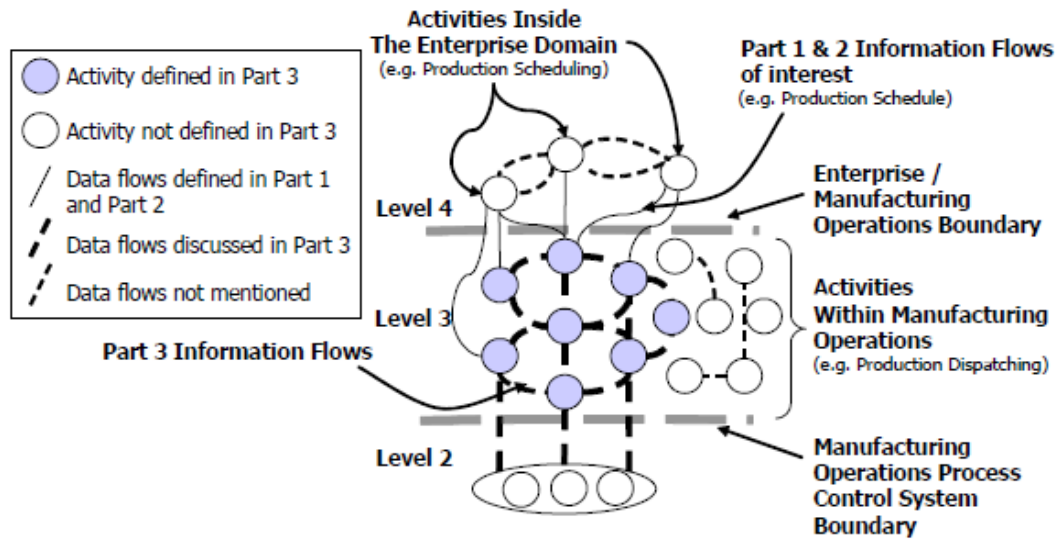


Figure 2.4: Activity Relationship, from [5]

A common way to visualize ISA-95 activities is to place the Level 3 activity model in relation to the activities defined at Levels 4 and 2. In this representation, the gray circles highlight the Level 3 activities that are detailed in Part 3 of the standard, while the heavy dashed lines show the main information flows between these activities. The diagram also indicates how Level 3 activities exchange information with dependent Level 2 activities, making explicit the role of Level 3 as an intermediate layer between enterprise planning and process control.[5]

## 2.3 ISA-95 Object and Activity Models

Beyond the hierarchical and functional views, ISA-95 defines object (information) models and activity models that describe how manufacturing information is structured and used in operations. The information models standardize how key entities, such as materials, equipment, personnel, and physical assets, are represented, introducing attributes, relationships, and identifiers that enable consistent representation across different systems.

For this work, the most relevant models are those that describe equipment/physical assets and materials. The role-based equipment model defines concepts such as Equipment Class, Equipment, Equipment Property, and Equipment Capability Test, and the relationships between them. This model supports the description of what equipment can do (capabilities) and how it is organized, which is essential for scheduling and resource allocation in a MES.

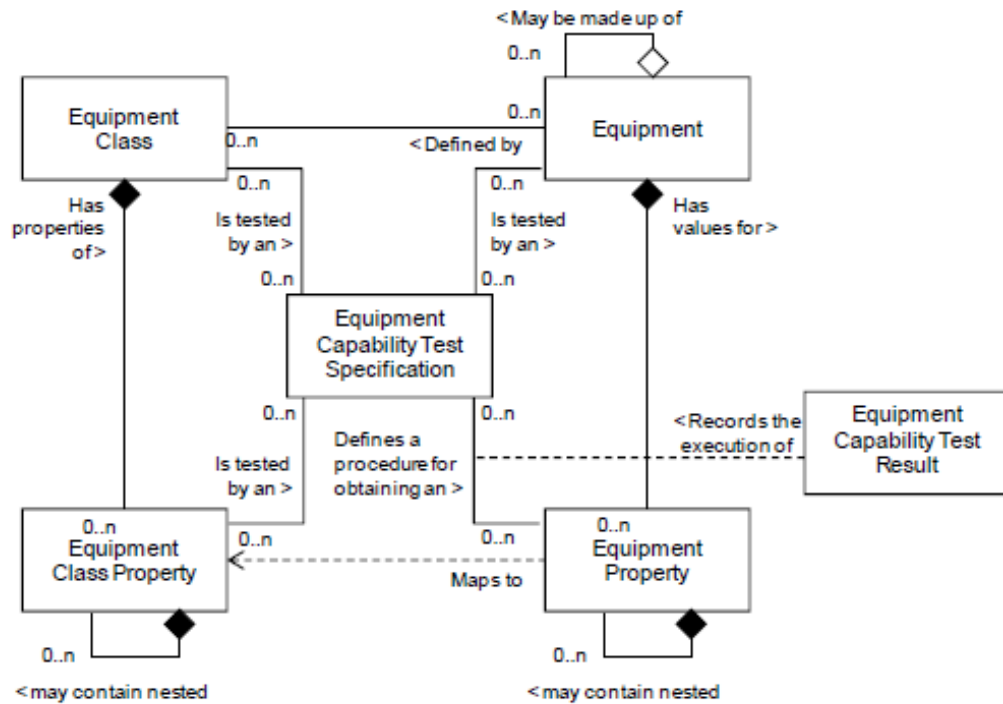


Figure 2.5: Role based equipment model, from [6]

Similarly, the material model defines Material Class, Material Definition, Material Lot, and Material Sublot, together with associated properties and test specifications. This model allows the MES to represent different product types, lots, and quality information in a structured way, which is important for tracking, and for ensuring that scheduling decisions respect material availability and constraints.

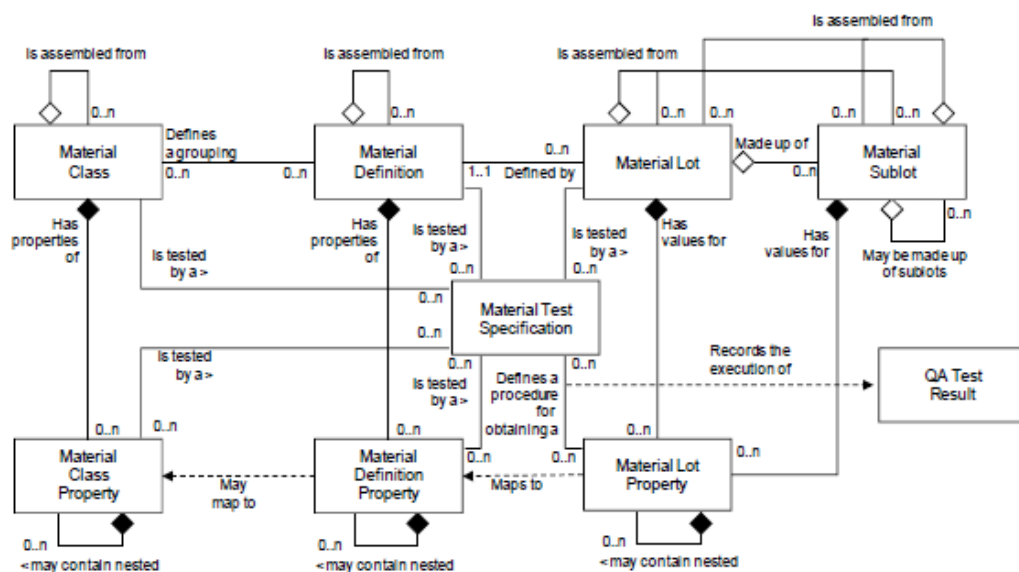


Figure 2.6: Material model, from [6]

On the activity side, ISA-95 describes manufacturing operations in four main domains: production, quality, maintenance, and inventory operations management. For each domain, it identifies activities such as production scheduling, dispatching, execution management, and performance analysis, and explains how they exchange information with each other and with enterprise-level functions. This structured view of activities helps define what a MES at Level 3 is expected to do and which information it must consume and produce.

In this project, these ISA-95 object and activity models will serve as a reference for defining the MES database schema and the interactions between modules. By aligning data structures and information flows with ISA-95, the MES can represent different manufacturing environments mainly through configuration and standardized entities, rather than custom as specific models

## 2.4 Manufacturing Execution Systems

Manufacturing Execution Systems (MES) are software systems that manage and monitor work-in-process on the shop-floor, linking business level planning with real-time shop-floor operations. Positioned between ERP and the control/supervision layers, MES receives production plans, transforms them into detailed schedules and dispatch lists, and coordinates the execution of operations using information from machines, operators, and other devices.

Typical MES functions include production scheduling and dispatching, coordination of materials and resources, real-time data collection from equipment, performance tracking, and product genealogy and traceability. By maintaining a consistent view of orders, states, and performance indicators, MES enables better decision making on the shop floor and supports continuous improvement of manufacturing processes.[7]

In Industry 4.0 environments, MES often serves as the operational hub connecting IIoT devices, analytics platforms, and enterprise applications, ensuring that the large volumes of data generated in smart factories are used to adjust and improve production in real time. ISA-95 explicitly places MES at Level 3 of the functional hierarchy and defines the activities, information flows, and models that characterize its interaction with higher-level business systems and lower-level control systems, making it a natural reference for MES design in this work.

## 2.5 Production Scheduling Fundamentals

Production scheduling is the process of deciding when and on which resources production orders and operations should be executed, given constraints such as machine availability, processing times, material availability, and due dates. In a typical manufacturing system, scheduling translates high-level production plans into detailed sequences of operations for machines, work centers, or lines, with the aim of meeting delivery deadlines while using resources efficiently.

Common scheduling objectives include minimizing makespan, total completion time, reducing tardiness, balancing workloads across machines, and minimizing changeovers or setup times. Depending on the production environment, problems are often modeled as flow-shop, job-shop

or flexible job-shop problems, each with different assumptions about how jobs move between resources. Because these problems are computationally complex and often NP-hard, practical solutions in industry rely on heuristic or rule-based methods, for example, earliest due date or shortest processing time rules, sometimes complemented by metaheuristics or mathematical optimization for more complex cases.

Within an ISA-95 compatible MES, the scheduling is a Level 3 activity within the MOM domain. The scheduler operates using standardized information about orders, process segments, equipment capabilities, and material availability, as defined by the ISA-95 information and activity models. Based on this information, it generates feasible production schedules that can be dispatched to shop-floor control systems.

This report focuses on these fundamentals to support the design of a generic MES. By relying on the standardized data models and configurable parameters, the scheduler can be adapted to different shop-floor configurations without being tied to a specific industrial environment or production layout, supporting the overall objective of a scalable and generic ISA-95 compatible MES architecture.[8]

## Chapter 3

# State-of-the-art

### 3.1 Commercial MES Solutions

Commercial MES platforms are typically delivered as modular suites that cover a broad set of functions, including production execution, quality management, traceability, performance analysis, and integration with ERP and automation systems. Vendors position MES as the central system for managing and monitoring real-time production, providing visibility over work-in-process, enforcing process rules, and generating key performance indicators for operations.[7]

Sepasoft provides MES functionality as modules on top of the Ignition SCADA/IIoT platform, offering overall equipment effectiveness (OEE), Track & Trace, recipe management, and scheduling within a common runtime and connectivity layer.[9], [10], [11], [12] IBM and GE Vernova describe MES as tightly integrated with IIoT infrastructures, cloud analytics, and enterprise applications, emphasizing capabilities such as multi-site coordination, genealogy, and advanced reporting. These solutions are powerful and mature, but they are proprietary, closely tied to specific technology stacks, and generally aimed at large-scale industrial deployments, which makes them less suitable as lightweight, generic platforms.[7], [13]

Critical Manufacturing is a Portuguese software company founded in 2009 and headquartered in Maia, Porto, Portugal. Since 2018, it has been part of ASMPT, a global provider of semiconductor and electronics manufacturing solutions. The company specializes in Manufacturing Execution Systems (MES) for complex discrete manufacturing industries, including semiconductor, electronics, medical devices, and industrial equipment manufacturing. Its solution has gained international recognition due to its scalability, modular architecture, and strong support for Industry 4.0 digital transformation initiatives [14].

The Critical Manufacturing MES is designed as a comprehensive platform that integrates manufacturing execution, production monitoring, quality management, traceability, material logistics, maintenance, analytics, and enterprise connectivity within a single solution. Rather than providing a monolithic application, the platform is composed of independent but interoperable modules that can be deployed according to the specific requirements of each manufacturing environment. This

modular approach allows organizations to gradually expand system functionality while minimizing unnecessary complexity and implementation effort [15].

A key characteristic of the platform is its alignment with Industry 4.0 principles. The system supports real-time data collection from production equipment, seamless integration with enterprise applications such as ERP systems, advanced analytics, and cloud-ready deployment. The platform also incorporates an integrated data infrastructure capable of supporting artificial intelligence applications and modern manufacturing intelligence solutions. These capabilities enable manufacturers to increase production visibility, improve decision-making, and respond rapidly to operational changes [16].

From the perspective of manufacturing standards, Critical Manufacturing explicitly adopts the ISA-95 reference model as the foundation for organizing manufacturing information and operations. The system implements the Level 3 manufacturing operations layer, providing standardized representations of manufacturing resources, materials, production activities, and equipment. Internal data models and functional modules are mapped to ISA-95 concepts, facilitating interoperability with enterprise systems while promoting standardized communication between business and shop-floor applications. This standards-based approach reduces integration complexity and improves system maintainability across heterogeneous manufacturing environments [17].

Migration from legacy MES platforms is also considered within the solution architecture. Critical Manufacturing promotes a phased migration strategy that allows manufacturers to modernize existing systems incrementally while maintaining production continuity. This approach reduces implementation risks and enables organizations to progressively adopt new functionalities without requiring a complete replacement of operational systems [18].

Overall, Critical Manufacturing represents one of the most advanced commercial implementations of an ISA-95-compliant MES platform. Its emphasis on modularity, interoperability, standards-based information models, and Industry 4.0 technologies makes it an important reference for the development of generic MES architectures.

## 3.2 Free and Open-Source MES Tools

Free and open-source solutions promise lower entry costs and greater transparency, but they tend to address only parts of the MES problem. The MDCplus survey lists several representative projects, including IMES[19], qcadoo MES, Libre[20], Apache OFBiz[21], [22], and frePPLe[23]. IMES and qcadoo MES focus on job routing, work-order tracking, and basic production management for smaller manufacturers. Libre targets IIoT-based monitoring and performance analysis, relying on technologies such as Grafana, InfluxDB, and PostgreSQL. Apache OFBiz embeds manufacturing functionality inside broader ERP frameworks, while frePPLe concentrates on advanced planning and scheduling rather than execution.[24]

Overall, these tools are valuable building blocks but do not typically provide a complete, ISA-95 aligned MES that can be configured mainly through data and models. Many are ERP

extensions or monitoring components, and a significant integration effort is required to combine planning, execution, and monitoring into a coherent system that reflects standard MES roles.

A summary of selected commercial and open solutions is presented in Table 3.1.

Table 3.1: Overview of selected commercial and open MES-related solutions

| Solution                | Type                       | License / Access               | Main scope                                              | ISA-95 focus                              | Main limitation                                   |
|-------------------------|----------------------------|--------------------------------|---------------------------------------------------------|-------------------------------------------|---------------------------------------------------|
| Sepasoft MES (Ignition) | Commercial MES modules     | Proprietary (Ignition modules) | Execution, OEE, traceability, some scheduling [10]      | Functional alignment                      | Vendor-locked, requires Ignition stack            |
| GE Vernova MES          | Commercial MES suite       | Proprietary                    | Full MES: execution, quality, genealogy, analytics [25] | Uses ISA-95 as reference architecture [4] | Complex, oriented to large deployments            |
| IMES                    | OSS MES                    | Apache-2.0                     | Basic job routing and tracking [24]                     | Not explicitly ISA-95                     | Limited features and integrations                 |
| qcadoo MES              | OSS MES                    | AGPL (Community Edition)       | Web-based production tracking and scheduling [24]       | Not explicitly ISA-95                     | Partial MES scope; CE may lag commercial version  |
| Libre                   | OSS monitoring tool        | Open-source                    | IIoT data capture and performance monitoring [24]       | Not ISA-95 focused                        | Little execution/scheduling; needs external stack |
| Apache OFBiz            | ERP with MES/MOM modules   | Apache-2.0                     | ERP suite with manufacturing and maintenance [24]       | Not explicitly ISA-95                     | Heavyweight; MES embedded in ERP                  |
| Odoo Community Edition  | ERP with manufacturing app | LGPLv3                         | ERP + manufacturing and inventory apps [24]             | Not explicitly ISA-95                     | Limited MES depth; advanced features are paid     |
| frePPLe                 | Planning / APS tool        | Open-source                    | Demand planning and production scheduling [24]          | Not ISA-95 focused                        | No execution layer; must be combined with MES     |

### 3.3 Academic MES and MOM Architectures

Several academic works address how MES and MOM architectures should evolve for IIoT-enabled, highly dynamic production environments. Mantravadi and Møller examine design choices for next-generation IIoT-connected MES/MOM systems, identifying key dimensions such as deployment models, on-premise vs. cloud, integration patterns, service-oriented, event-driven, and data management strategies. They conclude that modular, loosely coupled architectures that separate MES functions into services and rely on standardized interfaces are better suited to smart factories, where frequent changes and high connectivity are expected.[26]

Li et al. propose the Aalborg University (AAU) Open Source MES Architecture, an open-source MES designed for smart factories that explicitly exploits IIoT connectivity. Their architecture separates data acquisition, execution logic, and visualization into loosely coupled components and

demonstrates integration with a reconfigurable production line in a university lab. These works demonstrate that open, flexible, and scalable ISA-95 based MES designs are being researched and tested in academic settings, using model-driven and open-source platforms to address small and medium-sized enterprises (SME) needs, which highlights the potential for developing a generic, ISA-95 compatible MES core.[27]

Another approach is an agent-based MES architecture for short-series production, presented by Cupek et al., in which the functional models and information exchange are explicitly derived from the ANSI/ISA-95 (IEC/ISO 62264) standard and implemented in a fully decentralized architecture (heterarchical), cyber-physical environment. Holons connected to the real shop-floor in the Prototyping Department of Continental Ingolstadt interact through web services, and simulation-based planning is used to generate and adapt production schedules in the presence of frequent changes and disturbances. This shows that open, flexible, and adaptable ISA-95 based MES solutions are being researched and tested in realistic industrial settings, highlighting the potential for developing generic ISA-95 compatible MES cores that integrate execution control and advanced scheduling for highly dynamic, short-series production.[28]

### 3.4 ISA-95 Based MES

Several contributions focus specifically on using ISA-95 as the basis for MES requirements and data structures. The ISA-95 standard itself defines object models for materials, equipment, personnel, and processes, as well as activity models for production, quality, maintenance, and inventory operations, providing a rich foundation for structuring MES data and functions. Yue et al. propose guidelines for defining MES user requirement specifications (URS) based on ISA-95, illustrating how to derive functional requirements and system boundaries from the standard's activity and information models. This supports a systematic, standard driven approach to specifying MES functionality.[29]

Standardized ISA-95 and Business To Manufacturing Markup Language (B2MML) models are used to structure the data required for production scheduling, covering elements such as process segments, equipment information, materials, personnel, and operations capabilities. By mapping these elements into B2MML extended markup language (XML) schemas, the approach provides a generic, extensible platform for exchanging scheduling data and results between a scheduling component and production systems like MES or collaborative production management (CPM) suites, independently of the specific scheduling algorithms used. These ISA-95 based approaches show that the standard can effectively guide both requirements and information structures for scheduling and data exchange, although the focus remains on particular functions, such as scheduling integration, rather than on complete, generic MES implementations.[8]

### 3.5 Critical Synthesis and Research Gap

Existing MES solutions and research efforts already address many important aspects of manufacturing execution and operations management. However, these elements rarely converge into a simple, generic, and standards-based implementation. Commercial MES platforms are feature-rich and well integrated, but they are proprietary, complex, and typically unsuitable as lightweight, reusable platforms for experimentation, academic development, or rapid prototyping. Conversely, free and open-source tools provide valuable capabilities for production tracking, monitoring, or planning, yet they usually address only isolated parts of the MES scope and are not consistently structured around ISA-95 information and activity models.

Standard oriented research demonstrates that ISA-95 and B2MML can successfully define information structures, interfaces, and requirement specifications, supporting consistent data exchange and system interoperability. At the same time, academic architectures based on IIoT concepts, service oriented approaches, and agent or holonic systems show how flexible and adaptive MES behavior can be achieved in practice, sometimes explicitly grounded in ISA-95 principles. Nevertheless, these solutions are often made for specific production environments, experimental platforms, or isolated functions, rather than providing a reusable, generic MES core that can be configured across different shop-floor scenarios.

As a result, a clear gap remains: an open, lightweight, and scalable MES core that is explicitly modeled on ISA-95 and designed to be configured primarily through standardized data models is still missing. Addressing this gap motivates the work proposed in this dissertation, which aims to develop and validate a generic ISA-95 compatible MES architecture capable of scheduling, execution coordination, and real-time monitoring in a realistic yet controlled shop-floor simulation environment.

## Chapter 4

# ISA-95 Data Models

### 4.1 Introduction

This chapter delineates the formal modeling framework and the high-level software architecture of the developed Manufacturing Execution System (MES). To bridge the gap between static theoretical standards and active operational control, the proposed system adopts a data-driven paradigm. Rather than utilizing hard-coded control logic tailored to a single production environment, the core application dynamically adapts its behavior based on standardized data models instantiated within a relational database. This chapter details how the physical and logical assets of a shop-floor are abstracted using the ISA-95 framework, maps these abstractions onto a concrete database schema, and presents the modular architecture designed to manage scheduling, control, and real-time monitoring [6].

### 4.2 ISA-95 Information Models Adaptation

To ensure high interoperability and scalability, the foundational data structures of the MES are strictly derived from ISA-95 standard [6]. The objective is to represent all heterogeneous entities of a manufacturing facility, ranging from physical machinery to raw materials, using a uniform, object-oriented language that can be parsed by generic execution algorithms.

### 4.3 Equipment Model

The foundational layer of the system's architecture is the Equipment Model, which is directly derived from the Role Based Equipment Model defined in the ISA-95 standard. This model is essential for defining the hierarchical structure of the manufacturing environment, abstracting physical shop-floor assets into logical entities that can be interpreted by the MES core, as seen in the figure 4.1.

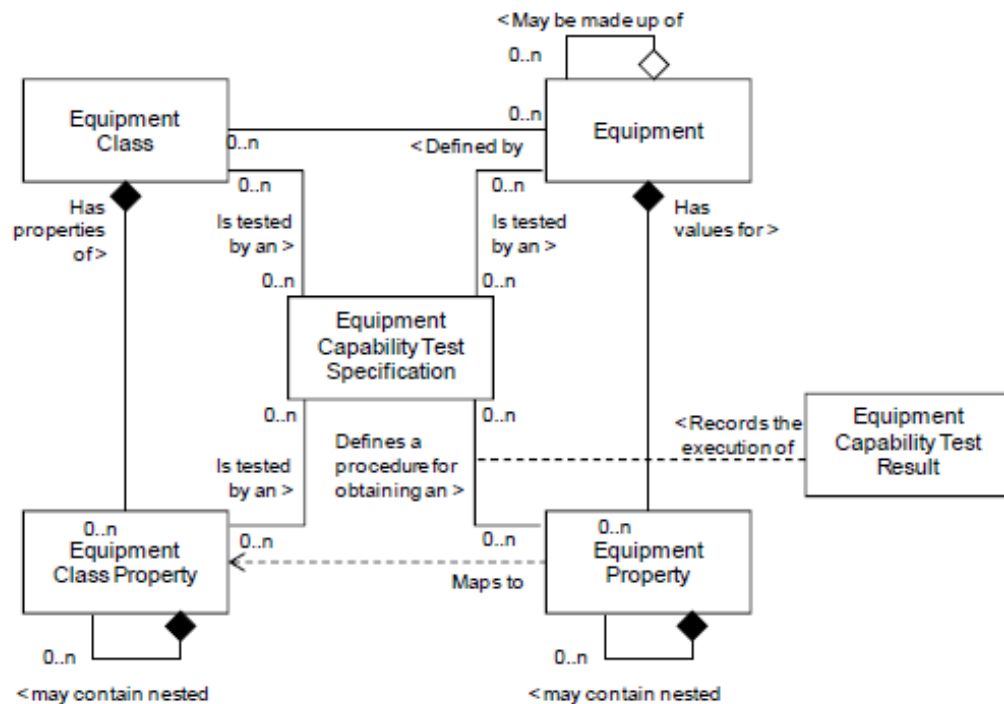


Figure 4.1: Role based equipment model, from [6]

The model is composed of six classes: Equipment , Equipment Class, Equipment Property, Equipment Class Property, Equipment Capability Test Specification and Equipment Capability Test Result.

For the developing of this project it was implemented for the database only the Equipment, Equipment Class, Equipment Property and Equipment Class Property.

#### 4.3.1 Equipment Class

Equipment Class defines a type or a category of equipment. It groups similar equipment into classes to help management and standardize attribute os equipment of the same category, class or type.

Table 4.1: Equipment Class Attributes [6]

| Attribute       | Description                                                                                                                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID              | A unique identification of a specific equipment class within the scope of the information exchanged (production capability, production schedule, production performance, etc.). The ID is used throughout the model whenever the equipment class must be identified. |
| Description     | Additional information about the equipment class.                                                                                                                                                                                                                    |
| Equipment Level | An identification of the level in the role-based equipment hierarchy.                                                                                                                                                                                                |

### 4.3.2 Equipment Class Property

Equipment Class Property is a characteristic or an attribute of an equipment class that describes the classes capacity, limitations, functions, or other important information related to the class.

Table 4.2: Equipment Class Property Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the equipment class property.           |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

### 4.3.3 Equipment

Equipment represents the elements that compose the equipment hierarchy defined in ISA-95 Part 1 [4]. These elements can correspond to different hierarchical levels, such as sites, areas, production units, production lines, work cells, process cells, units, storage zones, or storage units [6].

Table 4.3: Equipment Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the equipment.                          |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

### 4.3.4 Equipment Property

Equipment Property is a characteristic or an attribute of an equipment that describes the equipment's capacity, limitations, functions, or other important information related to the equipment.

Table 4.4: Equipment Property Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the equipment property.                 |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

## 4.4 Physical Asset Model

According to the ISA-95 standard, there is a fundamental distinction between the operational role that a resource plays on the shop floor and the actual physical object fulfilling that role. While the Equipment Model defines the logical role (e.g., a "Temperature Controller" node at a specific workstation), the Physical Asset Model represents the tangible piece of machinery, uniquely identified by a serial number or an asset tag.

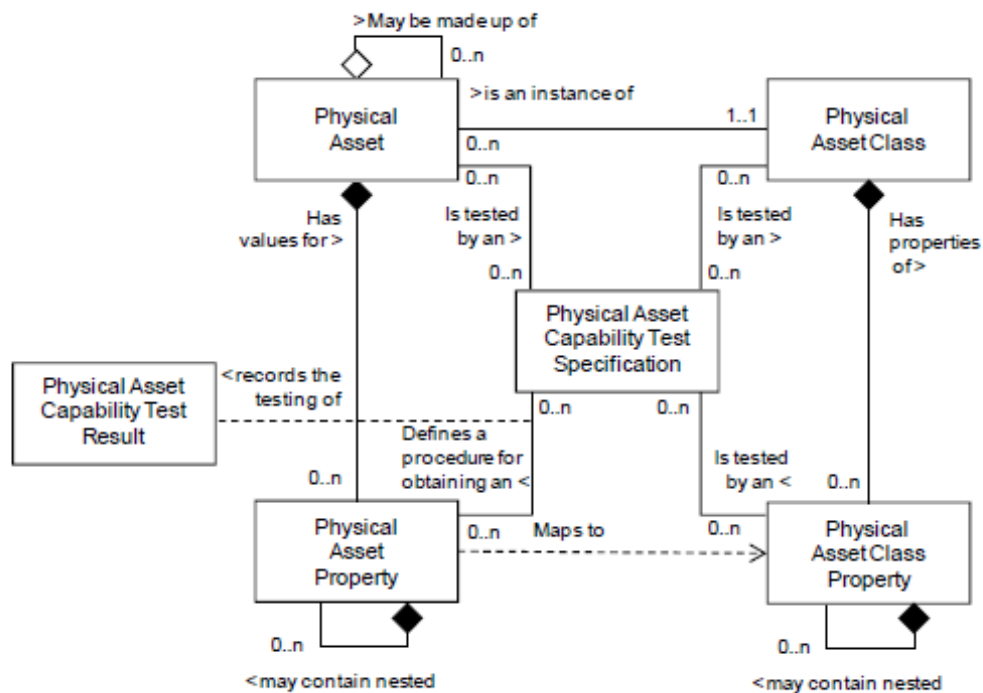


Figure 4.2: Physical Asset model, from [6]

As defined by the standard, if a physical machine breaks down and is replaced, the operational role (the Equipment ID) remains unchanged, but a new physical asset (with a new Physical Asset ID) assumes that role. In a traditional enterprise environment, managing this separation is critical for Level 4 tasks, such as tracking asset depreciation, managing life cycle maintenance, and logging physical capability tests.

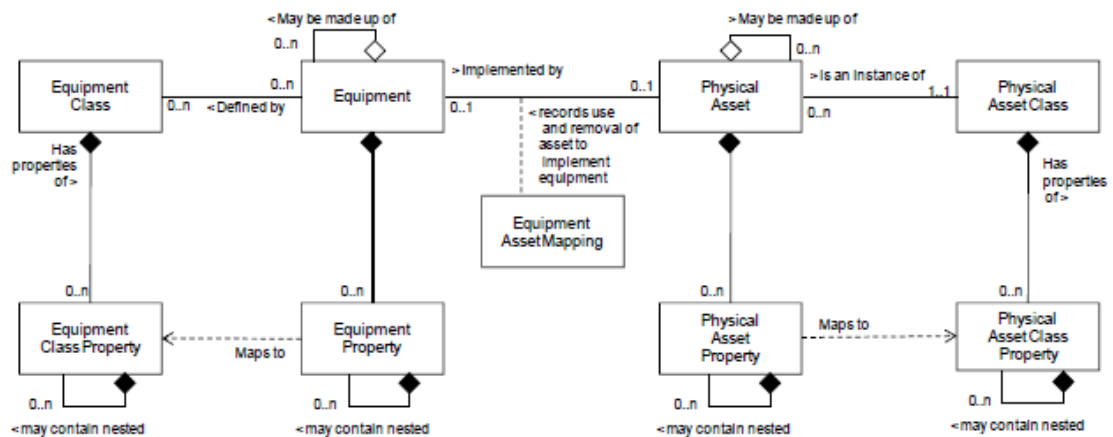


Figure 4.3: Physical Asset - Equipment relation, from [6]

The model is composed of six classes: Physical Asset , Physical Asset Class, Physical Asset Property, Physical Asset Class Property, Physical Asset Capability Test Specification and Physical Asset Capability Test Result.

For the developing of this project it was implemented for the database only the Physical Asset, Physical Asset Class, Physical Asset Property and Physical Asset Class Property.

#### 4.4.1 Physical Asset

A physical asset represents a physical piece or part of an equipment. Physical assets may be made up of other physical assets.

Table 4.5: Physical Asset Attributes [6]

| Attribute      | Description                                                                                 |
|----------------|---------------------------------------------------------------------------------------------|
| ID             | Defines a unique identification of a physical asset.                                        |
| Description    | Contains additional information and descriptions of the physical asset.                     |
| Location       | Actual physical location of the physical asset.                                             |
| Fixed Asset ID | Contains a unique identification for financial tracking as required by laws or regulations. |
| Vendor ID      | Contains a vendor 's serial number.                                                         |

#### 4.4.2 Physical Asset Property

Properties of physical assets will be listed as properties of physical assets. A physical asset can have zero or more physical asset properties. These specify the current property values of the physical asset for the correspondent physical asset class property.

Physical asset properties may contain nested physical asset properties.

Table 4.6: Physical Asset Property Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the physical asset property.            |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

#### 4.4.3 Physical Asset Class

A grouping of physical assets with similar characteristics for purposes of repair and replacement shall be used as a physical asset class. Any physical asset shall be a member of one physical asset class.

Table 4.7: Physical Asset Class Attributes [6]

| Attribute    | Description                                                             |
|--------------|-------------------------------------------------------------------------|
| Manufacturer | An identification of the manufacturer.                                  |
| ID           | Defines a unique identification of a physical asset class.              |
| Description  | Contains additional information and descriptions of the physical asset. |

#### 4.4.4 Physical Asset Class Property

Properties of a physical asset class will be listed as physical asset class properties.

Table 4.8: Physical Asset Class Property Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the physical asset class property.      |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

### 4.5 Material Model

The material model shown in Figure 4.4 defines the actual materials, material definitions, and information about classes of material definitions. Material information includes the inventory of raw, finished, intermediate materials, and consumables. The information about planned or actual material is contained in the material lot and material subplot information. Material classes are defined to organize materials.

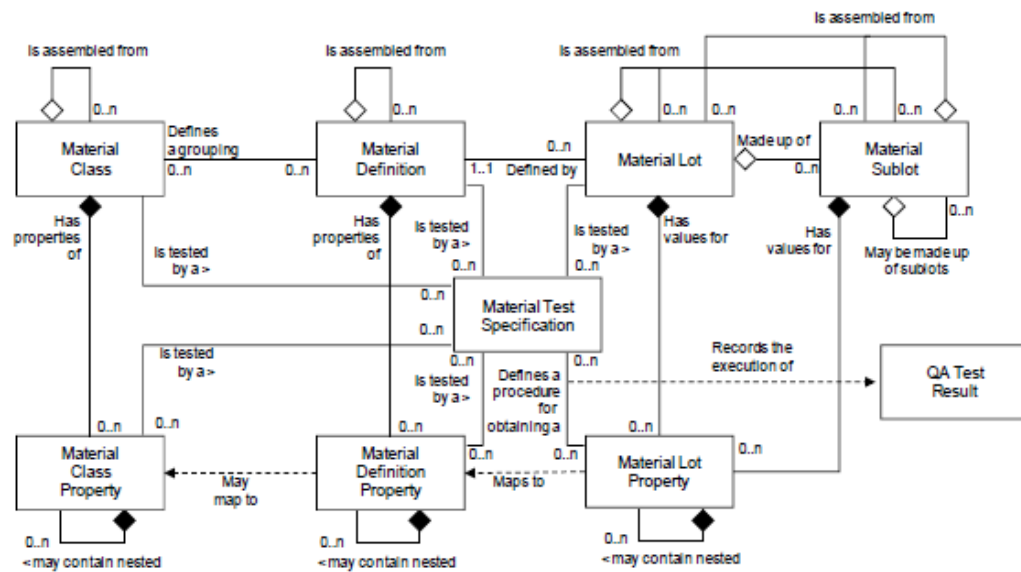


Figure 4.4: Material Model, from [6]

The model is composed of nine classes: Material Definition, Material Class, Material Definition Property, Material Class Property, Material Lot, Material Lot Property, Material Sublot, Material Test Specification and QA Test Result.

For the developing of this project it was implemented for the database only the Material Definition, Material Class, Material Definition Property, Material Class Property, Material Lot and Material Lot Property.

#### 4.5.1 Material Class

A material class is a representation of groupings of material definitions for a definite purpose such as manufacturing operations definition, scheduling, capability and performance.

A material definition shall belong to zero or more material classes.

Table 4.9: Material Class Attributes [6]

| Attribute             | Description                                                                                                                                                                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID                    | A unique identification of a specific material class, within the scope of the information exchanged.                                                                                                                                                                      |
| Description           | Additional information about the material class.                                                                                                                                                                                                                          |
| Assembly Type         | Optional: Defines the type of the assembly.<br>Types:<br>Physical – The components of the assembly are physically connected or in the same area.<br>Logical – The components of the assembly are not necessarily physically connected or in the same area.                |
| Assembly Relationship | Optional: Defines the type of the relationship.<br>Types:<br>Permanent – An assembly that is not intended to be split during the production process.<br>Transient – A temporary assembly using during production, such as a pallet of different materials or a batch kit. |

#### 4.5.2 Material Class Property

Properties of a material class shall be presented as material class properties. A material class shall be further characterized through zero or more material class properties.

Table 4.10: Material Class Property Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the material class property.            |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

#### 4.5.3 Material Definition

A representation of goods with similar name characteristics for the purpose of manufacturing operations definition, scheduling, capability and performance shall be shown as a material definition.

Any material lot shall be associated with one material definition.

Table 4.11: Material Definition Attributes [6]

| Attribute             | Description                                                                                                                                                                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID                    | A unique identification of a specific material class, within the scope of the information exchanged.                                                                                                                                                                      |
| Description           | Additional information about the material definition.                                                                                                                                                                                                                     |
| Assembly Type         | Optional: Defines the type of the assembly.<br>Types:<br>Physical – The components of the assembly are physically connected or in the same area.<br>Logical – The components of the assembly are not necessarily physically connected or in the same area.                |
| Assembly Relationship | Optional: Defines the type of the relationship.<br>Types:<br>Permanent – An assembly that is not intended to be split during the production process.<br>Transient – A temporary assembly using during production, such as a pallet of different materials or a batch kit. |

#### 4.5.4 Material Definition Property

Properties of a material definition shall be defined as material definition properties. A material definition shall be further characterized through zero or more material definition properties.

Material definition properties may contain nested material definition properties.

Table 4.12: Material Definition Property Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the material definition property.       |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

#### 4.5.5 Material Lot

Represents a unique identifiable specific amount of material, countable or weighable, which shall be named a material lot. A material lot describes the planned or actual total quantity or amount of material available, its current state, and its specific property values.

Table 4.13: Material Lot Attributes [6]

| Attribute                | Description                                                                                                                                                                                                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID                       | A unique identification of a specific material class, within the scope of the information exchanged.                                                                                                                                                                      |
| Description              | Additional information about the material lot.                                                                                                                                                                                                                            |
| Assembly Type            | Optional: Defines the type of the assembly.<br>Types:<br>Physical – The components of the assembly are physically connected or in the same area.<br>Logical – The components of the assembly are not necessarily physically connected or in the same area.                |
| Assembly Relationship    | Optional: Defines the type of the relationship.<br>Types:<br>Permanent – An assembly that is not intended to be split during the production process.<br>Transient – A temporary assembly using during production, such as a pallet of different materials or a batch kit. |
| Status                   | Status of the material lot. For example, released, approved, blocked, in process, in quality check.                                                                                                                                                                       |
| Storage Location         | An identification of the storage location or a physical location of the material lot.                                                                                                                                                                                     |
| Quantity                 | The quantity of the material lot.                                                                                                                                                                                                                                         |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                                                                                                                                                                                            |

#### 4.5.6 Material Lot Property

Each material can have unique values for zero or more material lot properties, such as a specific temperature value for the specific lot of material, or a specific density for the lot of material.

Table 4.14: Material Lot Property Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | An identification of the specific property.                          |
| Description           | Additional information about the material lot property.              |
| Value                 | The value, set of values, or range of the property.                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

## 4.6 Process Segment Model

While the Equipment Model defines the resources and the Material Model defines the inputs, the Process Segment Model defines how the manufacturing steps are executed. According to the ISA-95 standard, a process segment represents a logical grouping of personnel, equipment, and material resources required to carry out a specific production step.

In this data-driven architecture, the Process Segment is the fundamental building block of the manufacturing process. It abstracts discrete physical actions—such as "Drilling", "Assembling", or "Transporting", into standardized logical tasks.

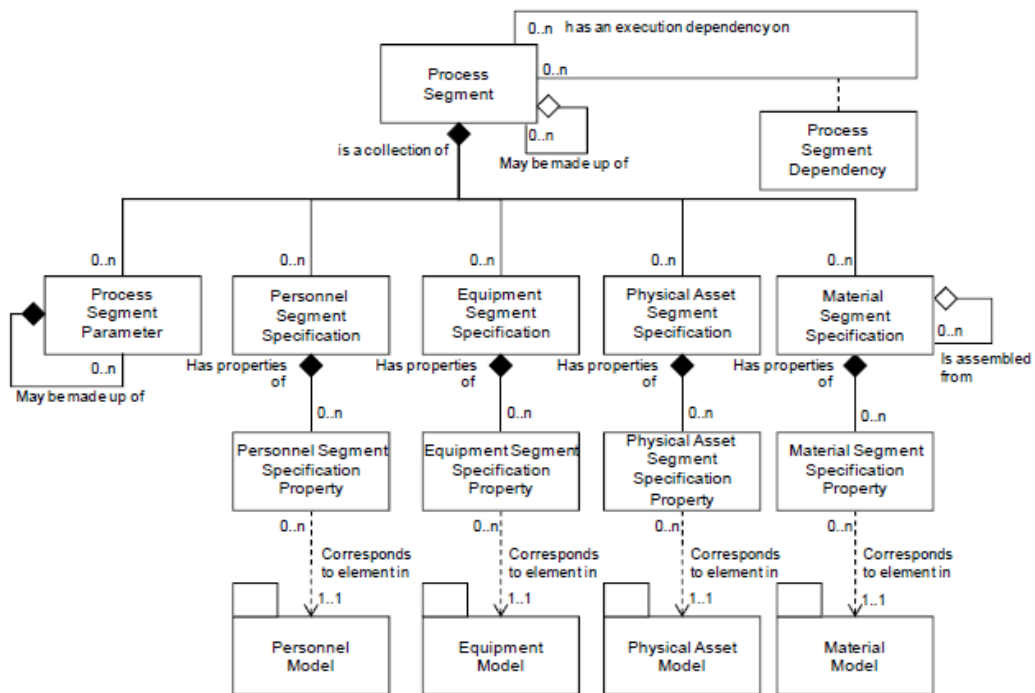


Figure 4.5: Process Segment Model, from [6]

For this project, we focus only on Equipment Segment Specification, Physical Asset Specification and Material Segment Specification.

The individual classes and their attribute table are on Appendix A.

## 4.7 Operations Definition Model

The Operations Definition Model acts as the central repository for manufacturing "recipes". It defines the sequence of operations, the material requirements, and the process segments necessary to produce a specific item. In the ISA-95 framework, this model bridges the gap between high-level production planning (Level 4) and shop-floor execution (Level 3).

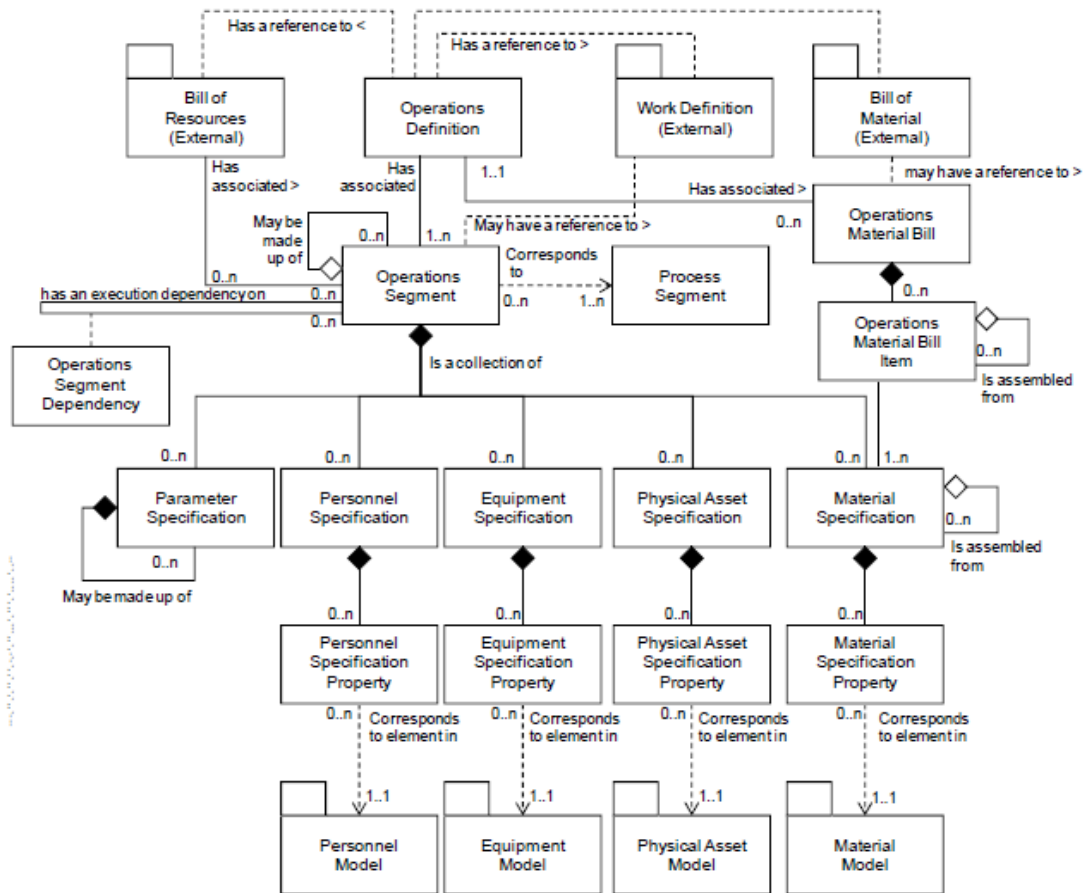


Figure 4.6: Operations Definition Model, from [6]

For the development of this project we focus only on the following classes of the model: Operations Definition, Operations Segment, Operations Segment Dependency, Parameter Specification, Equipment Specification, Physical Asset Specification and Material Specification.

The individual classes and their attribute table are on Appendix A.

## 4.8 Operations Schedule Model

The Operations Schedule Model is the most dynamic component of the proposed system. It represents the active plan for production, the "when", "what", and "how" of shop-floor execution.

An operations schedule shall be made up of one or more operations requests. The operations schedule may apply to scheduling of Production, Maintenance, Quality Test and Inventory.

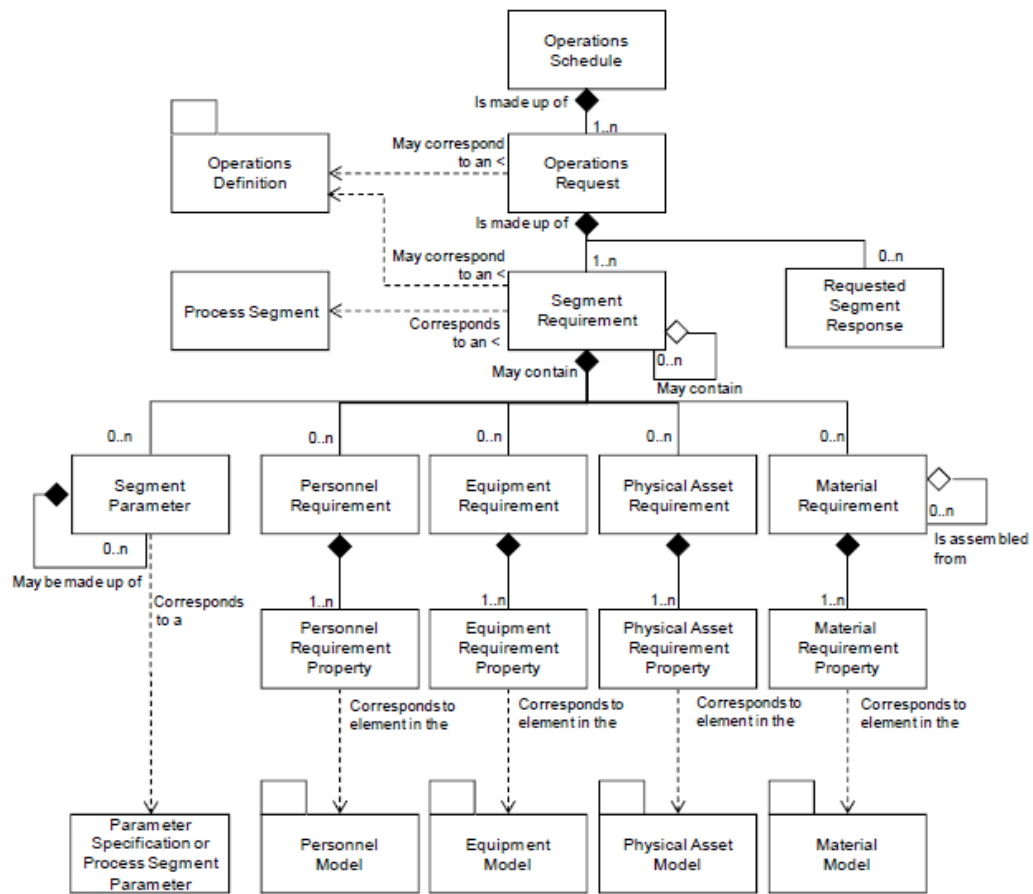


Figure 4.7: Operations Schedule Model, from [6]

For this project, we focused on the following classes: **Operations Schedule**, **Operations Request**, **Segment Requirement**, **Segment Parameter**, **Equipment Requirement**, **Physical Asset Requirement** and **Material Requirement**.

The individual classes and their attribute table are on Appendix A.

## Chapter 5

# System Design and Implementation

This chapter describes the practical implementation of the proposed ISA-95 compatible Manufacturing Execution System (MES). It bridges the theoretical information models detailed in Chapter 4 and their deployment into a functional, lightweight, and database-driven software architecture

### 5.1 Overall View of the System

The practical implementation of the MES system was developed based on a modular, distributed, and bidirectionally interconnected architecture, effectively linking the planning and management level with the physical execution level on the shop floor. The system is structured around four fundamental components: a web-based user interface (Web UI), a centralized relational database in MySQL, the central MES control application developed in Python, and the industrial automation layer simulated in the CODESYS environment through an OPC UA server.

The architectural cornerstone of the proposed system is the data-driven decoupling paradigm. By establishing the centralized MySQL relational database as the single source of truth, the business logic of the MES core and the presentation layer of the Graphical User Interface (GUI) are entirely isolated from one another. This decoupling is essential to fulfill the generic and adaptable nature required by modern Industry 4.0 applications. Rather than hard-coding manufacturing constraints, layout configurations, or execution rules within the Python engine or the PLC code, the entire system state and shop-floor topology are parameterized dynamically. Consequently, changing the production environment or introducing new materials does not need a recompilation of the control software, it merely requires a synchronization of the relational data rows, ensuring high scalability and interoperability.

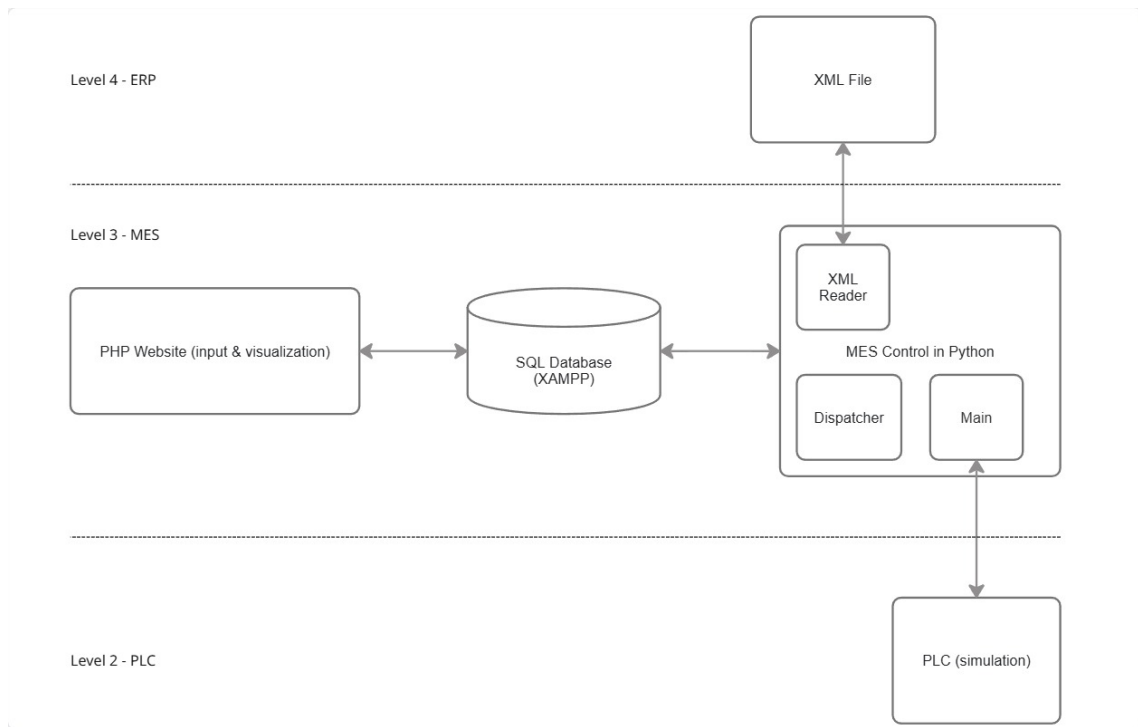


Figure 5.1: Project Overall View

From an information flow perspective, Figure 5.1, the architecture establishes strict boundaries that correspond directly to the functional hierarchy defined by the ISA-95 standard. The ingestion of externally generated XML files simulates the vertical integration between Level 4 (Enterprise Logistics/ERP) and Level 3 (Manufacturing Operations Management/MES). The Python-based core engine acts as the central execution supervisor of Level 3, orchestrating the temporal scheduling and material tracking. Finally, the interaction between the Python engine and the CODESYS environment via the OPC UA protocol bridges Level 3 with Level 2 (Shop-floor Supervisory Control). This industrial communication protocol abstracts the underlying hardware, allowing the core MES to read and write directly to the programmable logic controller (PLC) register variables through structured variables without architectural dependencies.

The information and control flow of the system is divided into two distinct phases: the structural parameterization phase and the production execution phase.

During the structural parameterization phase, the PHP-based web application serves as the main interface between the user and the system. Through this interface, the operator defines and configures the factory's static and structured data, including materials, equipment capabilities, process segments, and other parameters required for system operation. This information is stored in the database and provides the foundation for subsequent planning and control activities. By enforcing strict foreign key constraints at this stage, the system guarantees that no manufacturing order can be executed unless its required process segments, machinery classes, and raw materials are already fully authenticated and configured within the static tables of the database.

In parallel, production orders are introduced into the system through externally generated

XML files. A dedicated software module performs the parsing of these files, extracting the relevant production requirements and inserting the corresponding data into the Operations Schedule table of the database.

The second phase corresponds to production execution and is coordinated by the MES engine developed in Python. Initially, the system retrieves the records stored in the Operations Schedule table and executes the scheduling algorithm, which generates a structured Dispatch Plan according to the defined priority rules and production constraints. Only after the complete generation and consolidation of this dispatch plan does the system proceed to the execution stage. During this phase, the MES shifts from a passive storage schema to an active event-driven monitoring state. The Python engine continuously updates the dynamic tables (such as material lots and execution states), ensuring that the current state of the shop floor is reflected in real time on the web dashboard while synchronous commands are pushed to the physical automation devices.

Based on the generated Dispatch Plan, the MES engine determines the execution sequence for each manufacturing cell and issues the corresponding control commands to the PLC implemented in CODESYS. Communication between the MES and the automation layer is established through the OPC UA protocol, enabling the reliable transmission of production commands and ensuring coordinated shop-floor operation.

## 5.2 Database Implementation

Translating the object-oriented and hierarchical abstractions defined by the ISA-95 standard, presented in Chapter 4, into a physical, relational paradigm represents a core engineering challenge of this work. While the standard conceptually outlines industrial entities through high-level UML class diagrams, the Manufacturing Execution System (MES) developed herein implements these models using a highly normalized MySQL relational schema. To achieve this, conceptual models were mapped into physical tables where relational integrity is strictly enforced at the database level using Primary Keys (PK), Foreign Keys (FK), and associative tables. This approach ensures that the database acts not merely as a passive data repository, but as an active structural validator that prevents operational anomalies before any software execution loop begins.

The database was logically divided into two categories of data:

- **Static Configuration Data:** Tables that map the topology of the production system, such as material (Material Definition), equipment technical capabilities (Equipment), process segment (Process Segment), and operations definition (Operations Definition).
- **Dynamic Execution Data:** Tables that are modified in real time during the production cycle, including the state of the virtual inventory (Material Lot), order scheduling (Operations Schedule) and work request records (Operations Request).

All the database tables structure are provided in Appendix B

### 5.2.1 Equipment Model

To materialize the Role-Based Equipment Model, Figure 5.2, discussed in Section 4.3, the physical topology of the workshop must be mapped without hard-coding specific layout constraints. This flexibility is achieved by treating every manufacturing asset, regardless of whether it represents an entire multi-tool machine or an individual standalone tool, as a distinct row within a centralized equipment table.

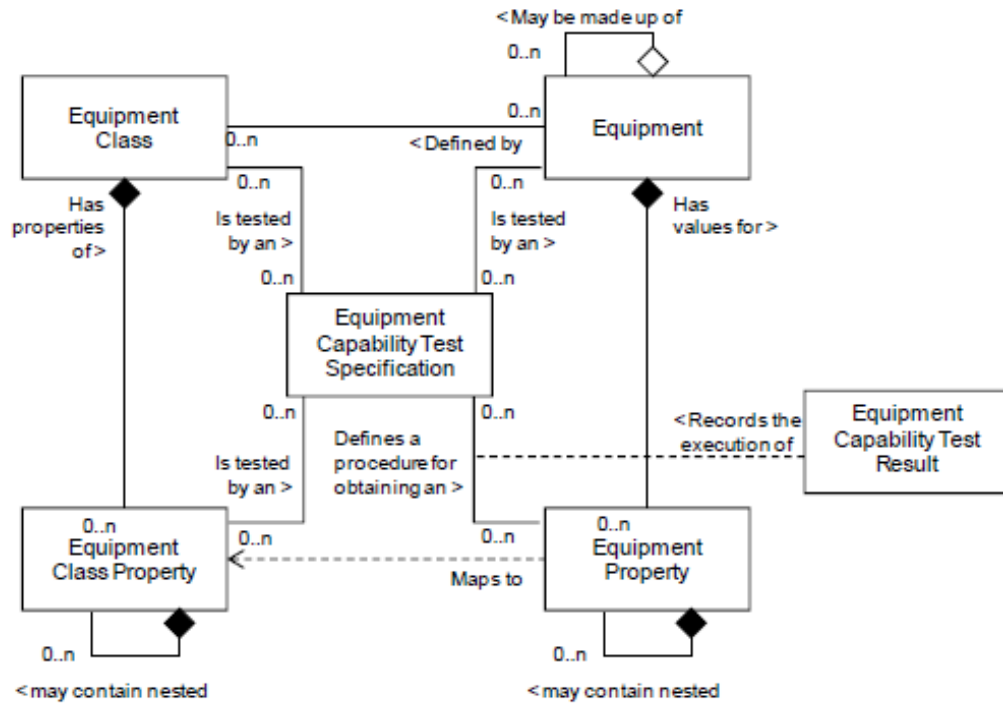


Figure 5.2: Role Based Equipment Model, from [6]

To map the conceptual models defined by the ISA-95 standard into the relational database structure, Figure 5.3, several structural adaptations were required. For many-to-many ( $N : N$ ) relationships, such as the one between *Equipment* and *Equipment Class* shown in Figure 5.2, an associative (junction) table was implemented to map and store all active correlations. Conversely, for one-to-many ( $1 : N$ ) relationships, such as the bond between *Equipment* and *Equipment Property*, the hierarchy was enforced by embedding a Foreign Key (FK) directly into the property table to explicitly point to its corresponding asset. Lastly, the self-referential  $N : N$  composition relationship, representing that a complex piece of equipment can be physically made up of other equipment units, was resolved by creating a dedicated junction table named `equipment_made_up_of`. This table maps the structural hierarchy by storing parent and child identifiers, both acting as Foreign Keys referencing the primary ID of the respective equipment elements.

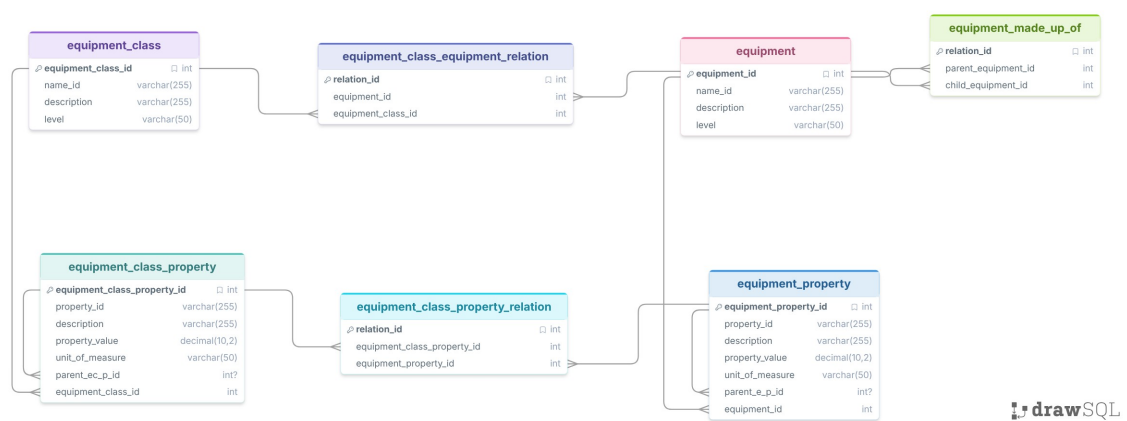


Figure 5.3: Equipment Database Table Structure

To model the real-world reality that complex machines are physically composed of multiple tools, a self-referential hierarchical design was deployed. Instead of creating rigid, separate tables for machines and tools, their structural parent-child dependency is enforced through an associative junction table named `equipment_made_up_of`. This table maps the technical relationships between parent equipment (the machine) and child equipment (the allocated tools). By querying this associative structure, the core MES can dynamically determine asset availability.

### 5.2.1.1 Process Segment and Manufacturing Recipes

The implementation of the manufacturing capability logic relies on the strict translation of the Process Segment Model, detailed in Section 4.6. In an ISA-95 compatible architecture, a process segment represents a logical manufacturing step that aggregates the specific resource requirements needed to execute a transformation, Figure 5.4. Physically, each core operation profile is registered as an independent entry inside the `process_segment` table.

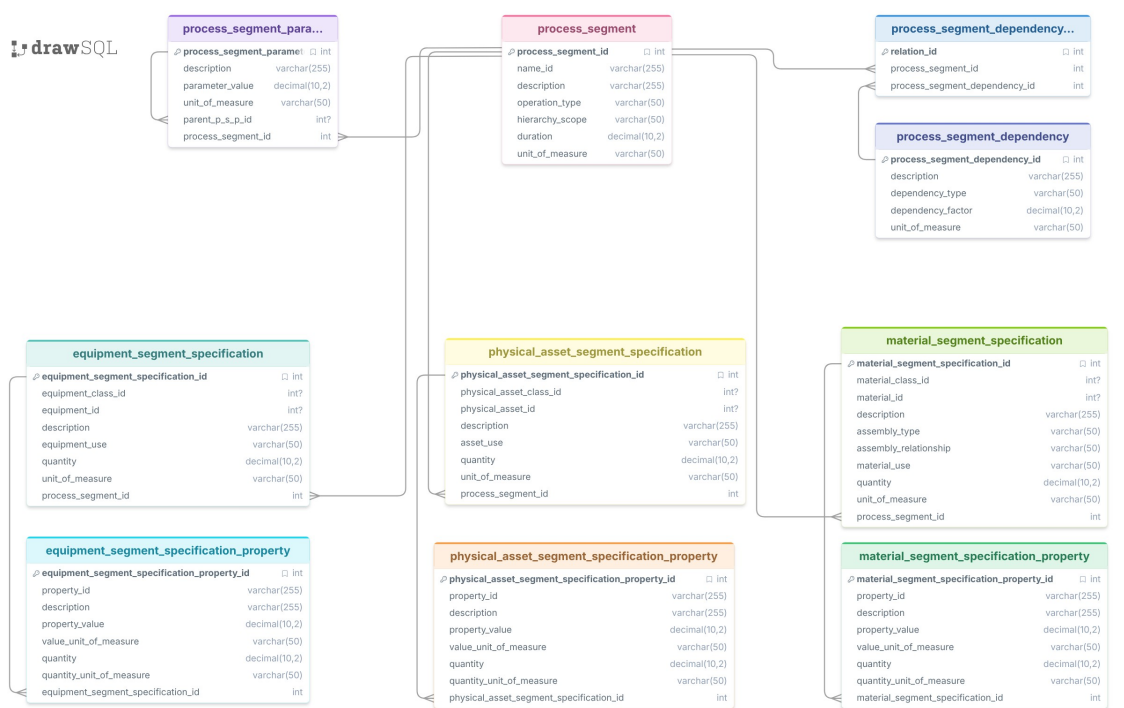


Figure 5.4: Process Segment Database Structure

However, a process segment cannot exist in isolation; it must explicitly define what materials it consumes, what it produces, and what tools it requires. To maintain a clean, third-normal-form (3NF) database schema, these resource specifications were decoupled into secondary tables linked via Foreign Keys:

- **equipment\_segment\_specification**: Binds the process segment to specific hardware requirements, defining the necessary tool codes.
- **physical\_asset\_segment\_specification**: Binds the process segment to specific physical hardware equipment, defining the necessary physical tool codes.
- **material\_segment\_specification**: Outlines the material flow logic by classifying resources into input components (raw materials) or output components (produced workpieces).

This relational mapping ensures that the database automatically validates the production recipes, guaranteeing that the core engine can only schedule transformations where resource and material requirements are perfectly synchronized.

### 5.2.2 Dynamic Scheduling and Production Inventory

While the equipment, physical asset, material, process segment, and operations definition tables establish the static baseline configuration of the factory, the system handles real-time production events through a dedicated group of dynamic tables. The ingestion of production orders from

upper enterprise levels is materialized within the `operations_schedule` table, detailed on Appendix B based on the model presented on Section 4.8, which acts as the dynamic queue for pending production tasks.

Concurrently, material tracking and physical inventory state changes are managed via the `material_lot` table, relates to the Material database, Appendix B. As raw items move through the production system and undergo physical transformations, the Python core engine continuously updates the records in these dynamic tables.

This architectural separation between static parameterization tables and dynamic operational tables prevents data redundancy, optimizes query execution times during the scheduling phase, and provides a continuous, auditable historical log of the shop floor's execution history.

### 5.3 Graphical User Interface

The presentation layer of the developed MES is materialized through a responsive web-based Graphical User Interface (GUI), deployed using an Apache web server, PHP, and Bootstrap for the front-end layout. Choosing a lightweight, server-side scripted web application rather than a heavy standalone desktop client guarantees platform independence, allowing operators to monitor and parameterize the shop floor from any terminal connected to the local network.

Architecturally, the GUI is designed around two functional pillars that mirror the production phases defined in Section 5.1:

1. **Structural Parameterization Interfaces:** Dedicated CRUD (Create, Read, Update, Delete) dashboards allow production engineers to populate the static ISA-95 tables. Through specialized forms, users can declare equipment classes, register individual machines, define materials, and construct manufacturing recipes (process segments) without interacting directly with the SQL command line. The interface enforces input validation to ensure that newly injected parameters comply with the relational integrity constraints of the database, Figure 5.5 and Figure 5.6.

**EQUIPMENT**

NAME ID

DESCRIPTION

EQUIPMENT LEVEL

CLASS

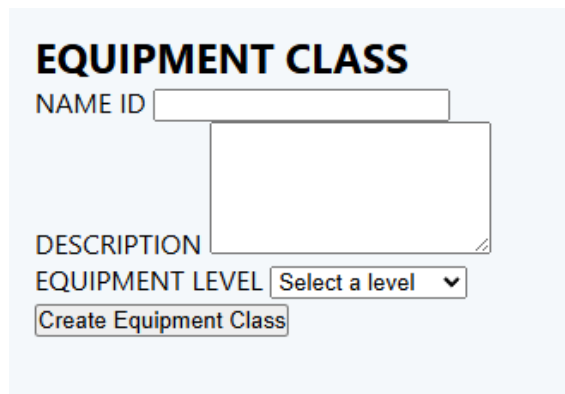
☐ Storage ☐ Transfer ☐ Tool\_T1 ☐ Tool\_T2 ☐ Tool\_T3 ☐ Tool\_T4 ☐ Tool\_T5 ☐ Tool\_T6 ☐ Tool\_T8 ☐ Tool\_T9 ☐ Tool\_T10 ☐ Tool\_T11 ☐ Ma ☐ Mb ☐ Mc

MADE UP OF ☐ No ☒ Yes

EQUIPMENT

☐ W1 ☐ W2 ☐ T ☐ L ☐ U ☐ T1\_001 ☐ T1\_002 ☐ T1\_003 ☐ T1\_004 ☐ T2\_001 ☐ T2\_002 ☐ T2\_003 ☐ T2\_004 ☐ T3\_001 ☐ T3\_002 ☐ T3\_003 ☐ T3\_004 ☐ T4\_001 ☐ T4\_002 ☐ T4\_003 ☐ T4\_004 ☐ T5\_001 ☐ T5\_002 ☐ T5\_003 ☐ T5\_004 ☐ T6\_001 ☐ T6\_002 ☐ T6\_003 ☐ T6\_004 ☐ T8\_001 ☐ T8\_002 ☐ T8\_003 ☐ T8\_004 ☐ T9\_001 ☐ T9\_002 ☐ T9\_003 ☐ T9\_004 ☐ T10\_001 ☐ T10\_002 ☐ T11\_001 ☐ T11\_002 ☐ C1 ☐ M1a ☐ M1b ☐ M1c ☐ C2 ☐ M2a ☐ M2b ☐ M2c ☐ C3 ☐ M3a ☐ M3b ☐ M3c ☐ C4 ☐ M4a ☐ M4b ☐ M4c

Figure 5.5: Equipment Form



The form is titled "EQUIPMENT CLASS" in bold. It contains the following fields and controls:

- NAME ID**: A text input field.
- DESCRIPTION**: A large text area with a small edit icon in the bottom right corner.
- EQUIPMENT LEVEL**: A dropdown menu with the text "Select a level" and a downward arrow.
- Create Equipment Class**: A button.

Figure 5.6: Equipment Class Form

2. **Dynamic Operations Monitoring:** The landing page (Home View) and the detail pages switch the GUI into a passive, data-retrieval monitoring tool. By executing queries against the dynamic tables (such as `operations_schedule`, `equipment_class`, and `material_lot`), the application aggregates operational statistics in real-time. This provides the factory supervisor with an instant visual overview of active batches, queue status, and machine utilization KPIs, completing the visibility loop of the MES, Figure 5.7, Figure 5.8, 5.9, and Figure 5.10 .

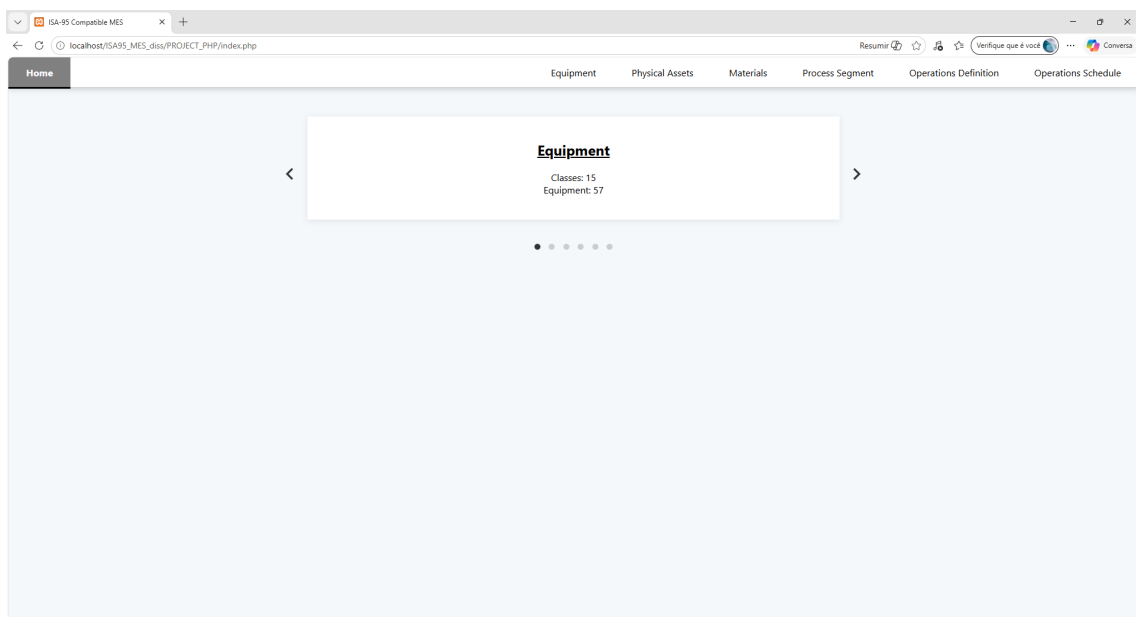


Figure 5.7: Index Page - Example

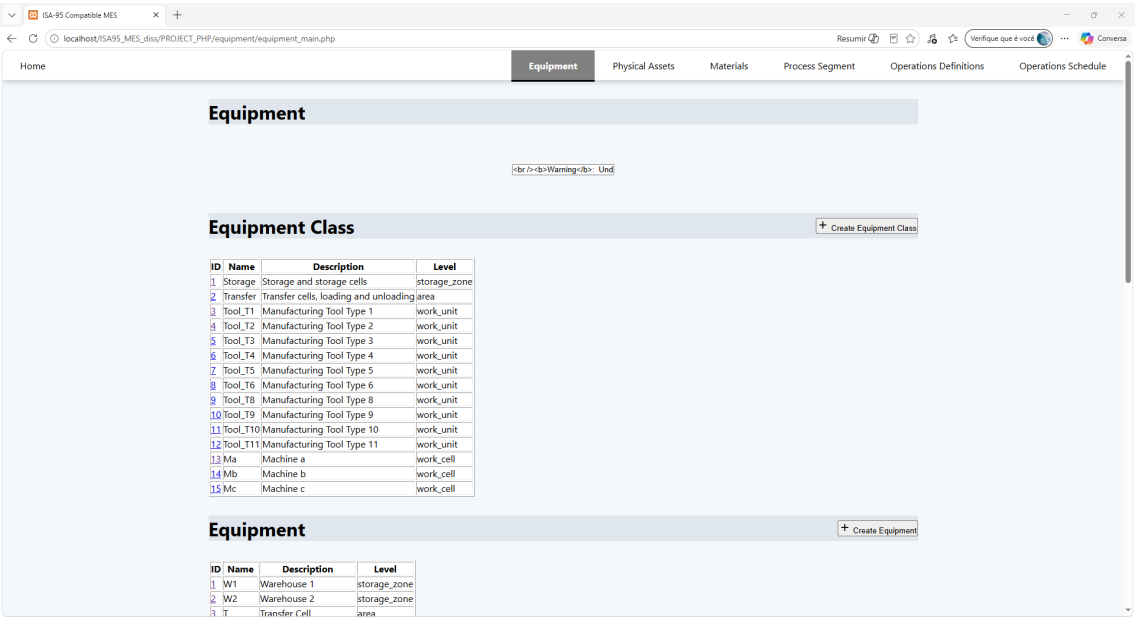


Figure 5.8: Equipment Main Page - Example

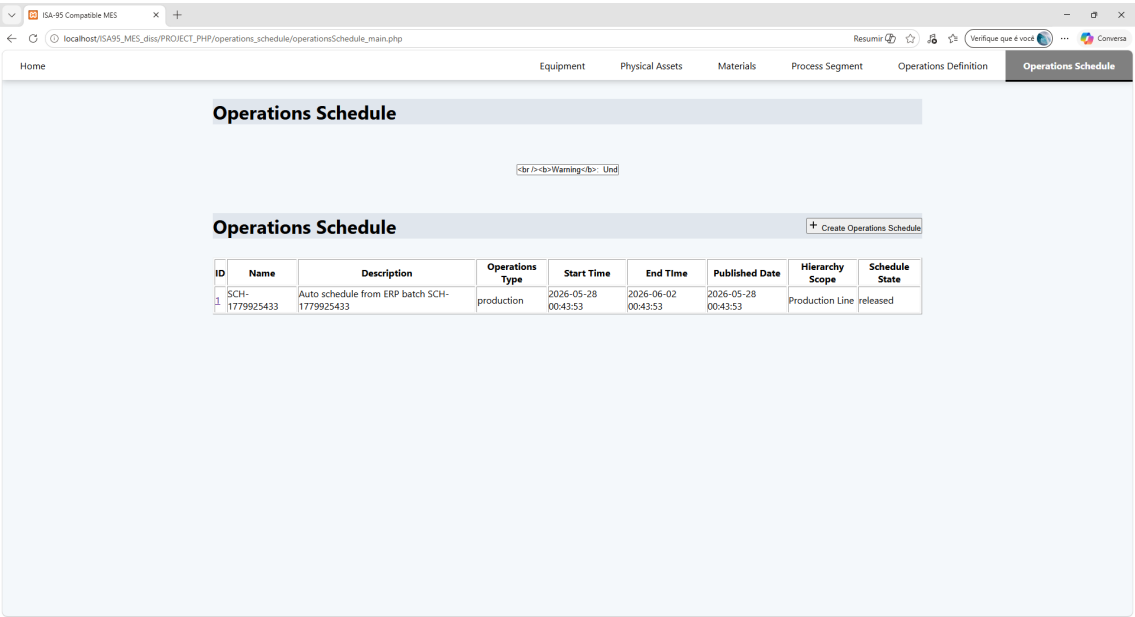


Figure 5.9: Operations Schedule Main Page - Example

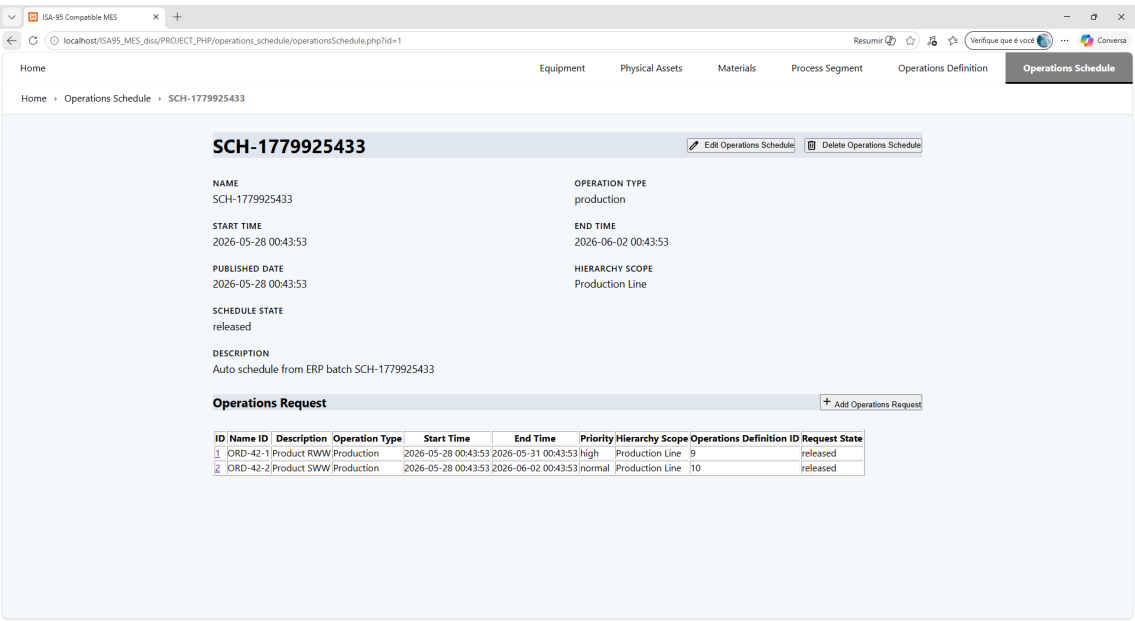


Figure 5.10: Operations Schedule Page - Example

In terms of overall navigation and structural organization, the web platform features a landing page (index.php) integrated with a dynamic carousel component that provides a summary and statistical overview of the current factory data metrics. Access to the different sections of the MES is facilitated by a global navigation bar (navbar), which contains dedicated routing paths corresponding directly to the ISA-95 information models implemented in the database: *Equipment*, *Physical Asset*, *Material*, *Process Segment*, *Operations Definition* and *Operations Schedule*. Each dedicated page within this framework serves a dual purpose, embedding specialized web forms to input new records into the respective class or table, alongside structured views to inspect existing records.

A visual catalog showing some of the available web pages, distinct forms, layout interfaces, and a flow diagram of the website is compiled and referenced in the Appendix C.

## 5.4 MES Core Program (Program MES)

The developed Manufacturing Execution System (MES) is based on a modular, service-oriented and event-driven architecture designed to bridge the gap between the business planning level (ERP) and the physical shop-floor control level (PLCs). This section describes the structural organization of the source code, the underlying data model, and the logical sequence of interactions between the system components.

### 5.4.1 Package Structure and Modular Organization

The MES software ecosystem is organized around the main ‘MES/’ directory. To promote maintainability, scalability, and separation of concerns, the application logic is distributed across specialized sub-packages, each responsible for a distinct functional domain.

- **communication/ (`erp_listener.py`):** Implements the external communication layer and operates as an asynchronous TCP/IP server responsible for receiving production orders from external systems.
- **parser/ (`'xml_parser.py'`):** Provides the translation layer that parses incoming XML documents and converts them into native Python data structures suitable for further processing.
- **core/ (`mes.py`, `scheduler.py`, `material_planner.py`, `dispatcher.py`):** Represents the core execution engine of the MES. This package contains the decision-making algorithms responsible for production scheduling, material planning, dispatching, and workflow orchestration.
- **data/ (`db_connect.py`, `queries.py`):** Implements the persistence layer, encapsulating database connectivity and SQL operations while abstracting access to the underlying relational data model.
- **opcua/ (`plc_client.py`):** Implements the industrial communication layer responsible for interoperability with the shop-floor automation system through the OPC UA protocol.

This modular organization reduces coupling between software components and facilitates future maintenance, testing, and system extensions. The full package structure can view on Appendix D, Section D.1.

### 5.4.2 Sequential Workflow and Component Interaction

The lifecycle of a production order within the MES follows a structured and transactional workflow composed of five major stages. The Workflow diagram can be viewed in represented on Figure 5.11.

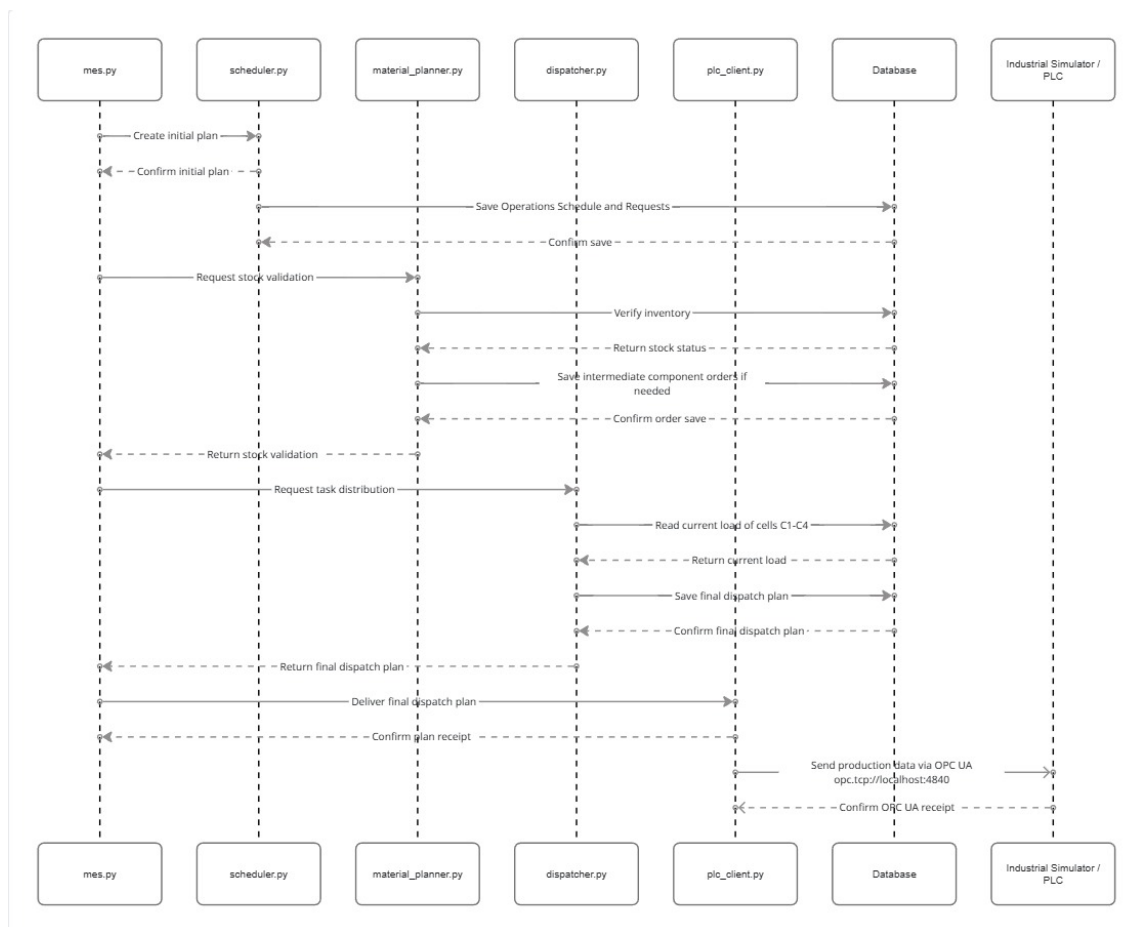


Figure 5.11: Workflow Diagram

#### 5.4.2.1 ERP Data Ingestion and XML Parsing

The transition from enterprise-level manufacturing requests (Level 4) to operational execution (Level 3) is initiated by the data ingestion module. High-level production orders are received in an XML format, design for this project, simulating the typical output of an ERP system.

The data extracted from the ERP is structured in a hierarchical XML format, encapsulated by the root element `<ERP_Export>`. This root node contains a `<ClientOrder>` element, which stores the specific transaction and profile details of a customer. Within `<ClientOrder>`, the system defines three primary metadata fields:

`<Name>`: Stores the commercial name of the client, e.g. *"Fools'n Horses"*.

`<NIF>`: Holds the client's tax identification number, represented by the 9-digit value, e.g. 501234567.

`<OrderID>`: A unique integer identifier assigned to the transaction, e.g. 42.

Furthermore, the client order includes a nested `<Orders>` container that aggregates individual order items under separate `<Order>` tags. In this instance, two specific orders are detailed, each defined by four core attributes:

`<Type>`: An alphanumeric code specifying the order category (e.g., *"RWW"* and *"SWW"*).

<Quantity>: An integer representing the number of units requested (e.g 5 and 10 units, respectively).

<DDate>: An integer field indicating the delivery deadline or relative due date (e.g 3 and 5).

<Penalty>: A numerical value defining the financial penalty applied in case of delay (e.g 50 and 100, respectively).

The Python core engine incorporates a dedicated parsing module (`xml_parser.py`) that utilizes the `xml.etree.ElementTree` library to dynamically traverse the hierarchical file structure.

The parser extracts critical production data, including unique order identifiers, target product codes, quantities, and due dates. Instead of keeping this data in volatile script memory, the module automatically translates the parsed XML attributes into structured SQL `INSERT` statements. These queries inject the records directly into the dynamic `operations_schedule` table, effectively populating the active manufacturing queue and making the enterprise requests available for the scheduling engine.

#### 5.4.2.2 Operations Schedule Generation

Following the ingestion and parsing of production data from the Enterprise Resource Planning (ERP) system by the XML parser, the orchestration flow shifts to the `scheduler.py` module. This module is responsible for synthesizing a deterministic execution timeline and mapping macro-orders into granular, database-backed operational structures. The generation process follows a top-down hierarchical approach that structurally mirrors industrial orchestration standards (such as ISA-95), moving from an overarching schedule down to localized resource requirements.

The routine begins within `create_schedule_from_orders()`, where a macro time frame is established. The scheduling window is bounded between the current system timestamp (`start_time`) and the furthest constraint limit found in the batch, defined mathematically as:

$$end\_time = \max_{o \in orders} (due\_date_o)$$

To prevent redundant execution envelopes within the Manufacturing Execution System (MES) layer, the module queries the persistence layer using `get_operations_schedule_by_end_time()`. If no scheduling envelope conflicts with this frame, a new Operations Schedule entity is instantiated with a unique identifier tokenized by the publication timestamp (SCH-timestamp). This entity establishes the global boundaries, production scope (Production Line), and an initial forecast state. Once the master schedule is finalized, it is decomposed into individual execution threads through `create_requests()`. For every individual order in the batch, an Operations Request is created. Within this phase, a deterministic priority heuristic is applied: the system evaluates the entire batch to find the absolute minimum due date (`min_ddate`). Orders characterized by immediate urgency are flagged with a high priority status, whereas the remaining volume defaults to normal:

$$\text{Priority}(o) = \begin{cases} \text{high}, & \text{if } \text{due\_date}_o \leq \min_{i \in \text{orders}} \text{due\_date}_i, \\ \text{normal}, & \text{otherwise.} \end{cases}$$

This ensures that critical execution paths are explicitly visible to lower-level dispatching mechanisms.

Following request instantiation, the scheduler systematically maps resource dependencies by expanding each request into a Segment Requirement. This step extracts the specific process recipe attributes (`recipe_id`), retrieving the nominal manufacturing duration and technical unit of measure from the database.

The phase concludes with a dual-pronged resource specification expansion:

- **Equipment Requirements:** The scheduler maps the process segment to its mandatory machine constraints via `get_equipment_segment_specification_by_process_segment_id()`, resolving requirements down to explicit equipment classes or unique physical assets.
- **Material Requirements:** The scheduler calculates the total required material volumes by multiplying the recipe's unit factor by the target order quantity. Concurrently, it interfaces with the physical inventory topology, assessing specific storage locations mapped in the database topology (distinguishing between inventory consumption zones and production output warehouses) to ensure structural tracking before execution handover.

### 5.4.2.3 Material Planning and Requirement Expansion

The dynamic validation of resource viability and sub-assembly manufacturing is managed by the `material_planner.py` module. Acting as an execution-level Material Requirements Planning (MRP) engine embedded directly within the MES layer, this module mitigates inventory deficits in real time through closed-loop verification, ensuring that shop-floor dispatching only occurs when material constraints are entirely satisfied.

The core operational logic is driven by a recursive Bill of Materials (BOM) explosion mechanism implemented in `ensure_material_available()`. When the master execution loop inside `mes.py` processes a segment requirement, it isolates the required material inputs and queries their real-time availability. The material planner aggregates physical stock quantities across active tracking lots associated with raw material storage locations via `get_available_stock()`.

If the aggregated stock balance encompasses the requested quantity, the constraint is cleared, returning a validated state. However, if a deficit is identified, the engine triggers an automated requirement expansion process based on the following logical sequence, Figure 5.12:

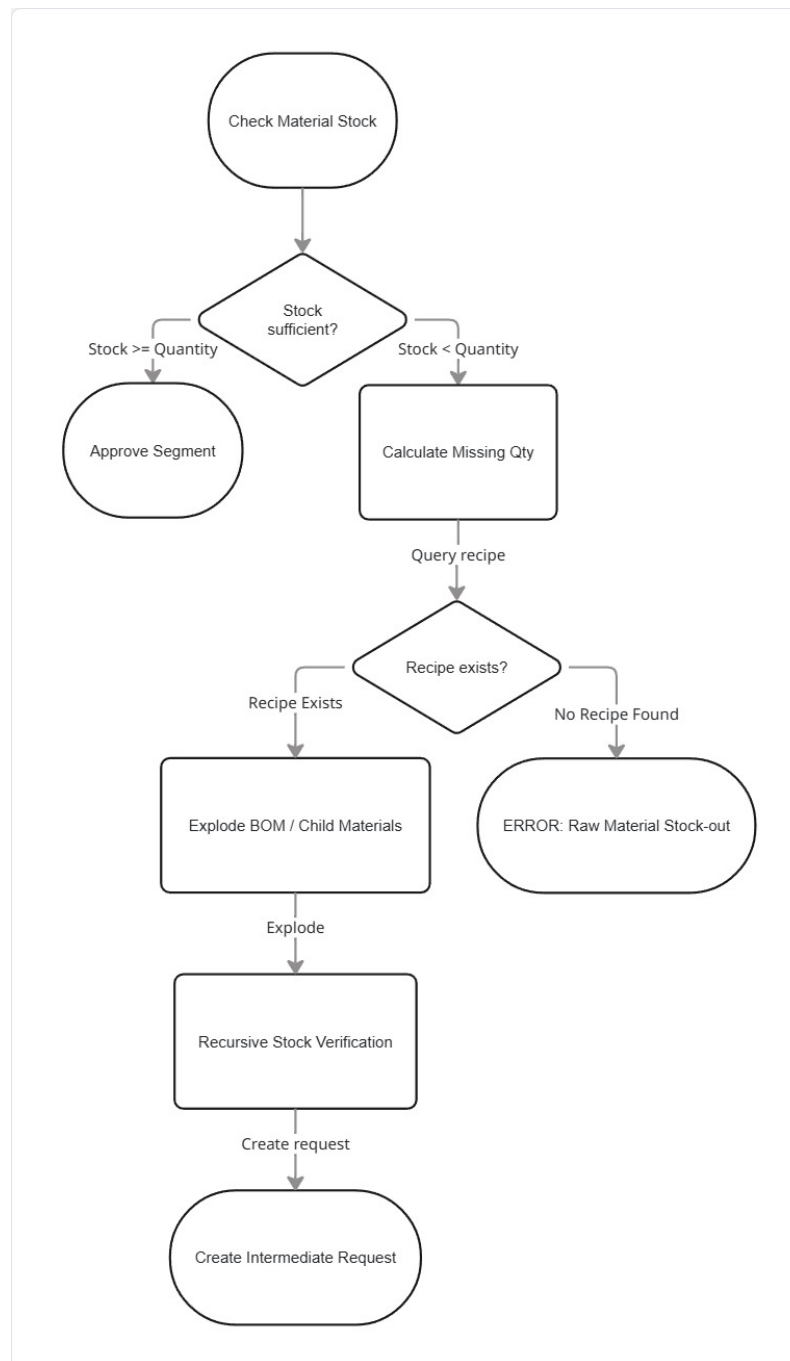


Figure 5.12: Material Planner Logic

1. **Deficit Quantification:** The net shortage is computed by subtracting the available physical lot stock from the targeted execution allocation ( $missing\_quantity = quantity - stock$ ).
2. **Recipe Resolution Typology Classification:** The planner queries the database using `get_recipes_by_material_id()`. If no corresponding recipe segment is registered for the deficient item, the asset is classified as an unresolvable raw material, halting the automation chain and logging an explicit stock-out exception ([ERROR] Raw material).

3. **Recursive BOM Explosion:** If a valid recipe is discovered, the item is recognized as an intermediate sub-assembly. The planner invokes `get_materials_consumed()` to extract its constituent child components. It then recursively executes `ensure_material_available()` for each child dependency, scaling the validation vector to match the missing structural requirements. This depth-first traversal guarantees that multi-level manufacturing dependencies are validated from raw items upward.
4. **Intermediate Production Injection:** Once child-tier availability is structurally assured, the module calls `create_intermediate_request()` to inject an ad-hoc, auto-generated manufacturing requirement into the active schedule context.

The instantiated intermediate requirement bypasses the standard external ERP pipeline, generating a dynamic Segment Requirement identified by an AUTO- prefix. This record inherits the sub-assembly's nominal recipe duration, computing its boundaries relative to the immediate execution window ( $latest\_end\_time = now + duration$ ).

To ensure complete operational execution capability, this dynamic segment undergoes an automated resource expansion identical to primary orders: it automatically binds required machine configurations via equipment specification lookups and instantiates the necessary consumption/production material requirements.

Furthermore, the planning layer incorporates an autonomous inventory safety net. If a material inquiry yields zero physical tracking lots in the database, the planner invokes `create_material_lot()` to instantiate an uninitialized lot structure with a quantity of zero. This structural fallback prevents relational database integrity failures during lookup operations, allowing the system to maintain a clear chain of traceability from raw inputs to intermediate components, and ultimately to the finished product.

#### 5.4.2.4 Dynamic Dispatching and Resource Allocation

Once production tasks, structural process specifications, and material constraints have been fully validated and registered within the persistence layer, the execution horizon shifts to the execution engine governed by the Dispatcher module. The primary objective of this component is to transform passive, asynchronous relational database records into a highly structured, time-sequenced, and optimized physical execution plan.

The dispatching algorithm operates on a multi-tier sorting optimization matrix explicitly designed to minimize manufacturing lead times and prevent physical bottlenecks on the plant floor. Upon invocation via the `dispatch_schedule()` routine, the module extracts all active operational requests associated with the master schedule identifier. To enforce strict compliance with target delivery windows, the engine applies a macro-level Earliest Due Date (EDD) scheduling heuristic. The collection of requests  $R$  is chronologically sorted based on their localized operational end-time constraints:

$$\text{sort}(R) \implies \text{with respect to } \min(e_i), \quad \forall e_i \in \text{end\_time}(r_i), r_i \in R$$

At the micro-level, within each individual request thread, the dispatcher processes a series of segment requirements that must be ordered based on structural manufacturing dependencies. This sequence is established through a nested deterministic evaluation function (`priority_segment`). The engine queries the database to identify the specific material classes generated as an output by each individual process segment.

If a process segment yields a component mapped to an intermediate material (such as sub-assemblies or pre-processed components), it is dynamically assigned a high execution priority ( $\omega = 1$ ). Conversely, segments that produce the final, merchantable goods default to a lower priority ( $\omega = 2$ ). This structural priority mapping ensures that the execution pipeline enforces a strict topological sorting, guaranteeing that prerequisite sub-components are fully manufactured and available before final assembly stages commence.

Resource allocation and machine assignment are handled dynamically through a Greedy Load Balancing heuristic implemented within the `select_equipment_2()` routine. Rather than binding manufacturing operations statically to physical hardware assets during the planning phase, the system evaluates the live state of the shop floor using an in-memory tracking matrix (`cell_workload`) that logs the accumulated workload distributed across each manufacturing cell. For every target segment, the dispatcher resolves the compatible physical assets from the authorized equipment classes. For each candidate piece of physical equipment ( $eq_i$ ), the module performs a reverse relational traversal up the equipment model hierarchy to resolve its structural parent machine and ultimate manufacturing cell designation ( $C_{name}$ ). The algorithm evaluates the cumulative processing load currently allocated to that work cell and selects the optimal asset according to the following optimization function:

$$\text{Selected Equipment ID} = \arg \min_{eq_i} (\text{cell\_workload}[C_{name}(eq_i)])$$

Once the target asset is selected, the engine updates the future load state of that specific cell by accumulating the nominal recipe duration ( $d$ ) defined by the process definitions:

$$\text{cell\_workload}[C_{name}] \leftarrow \text{cell\_workload}[C_{name}] + d$$

This dynamic feedback loop effectively routes sequential operations away from highly utilized work cells, maximizing parallel throughput across the manufacturing lines.

This phase culminates in the generation of a comprehensive, multi-dimensional dispatch ledger object (`dispatch_entry`). This object unifies the relational identifiers of the master schedule, request, and segment requirements with concrete physical execution parameters, including target cell allocations, specific machine identifiers, nominal operational durations, input/output material tracking lot requirements, and total target quantities. This consolidated schema functions as a strict software interlock, halting downstream communication pipelines and blocking automation transmissions until the structural integrity of the entire routing matrix is verified.

### 5.4.3 OPC UA Communication and Four-Stage Handshake Protocol

To close the vertical integration loop between Level 3 (MES) and Level 2 (Shop-floor Control), the finalized Dispatch Plan must be transmitted to the physical automation layer. This interface is managed asynchronously by the `plc_client.py` module, which establishes an OPC UA session with the CODESYS automation server running on the endpoint `opc.tcp://localhost:4840`. On the server side, variables are exposed using the CODESYS *Symbol Configuration* manager, mapping variables from the global PLC architecture to the OPC UA address space.

Because the MES and the PLC operate on different environments, a hybrid sensor-driven synchronization mechanism was developed to orchestrate data transfers and track production without race conditions. For each workpiece in the dispatch plan, the execution follows a strict four-stage sequential flow:

1. **Stage 1 (Recipe Writing and Command Activation):** The Python client prepares the workstation by writing the structured recipe data—including the workpiece identifier (`InitPiece`), the total number of operations (`Last_Operation`), and an array containing specific tool codes and processing times, directly into the PLC nodes. Once the payload is written, the MES sets the boolean command flag `recv_cmd` to `TRUE` to signal the PLC to load the new recipe.
2. **Stage 2 (Physical Entry Tracking):** Instead of waiting for a software acknowledgment flag, the MES synchronizes with the physical flow of the shop floor by continuously polling the workstation's entry sensor node (`In[sensor_top_var]`). The MES blocks and waits until the workpiece is detected arriving at the conveyor (`sensor_top == TRUE`) and subsequently monitors it until it completely clears the entry sensor zone (`sensor_top == FALSE`).
3. **Stage 3 (Interlock Reset and Buffer):** Once the workpiece has safely entered the processing zone and cleared the entry conveyor, the Python client enforces a 1.0-second safety buffer delay. After this timeout, it clears the activation flag by writing `FALSE` to `recv_cmd`. This resets the communication interface, leaving the workstation's entry stage ready to accept a new command for the next piece in the batch.
4. **Stage 4 (Asynchronous Processing and Warehouse Registration):** Concurrently with Stage 3, the MES fires a non-blocking background task (`register_in_warehouse_task`). This task sleeps for the duration of the physical operation and then actively polls the warehouse reception sensor (`In[sensor_bot_var]`). When this exit sensor transitions from `TRUE` to `FALSE` (indicating the finalized piece has entered the warehouse), the task triggers a database transaction via the `Queries` module. It checks the MySQL database for existing stock at storage location, incrementing the material lot quantity or dynamically creating a new material lot if none exists.

## Chapter 6

# Validation and Results

This chapter presents the experimental validation and the results obtained from the deployment of the developed ISA-95 compatible MES. The evaluation focuses on verifying the correctness of the full data pipeline, the alignment of the scheduling logic with the provided case study, and the operational robustness of the sequential communication loop established between the Python core and the CODESYS PLC.

### 6.1 Experimental Setup

The experimental validation environment was established by deploying all core software components on a local workstation to simulate a unified industrial control network. The automation layer was executed using the CODESYS Control Win V3 x64 virtual PLC, which hosted the OPC UA server and simulated the physical behaviors, sensors, and actuators of the manufacturing cells.

The central relational database operated in parallel, handling all data state changes in real time. The MES Python Core acted as the centralized system, establishing client sessions with both the database engine and the OPC UA server. This architecture created a closed-loop system capable of testing the translation of high-level manufacturing orders into deterministic low-level physical operations.

### 6.2 Case Study and Validation Dataset

To evaluate the system under a realistic and industrial scenario, the layout topology, station toolsets, and manufacturing recipes were derived directly from the project framework used as a teaching aid at FEUP, where the goal of this project is to control a factory that produces tables with different materials, wood and metal. Reusing this academic benchmark ensures that the validation is grounded on structured engineering constraints, utilizing official definitions for piece identifiers, processing times, tool codes, and technological routing dependencies.

### 6.2.1 Simulator Layout

During the validation stage, a shop-floor simulator developed by Eng. André Restivo was used [30].

The layout of the shop-floor to be automated consists of 9 cells, Figure 6.1:

- 2 automatic warehouses (W1, W2),
- 1 transfer cell with 5 conveyors to transfer workpieces from W2 to W1 (T),
- 4 machining cells with 3 machines each (C1 to C4),
- 1 loading cell with 5 loading docks (L).
- 1 unloading cell with 5 unloading docks (U).

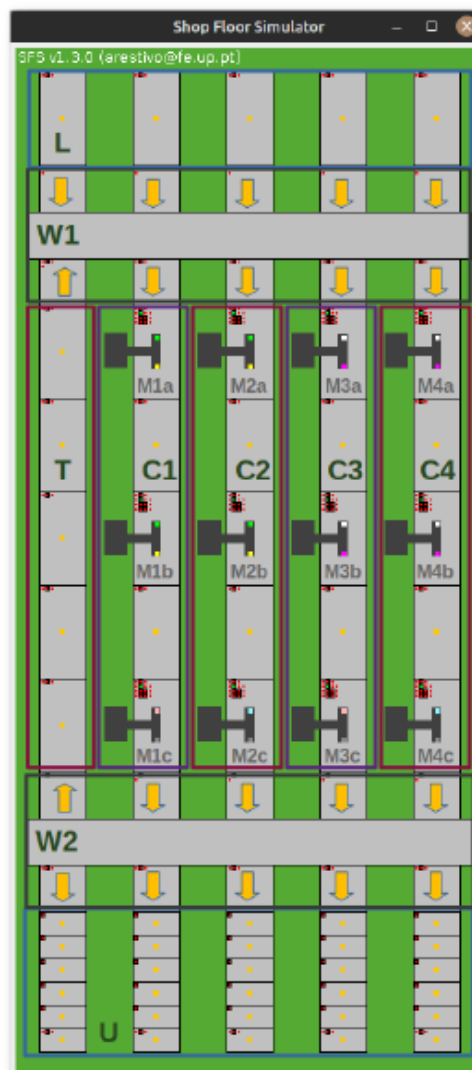


Figure 6.1: Shop-floor Layout

From the Figure 6.1 we can see:

- The large horizontal rectangles represent automatic warehouses (W1 and W2). Coloured dots inside these rectangles will later be shown to represent the workpieces currently stored in that warehouse. The workpiece type is represented by the colour.
- All conveyors (grey or bluish grey rectangles) are bi-directional. The small yellow dots on the conveyors represent sensors that can detect the presence of a part/work-piece;
- Conveyors marked with an orange arrow can either insert (load) or remove (unload) part/work-pieces from the respective warehouse. Each of these stations is dedicated to either loading or unloading.
- The equipment in black represent machines, able to transform a part/work-piece from one type to another.
- Typically raw material (raw parts) will arrive on the top loading stations of W1. These parts will be unloaded onto one of the 4 processing cells (C1, C2, C3, and C4) where they will be transformed into another part/workpiece type, and then stored in W2. Transformed workpieces are then unloaded into the loading docks.
- The transfer cell (T) allows to transfer workpieces from W2 back to W1 if necessary.

### 6.2.2 Simulator Information Data

The static capabilities of the workshop cells and the dynamic product recipes were extracted from the project specification and mapped into the corresponding ISA-95 tables of the relational database. To illustrate how this benchmark parameterization was structured within the MES to guide the execution loop, a sample of the compiled data is presented below. Table 6.1 showcases the static parameterization of the shop-floor equipment, while Table 6.2 describes the transformation that may occur.

### 6.2.2.1 Equipment Data

Table 6.1: Equipment - Tools available on each machine

| Machine  | Tool        |
|----------|-------------|
| M1a, M1b | T1, T2, T3  |
| M1c      | T8, T9, T11 |
| M2a, M2b | T1, T2, T3  |
| M2c      | T8, T9, T10 |
| M3a, M3b | T4, T5, T6  |
| M3c      | T8, T9, T11 |
| M4a, M4b | T4, T5, T6  |
| M4c      | T8, T9, T10 |

This technical specification was mapped directly into the database tables to accurately replicate the shop floor. Every machine and tool is treated as a separate row in the equipment inventory. To model the fact that machines are composed of specific tools, a self-referential hierarchy was deployed.

Taking machine M1a as an example, the machine unit and its specific tools are initially populated as entries in the `equipment` table, as shown in Figure 6.2 and Figure 6.3. Following this insertion, their architectural connection is established in the `equipment_made_up_of` table, which maps the parent-child relations to validate the equipment hierarchy, as verified in Figure 6.4.

|                          |                                                                                            |                                                                                            |                                                                                            |    |        |                                |
|--------------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|----|--------|--------------------------------|
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 6  | T1_001 | Tool Type 1 number 1 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 7  | T1_002 | Tool Type 1 number 2 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 8  | T1_003 | Tool Type 1 number 3 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 9  | T1_004 | Tool Type 1 number 4 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 10 | T2_001 | Tool Type 2 number 1 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 11 | T2_002 | Tool Type 2 number 2 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 12 | T2_003 | Tool Type 2 number 3 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 13 | T2_004 | Tool Type 2 number 4 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 14 | T3_001 | Tool Type 3 number 1 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 15 | T3_002 | Tool Type 3 number 2 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 16 | T3_003 | Tool Type 3 number 3 work_unit |
| <input type="checkbox"/> |  Editar |  Copiar |  Apagar | 17 | T3_004 | Tool Type 3 number 4 work_unit |

Figure 6.2: Equipment Table - Tools Entry

|                          |                                                                                                          |                                                                                                          |                                                                                                          |    |     |                     |           |
|--------------------------|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|----|-----|---------------------|-----------|
| <input type="checkbox"/> |  <a href="#">Editar</a> |  <a href="#">Copiar</a> |  <a href="#">Apagar</a> | 43 | M1a | Machine 1a - Cell 1 | work_cell |
| <input type="checkbox"/> |  <a href="#">Editar</a> |  <a href="#">Copiar</a> |  <a href="#">Apagar</a> | 44 | M1b | Machine 1b - Cell 1 | work_cell |
| <input type="checkbox"/> |  <a href="#">Editar</a> |  <a href="#">Copiar</a> |  <a href="#">Apagar</a> | 45 | M1c | Machine 1c - Cell 1 | work_cell |

Figure 6.3: Equipment Table - Machine M1a entry

| <div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div> |                                                         |                                                         |                                                         | relation_id | parent_equipment_id | child_equipment_id |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------|-------------|---------------------|--------------------|
| <div><div><div></div><div></div><div></div></div></div>                                                                                                    | <div><div><div></div><div></div><div></div></div></div> | <div><div><div></div><div></div><div></div></div></div> | <div><div><div></div><div></div><div></div></div></div> | 1           | 43                  | 6                  |
| <div><div><div></div><div></div><div></div></div></div>                                                                                                    | <div><div><div></div><div></div><div></div></div></div> | <div><div><div></div><div></div><div></div></div></div> | <div><div><div></div><div></div><div></div></div></div> | 2           | 43                  | 10                 |
| <div><div><div></div><div></div><div></div></div></div>                                                                                                    | <div><div><div></div><div></div><div></div></div></div> | <div><div><div></div><div></div><div></div></div></div> | <div><div><div></div><div></div><div></div></div></div> | 3           | 43                  | 14                 |

Figure 6.4: Equipment Made Up Of Table

This data mapping is subsequently displayed on the website’s interface, where the specific profile and tool configuration for machine M1a can be inspected by the operator, as shown in Figure 6.5.

Home

EquipmentPhysical AssetsMaterialsProcess SegmentOperations DefinitionsOperations Schedule

Home > Equipments > M1a

M1a

 [Edit Equipment](#) [Delete Equipment](#)

NAME

M1a

LEVEL

work\_cell

DESCRIPTION

Machine 1a - Cell 1

Properties

 [Add Equipment Property](#)

No equipment properties found.

Classes List

| ID                 | Name | Description | Level     |
|--------------------|------|-------------|-----------|
| <a href="#">13</a> | Ma   | Machine a   | work_cell |

Made Up Of

| ID                 | Name   | Description          | Level     |
|--------------------|--------|----------------------|-----------|
| <a href="#">6</a>  | T1_001 | Tool Type 1 number 1 | work_unit |
| <a href="#">10</a> | T2_001 | Tool Type 2 number 1 | work_unit |
| <a href="#">14</a> | T3_001 | Tool Type 3 number 1 | work_unit |

Figure 6.5: Machine M1a Page

### 6.2.2.2 Transformations Data

Table 6.2: Work-Piece Transformations

| Starting Piece(s)     | Produced Piece | Tool | Processing Time |
|-----------------------|----------------|------|-----------------|
| Wood (1)              | RtopW (3)      | T1   | 30 s            |
| Wood (1)              | StopW (4)      | T2   | 20 s            |
| Wood (1)              | LegW (5)       | T3   | 10 s            |
| Wood (1)              | LegW (5)       | T10  | 30 s            |
| Metal (2)             | RtopM (6)      | T4   | 35 s            |
| Metal (2)             | StopM (7)      | T6   | 25 s            |
| Metal (2)             | LegM (8)       | T5   | 30 s            |
| Metal (2)             | LegM (8)       | T11  | 40 s            |
| RtopW (3), LegW, LegW | RWW (9)        | T8   | 10 s            |
| StopW (4), LegW, LegW | SWW (10)       | T8   | 10 s            |
| RtopW (3), LegM, LegM | RWM (11)       | T9   | 10 s            |
| StopW (4), LegM, LegM | SWM (12)       | T9   | 10 s            |
| RtopM (6), LegM, LegM | RMM (13)       | T8   | 10 s            |
| StopM (7), LegM, LegM | SMM (14)       | T8   | 10 s            |

To map the operational workflows and technological routes into the database, the process segment specifications were systematically integrated into the corresponding database tables. In this model, each logical manufacturing step is recorded as an independent entry inside the process segment table.

To provide a clearer visualization of the manufacturing workflows, the schematic for the transformation "Wood (1) & RtopW (3) & T1 & 30 s" is illustrated in Figure 6.6. The complete set of diagrams for the remaining transformations can be found in Appendix E. Using this specific process segment as an example, the initialization process first requires inserting the primary record of this workflow and the processing time of 30 seconds into the `process_segment` table, as illustrated in Figure 6.7. Subsequently, based on the resource requirements of this specific segment, the corresponding operational specifications are created using Foreign Keys in the secondary tables. The machinery and tool requirements are parameterized in the `equipment_segment_specification` table (associating tool T1), while the product inputs and outputs are registered in the `material_segment_specification` table (defining the raw material Wood and the generated component RtopW), as demonstrated in Figure 6.8 and Figure 6.9. Consequently, this relational structure is dynamically retrieved by the server and rendered on the user interface, showing up in an aggregated view on the website's dedicated page for this process segment profile, as seen in Figure 6.10.

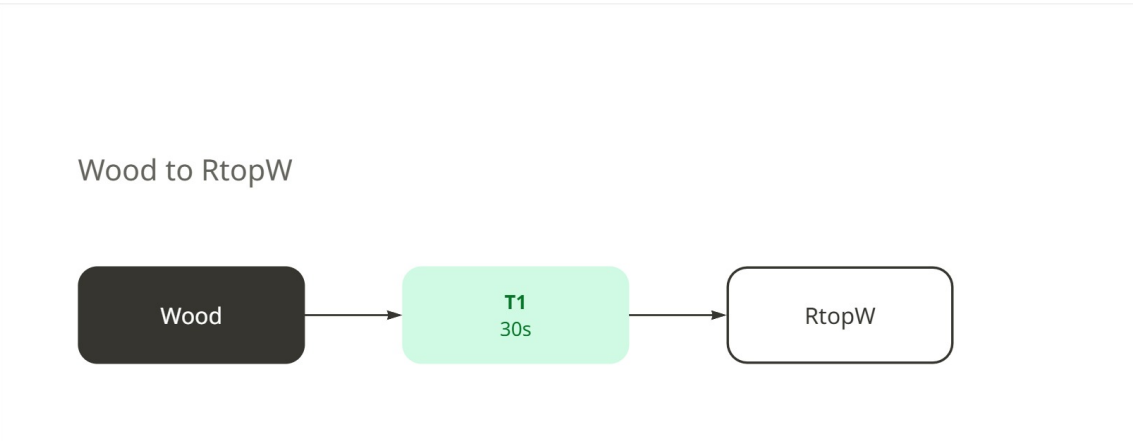


Figure 6.6: Transformation - Wood to RtopW

|                                                 | process_segment_id | name_id            | description                 | operation_type | hierarchy_scope | duration | unit_of_measure |
|-------------------------------------------------|--------------------|--------------------|-----------------------------|----------------|-----------------|----------|-----------------|
| <input type="checkbox"/> Editar  Copiar  Apagar | 1                  | PROC_WOOD_RTOWW_T1 | Wood to RtopW using Tool T1 | production     | Machine         | 30.00    | s               |

Figure 6.7: Process Segment Table - Transformation Wood - RtopW

|                                                 | equipment_segment_specification_id | equipment_class_id | equipment_id | description | equipment_use  | quantity | unit_of_measure | process_segment_id |
|-------------------------------------------------|------------------------------------|--------------------|--------------|-------------|----------------|----------|-----------------|--------------------|
| <input type="checkbox"/> Editar  Copiar  Apagar | 1                                  | 3                  | NULL         | Tool T1     | Transformation | 1.00     | unit            | 1                  |

Figure 6.8: Equipment Segment Specification Table - Transformation Wood - RtopW

|                                                 | material_segment_specification_id | material_class_id | material_id | description  | assembly_type | assembly_relationship | material_use | quantity | unit_of_measure | process_segment_id |
|-------------------------------------------------|-----------------------------------|-------------------|-------------|--------------|---------------|-----------------------|--------------|----------|-----------------|--------------------|
| <input type="checkbox"/> Editar  Copiar  Apagar | 1                                 | NULL              | 1           | Input Wood   | physical      | permanent             | consumed     | 1.00     | unit            | 1                  |
| <input type="checkbox"/> Editar  Copiar  Apagar | 2                                 | NULL              | 3           | Output RtopW | physical      | permanent             | produced     | 1.00     | unit            | 1                  |

Figure 6.9: Material Segment Specification Table - Transformation Wood - RtopW

Home

EquipmentPhysical AssetsMaterialsProcess SegmentOperations DefinitionsOperations Schedule

Home > Process Segment > PROC\_WOOD\_RTOWW\_T1

PROC\_WOOD\_RTOWW\_T1

Edit Process Segment Delete Process Segment

NAME

PROC\_WOOD\_RTOWW\_T1

OPERATION TYPE

production

HIERARCHY SCOPE

Machine

DURATION

30.00

DURATION UNIT OF MEASURE

s

DESCRIPTION

Wood to RtopW using Tool T1

Equipment Segment Specification

Add Equipment Segment Specification

| ID | Equipment Class ID | Equipment ID | Description | Equipment Use  | Quantity | Unit of Measure |
|----|--------------------|--------------|-------------|----------------|----------|-----------------|
| 1  | 3                  |              | Tool T1     | Transformation | 1.00     | unit            |

Physical Asset Segment Specification

Add Physical Asset Segment Specification

No physical asset segment specification found.

Material Segment Specification

Add Material Segment Specification

| ID | Material Class ID | Material Definition ID | Description  | Assembly Type | Assembly Relation | Material Use | Quantity | Unit of Measure |
|----|-------------------|------------------------|--------------|---------------|-------------------|--------------|----------|-----------------|
| 1  |                   | 1                      | Input Wood   | physical      | permanent         | consumed     | 1.00     | unit            |
| 2  |                   | 3                      | Output RtopW | physical      | permanent         | produced     | 1.00     | unit            |

localhost/SAPS\_MES\_diss/PROJECT\_PHP/index.php

Figure 6.10: Process Segment Wood - RtopW Page

### 6.3 Data Pipeline and XML Ingestion Validation

The first stage of validation focused on verifying the MES's ability to ingest external production requirements. A test XML file structured with multiple production orders, containing various final product, requested quantities, and target timelines, was submitted to the ingestion module.

The data extracted from the ERP is structured in a hierarchical XML format, as mention on Chapter 5, Section 5.4.2.1.

The specific XML file utilized for this test is presented below, Figure 6.11:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <ERP_Export>
3    <ClientOrder>
4      <Name>Fools'n Horses</Name>
5      <NIF>501234567</NIF>
6      <OrderID>42</OrderID>
7      <Orders>
8        <Order>
9          <Type>RWW</Type>
10         <Quantity>5</Quantity>
11         <DDate>3</DDate>
12         <Penalty>50</Penalty>
13       </Order>
14       <Order>
15         <Type>SWW</Type>
16         <Quantity>10</Quantity>
17         <DDate>5</DDate>
18         <Penalty>100</Penalty>
19       </Order>
20     </Orders>
21   </ClientOrder>
22 </ERP_Export>

```

Figure 6.11: XML File Used for Test

This XML file represents customer order number 42, associated with the company Fools'n Horses (Tax ID 501234567). The order consists of two different types of products: the first (type RWW, defined in Table 6.2) stipulates the delivery of 5 units within a period of 3 days, subject to a penalty of 50, the second (type SWW, defined in Table 6.2) comprises 10 units with an expected delivery in 5 days and an associated penalty of 100.

The core program successfully parsed the hierarchical XML elements, isolated the relevant dynamic parameters, and mapped them into relational rows within the Operations Schedule database tables. This process confirmed the structural correctness of the data pipeline.

To verify that the processed data was successfully integrated into the system, the parsed records can be examined across the hierarchical views of the presentation layer. The overall batch execution and queued production orders are initially rendered on the general dashboard, as seen in Figure 6.12, which transitions into a granular perspective upon selecting a specific entry, as demonstrated in Figure 6.13. Within this profile, the system isolates individual sub-components and targeted item lines through the Operations Request view, illustrated in Figure 6.14. Finally, the concrete

manufacturing operations, technological transformations, and resource constraints needed to produce the final requested product are fully mapped and detailed inside the Segment Request page, as visualized in Figure 6.15.

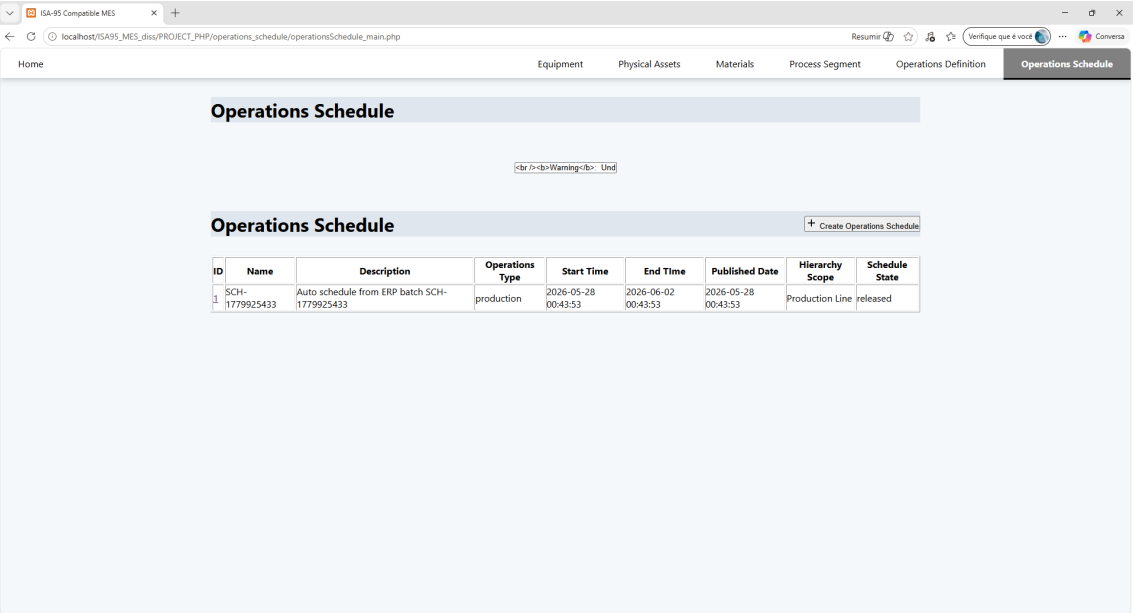


Figure 6.12: Operations Schedule Main Page

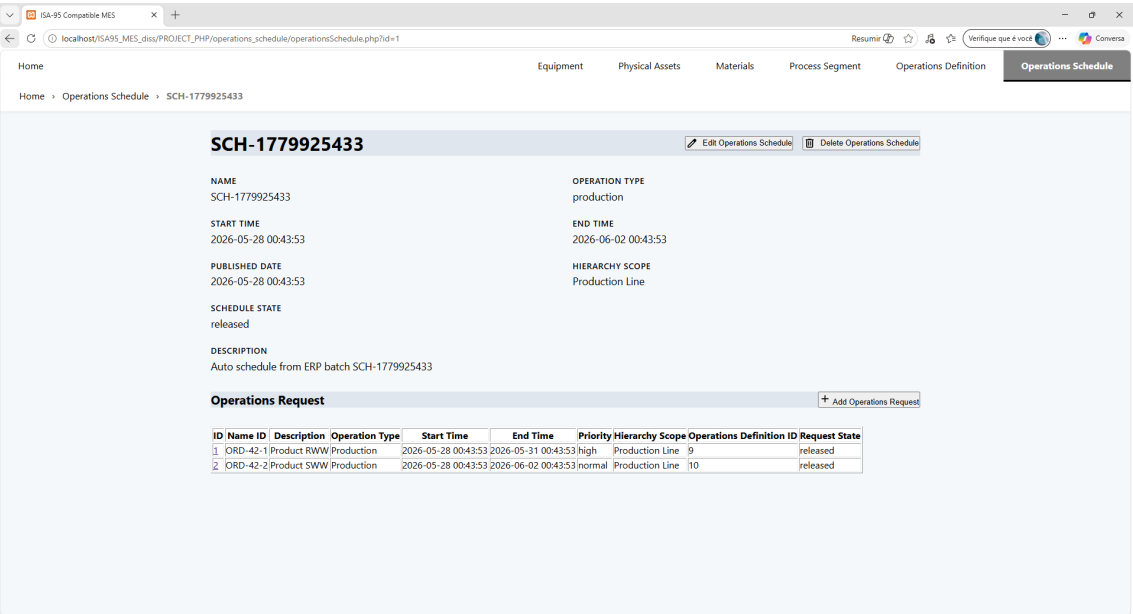


Figure 6.13: Operations Schedule Detail Page

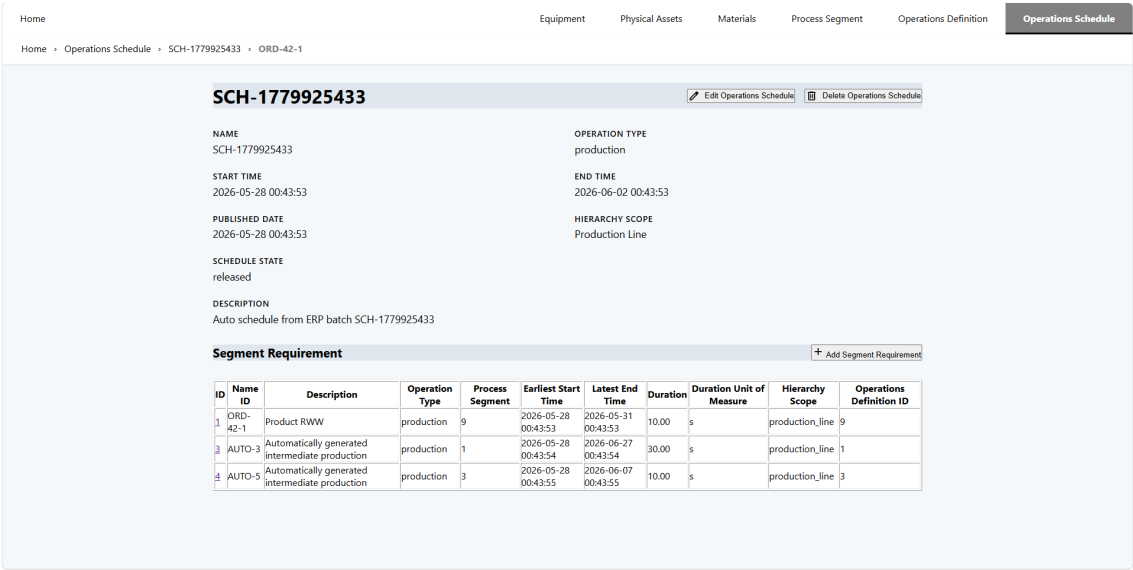


Figure 6.14: Operations Request Page

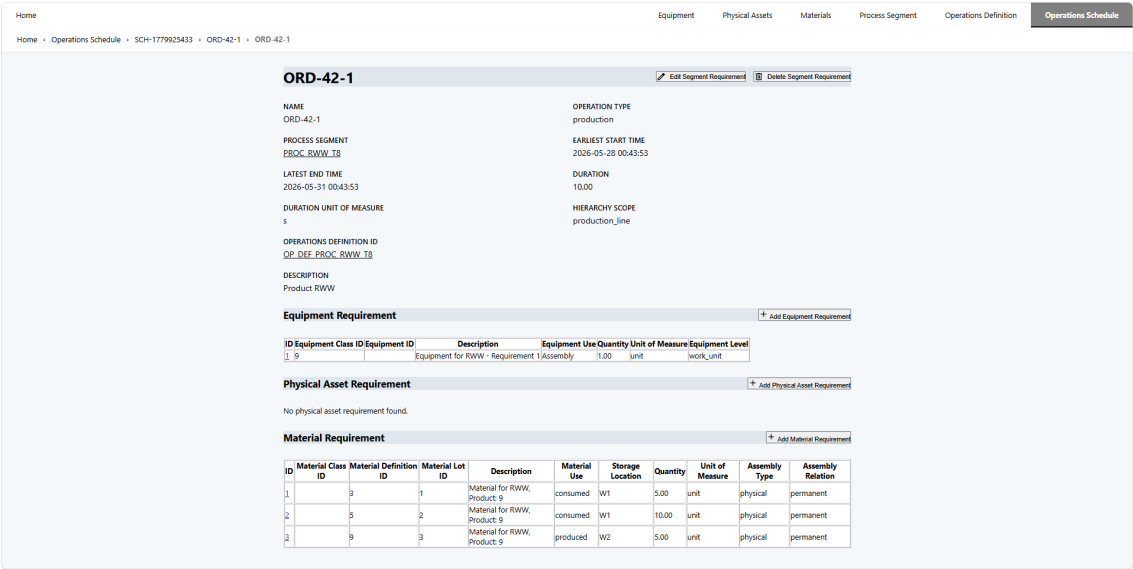


Figure 6.15: Segment Request Page

## 6.4 Dispatch Plan Generation and Scheduling

Once the raw production orders were safely stored in the `operations_schedule` table, the Production Dispatcher module was executed. The objective of this test was to validate the transition from a passive database record to an active, structured execution plan ready for shop-floor deployment.

To achieve this, the Dispatcher operates through a sequential data aggregation process. First, it actively queries the database to identify pending production orders. Upon detecting a new request, the module extracts the requested product and its associated process segment. Using this logical

reference, the Dispatcher automatically cross-references the underlying relational tables, specifically the equipment and material segment specifications, to retrieve the complete technological recipe. This retrieval includes the designated manufacturing cell, the specific tool codes required for the transformation, and the exact processing times allocated for each station.

Instead of sending these parameters individually, the Python core compiles them into a consolidated, in-memory data structure known as the Dispatch Plan. This data object effectively translates the relational SQL data into the array-based format expected by the PLC memory registers. The module correctly organized the chronological order of fabrication, ensuring that each workpiece was explicitly bound to its technological route before any physical action was triggered. Finally, the system successfully held the execution loop until the Dispatch Plan was completely assembled and validated, acting as a software interlock that prevents unsequenced or incomplete orders from being transmitted to the automation layer. The structured data package generated in the Python core memory during this validation test can be observed in the terminal output illustrated in Figure 6.16.

```
===== FINAL DISPATCH PLAN =====
[{"schedule_id": 5, "request_id": 9, "segment_requirement_id": 19, "cell_id": "42", "cell_name": "C1", "machine": "43 - M1a", "equipment_id": 6, "equipment_name": "T1_001", "duration": Decimal("30.00"), "init_piece": [1], "final_piece": [3], "quantity": [5], "operation_id": 1, "operation_name": "PROC_M001_RTOPM_T1"},
{"schedule_id": 5, "request_id": 9, "segment_requirement_id": 20, "cell_id": "46", "cell_name": "C2", "machine": "47 - M2a", "equipment_id": 16, "equipment_name": "T3_003", "duration": Decimal("10.00"), "init_piece": [1], "final_piece": [3], "quantity": [10], "operation_id": 9, "operation_name": "PROC_M001_L604_T3"},
{"schedule_id": 5, "request_id": 9, "segment_requirement_id": 17, "cell_id": "50", "cell_name": "C3", "machine": "53 - M3c", "equipment_id": 32, "equipment_name": "T8_003", "duration": Decimal("10.00"), "init_piece": [3, 5], "final_piece": [9], "quantity": [5, 10], "operation_id": 9, "operation_name": "PROC_M04_T8"},
{"schedule_id": 5, "request_id": 10, "segment_requirement_id": 21, "cell_id": "48", "cell_name": "C2", "machine": "47 - M2a", "equipment_id": 12, "equipment_name": "T2_003", "duration": Decimal("20.00"), "init_piece": [1], "final_piece": [4], "quantity": [10], "operation_id": 2, "operation_name": "PROC_M002_ST09M_T2"},
{"schedule_id": 5, "request_id": 10, "segment_requirement_id": 22, "cell_id": "46", "cell_name": "C2", "machine": "47 - M2a", "equipment_id": 12, "equipment_name": "T3_004", "duration": Decimal("10.00"), "init_piece": [1], "final_piece": [5], "quantity": [20], "operation_id": 3, "operation_name": "PROC_M002_L604_T3"},
{"schedule_id": 5, "request_id": 10, "segment_requirement_id": 18, "cell_id": "54", "cell_name": "C4", "machine": "57 - M4c", "equipment_id": 33, "equipment_name": "T8_004", "duration": Decimal("10.00"), "init_piece": [4, 5], "final_piece": [10], "quantity": [10, 20], "operation_id": 10, "operation_name": "PROC_S04_T8"}
Database connection successful.
```

Figure 6.16: Dispatch Plan

## 6.5 Sequential PLC Control and Handshake Robustness

The core execution logic was validated by running the compiled Dispatch Plan through the sequential OPC UA controller. The system was configured to process one workpiece and one manufacturing cell at a time. This synchronous control model proved to be highly robust, completely avoiding network overhead and preventing data collisions in the shared memory registers of the PLC.

For every individual workpiece processed, the MES Python core successfully performed the four-stage handshake mechanism with the CODESYS PLC\_PRG program variables:

- **Recipe Transmission:** The MES performed synchronous write operations to populate the PLC variables (`InitPiece`, `Last_Operation`, `OpTime`, and `Tool`) with the exact metrics defined in the Dispatch Plan.
- **Command Interlocking:** The MES raised the global control bit `recv_cmd` to `True`, signaling to the PLC that the data was stable and execution could begin.
- **Reactive Monitoring:** The Python script entered an optimized monitoring loop, reading the cell's simulated physical entry and exit sensors via OPC UA. Incorporating small temporal sleep intervals (`asyncio.sleep(0.1)`) successfully prevented CPU starvation and network congestion.

- **Cycle Release:** Upon physical exit of the piece from the conveyor belt, the MES safely reset the `recv_cmd` bit to False, clearing the communication channel and leaving the cell ready to receive the next scheduled part.

The PLC processed every piece according to the transmitted recipe parameters without a single communication timeout or data corruption error, validating the stability of the developed handshake mechanism.

## 6.6 Discussion and Limitations

The experimental results demonstrate that the developed MES successfully implements a lightweight, stable, and database-driven architecture compliant with the core concepts of the ISA-95 standard. The data loop successfully bridged high-level XML planning down to deterministic, sequential shop-floor execution.

However, several critical limitations and architectural boundaries were identified during this validation phase, highlighting areas where the system transitions from a generic framework to a more project-specific implementation:

- **Synchronous Sequential Bottleneck and Limited Multi-Cell Control:** While the sequential execution model guarantees exceptional communication stability and eliminates protocol collisions, it restricts the MES from driving simultaneous, parallel operations across different manufacturing cells. The Python core currently lacks the asynchronous multi-threading capabilities required to control and monitor diverse robotic cells concurrently; each workpiece must complete its lifecycle or handshake segment before the main loop commands the next resource.
- **Simplified Scheduling Algorithm:** The Production Dispatcher uses a very straightforward sequencing logic primarily bounded by the order's *Due Date*. While functional for the case study, this approach does not account for complex industrial scheduling variables, such as dynamic machine bottlenecks, setup time optimization, or multi-criteria heuristic adjustments.
- **Incomplete Closed-Loop Intermediate Logistics:** Although the system monitors material states, the logic is not fully implemented to dynamically instruct the PLC to route intermediate or semi-finished components back into the production line to complete a final product composite. Currently, the automated replenishment and transfer of parts between storage locations are handled locally at the PLC level, meaning the MES core lacks the programmatic capability to calculate or track these background material movements.
- **Restricted Presentation Layer Functionality (GUI Constraints):** The web interface was developed focusing almost entirely on the initial structural parameterization and real-time dashboard data visualization. Consequently, full CRUD (Create, Read, Update, Delete)

capabilities are incomplete, as the GUI currently lacks implemented front-end operations to directly edit or delete existing database entries.

- **Absence of Cascade Depletion at the Database Level:** Although the relational database is robustly structured, the relational schema purposefully lacks `ON DELETE CASCADE` constraints across several core tables. To prevent accidental orphan records and maintain data traceability, users cannot delete an entry regardless of its dependencies; instead, data must be removed manually in a strict, bottom-up hierarchical sequence.
- **Semi-Generic Architectural Scope:** Despite being designed as a generic, metadata-driven solution applicable to different industrial layouts, certain operational definitions, hardcoded handshake variables, and physical routing assumptions remain tightly coupled with this specific laboratory project, classifying the current deployment as a semi-generic framework rather than a fully universal MES platform.

Despite these limitations, the system fully achieves its primary goal, proving that a lightweight Python-based core combined with an ISA-95 relational database can reliably manage and execute industrial production recipes on a modern OPC UA architecture.

## Chapter 7

# Conclusion and Future Work

This final chapter summarizes the core contributions of this dissertation, reviews the objectives achieved during the implementation of the ISA-95 compatible MES, and discusses potential avenues for future research and development based on the identified system limitations.

### 7.1 Conclusion

The primary objective of this dissertation was the development and validation of a lightweight, database-driven MES architecture capable of bridging high-level manufacturing planning with deterministic shop-floor execution. By anchoring the system design on the core structural principles of the ISA-95 standard, a clear abstraction layer was successfully created, ensuring data integrity and standardizing how industrial assets, materials, and process operations are represented.

The practical validation of the system demonstrated the success of the implemented modular data pipeline. The integration of an XML ingestion module successfully simulated an ERP-to-MES interface, automating order ingestion directly into the relational database. Furthermore, the Production Dispatcher proved capable of consuming these raw records and compiling a structured, in-memory Dispatch Plan rooted in the realistic operational constraints of the FEUP Industrial Informatics case study.

At the execution level, the sequential and synchronous OPC UA controller established a highly stable communication bridge with the CODESYS PLC. The developed four-stage handshake mechanism ensured a deterministic cycle of recipe transmission, execution interlocking, and reactive sensor monitoring. By processing workpieces sequentially, the system completely eliminated network overhead, data corruption, and session collisions within the PLC shared memory registers.

Additionally, the development of the PHP Web UI provided a localized, intuitive, and portable platform for operators to configure static shop-floor assets and inspect dynamic data logs without requiring local software installations. In conclusion, despite the simplified execution model, the developed architecture fulfills its core engineering requirements, offering a robust, traceable, and standards-compliant solution for modern digital manufacturing environments.

## 7.2 Future Work

While the developed MES successfully validated the core data flows and synchronization loops required by the ISA-95 standard, the experimental phase revealed several functional constraints that establish a foundation for future research and optimization:

- **Transition to Concurrency and Asynchronous Parallel Execution:** The current sequential model processes one manufacturing cell and one piece at a time to ensure communication stability. Future iterations should focus on implementing asynchronous concurrency (using Python's `asyncio` framework combined with distinct, isolated OPC UA client session contexts). This upgrade would allow the MES to drive parallel production operations across multiple manufacturing cells simultaneously, significantly increasing the plant's throughput.
- **Advanced Adaptive Scheduling and Real-Time Disruption Management:** The current Production Dispatcher relies on a deterministic sequencing model bound by due dates. Integrating dynamic scheduling algorithms (such as genetic algorithms or heuristic-based dispatching) would allow the MES to optimize schedules in real time. Beyond baseline optimization, future research should incorporate event-driven monitoring mechanisms designed to autonomously detect unexpected shop-floor disruptions. Upon identifying unforeseen events, such as sudden machine failures, tool wear anomalies, or resource bottlenecks, the core engine should possess the algorithmic capability to automatically reroute pending manufacturing orders and trigger instantaneous rescheduling routines. This proactive operational adaptability is critical to minimize propagation delays, optimize alternative asset utilization, and mitigate the overall cascading impact of disruptions on the global production plan.
- **Automated Internal Logistics and Intermediate Closed-Loop Routing:** Currently, the physical replenishment and transfer of material lots between storage locations are handled locally by the PLC logic. Future versions of the MES should incorporate an active tracking loop to monitor these background material movements. Crucially, this module should implement closed-loop decision logic to dynamically instruct the PLC to route intermediate, semi-finished components back into specific manufacturing cells when multi-stage transformations are required to complete a composite final product.
- **Production Traceability and Material Genealogy:** While the current framework monitors active material lots, it lacks a formal product genealogy infrastructure. Future work should focus on implementing complete forward and backward traceability loops. This module would allow the MES to automatically record the exact provenance of every raw material item, precisely log which sub-components were consumed within specific process segments, and track the exact machine routing history of every finished workpiece, fulfilling strict industrial quality and audit requirements.

- **Statistical Production Analytics and Performance KPIs:** The core engine currently logs raw event timestamps but lacks a built-in statistical processing layer. A major avenue for future development is the implementation of an analytics module dedicated to calculating critical manufacturing Key Performance Indicators (KPIs). This includes tracking precise machine cycle times, operational idle durations, setup overheads, and Overall Equipment Effectiveness (OEE), providing factory supervisors with visual, data-driven insights directly on the Web UI.
- **Full Integration of All ISA-95 Class Models within the Schema:** The developed relational database currently implements a strategic subset of the ISA-95 standard optimized for the case study. To transform this prototype into a universal repository, future iterations must systematically expand the database architecture to map every single object model and class proposed by the standard. This namely entails deploying the complete relational tables for Personnel Model, detailed Maintenance request tracking, Quality testing specifications, and Physical Asset validation models, ensuring 100% structural coverage of the standard.
- **Database Scaling and Real-Time Event Logging:** As production volume increases, the relational database model may experience bottlenecks due to continuous read/write cycles from synchronous loops. Future work could explore the integration of time-series databases or Message Queues (such as MQTT or Apache Kafka) to handle high-frequency events, performance metrics (KPIs), and alarm history logging from the shop floor.
- **Standardization of Data Ingestion via B2MML:** The current XML parsing module relies on a simplified, custom-built XML schema designed specifically to simulate ERP production orders for testing purposes. To achieve full industrial interoperability, future iterations should upgrade the ingestion engine to natively parse and process B2MML (Business To Manufacturing Markup Language) files. As the official XML implementation of the ISA-95 standard, adopting B2MML would guarantee seamless, out-of-the-box integration with commercial high-level enterprise systems (e.g., SAP, Oracle) and ensure strict compliance with global data exchange protocols.
- **Vertical Order Completion Feedback Loop from Shop-Floor to Enterprise:** The current architecture successfully captures execution completion flags at the PLC layer (Level 2) and updates the local MES database (Level 3). However, it lacks an automated upstream reporting mechanism to notify the higher enterprise logistics layer (Level 4/ERP). Future development should focus on implementing a complete end-to-end bidirectional notification pipeline. Upon detecting a final production sequence completion from the PLC, the MES core should dynamically compile an outbound production performance report—utilizing standardized B2MML Response or Production Performance schemas—and push an asynchronous alert back to the ERP system. This upgrade would achieve true closed-loop vertical integration, enabling real-time order status reconciliation, automated resource depletion tracking, and instantaneous enterprise-level production accounting.

- **Web UI Enhancements and Full CRUD Capabilities:** The current implementation of the Web UI focuses primarily on data insertion and visualization. Future development should aim to deliver full CRUD (Create, Read, Update, Delete) functionalities by integrating dedicated "Edit" and "Delete" actions for all database entries directly within the frontend tables. Furthermore, the platform's aesthetics and user experience (UX/UI) should be refined using modern CSS frameworks to match industrial dashboard standards. Additionally, to abstract the manual deletion restrictions currently enforced to protect database integrity, a backend relational-manager layer should be developed to handle safe, automated cascade deletions across multi-dependent tables.
- **Authentication and Security (Login Option):** To transition the web application from a local laboratory prototype to a secure industrial tool, a robust user authentication and authorization module must be deployed. Implementing a secure login system with Role-Based Access Control (RBAC) would ensure that only authorized personnel (e.g., plant managers, operators, or maintenance technicians) can alter critical factory configurations or submit new production schedules.
- **Comprehensive Fault Handling and Error Reporting:** The current core system is optimized for nominal, error-free execution. Future work should introduce an integrated exception management framework to actively monitor network timeouts, OPC UA communication dropouts, and physical PLC faults. Upon detecting a failure, the MES should automatically log a timestamped error report into the database and trigger real-time visual alerts on the Web UI to accelerate troubleshooting and minimize plant downtime.

# References

- [1] International Society of Automation. “ISA-95 Standards for Enterprise-Control System Integration.”[Online]. Available: <https://www.isa.org/standards-and-publications/isa-standards/isa-95-standard>.
- [2] IBM Corporation. “What is industry 4.0?” Accessed: 2026-01-05. [Online]. Available: <https://www.ibm.com/think/topics/industry-4-0>.
- [3] K. Schweichhart, “Reference architectural model industrie 4.0 (RAMI 4.0) – an introduction,” in *Proceedings of Plattform Industrie 4.0*, Available at the European Commission Futurium portal: [https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference\\_architectural\\_model\\_industrie\\_4.0\\_rami\\_4.0.pdf](https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf), Berlin, Germany, 2016.
- [4] *Enterprise-control system integration – Part 1: Models and Terminology*, International Society of Automation and American National Standards Institute, 2010.
- [5] *Enterprise-control system integration – Part 3: Activity Models of Manufacturing Operations Management*, International Society of Automation and American National Standards Institute, 2005.
- [6] *Enterprise-control system integration – Part 2: Object Model Attributes*, International Society of Automation and American National Standards Institute, 2010.
- [7] J. Gerstl. “Manufacturing execution systems (mes) – a comprehensive guide.” Blog article, accessed: 2026-01-06, GE Vernova. [Online]. Available: <https://www.gevernova.com/software/blog/manufacturing-execution-systems-mes-comprehensive-guide>.
- [8] I. Harjunoski and R. Bauer, “Sharing data for production scheduling using the ISA-95 standard,” *Frontiers in Energy Research*, vol. 2, p. 44, 2014. DOI: 10.3389/fenrg.2014.00044. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fenrg.2014.00044/full>.
- [9] Sepasoft. “MES Enterprise – Global MES Enterprise Software.” Accessed: 2026-01-06. [Online]. Available: <https://www.sepasoft.com/products/mes-enterprise/>.
- [10] Sepasoft, *MES Enterprise Functionality Datasheet*, Product datasheet, Accessed: 2026-01-06, 2024. [Online]. Available: [https://s3.us-east-1.amazonaws.com/files.sepasoft.com/website/datasheets/2024%2BMES%2BEnterprise%2BFunctionality\\_091324.pdf](https://s3.us-east-1.amazonaws.com/files.sepasoft.com/website/datasheets/2024%2BMES%2BEnterprise%2BFunctionality_091324.pdf).
- [11] Inductive Automation. “What is MES?” Accessed: 2026-01-05. [Online]. Available: <https://inductiveautomation.com/resources/article/what-is-mes>.
- [12] Inductive Automation. “Ignition SCADA software.” Accessed: 2026-01-06. [Online]. Available: <https://inductiveautomation.com/scada-software/>.

- [13] IBM Corporation. “What is a manufacturing execution system (mes)?” Accessed: 2026-01-05. [Online]. Available: <https://www.ibm.com/think/topics/mes-system>.
- [14] Critical Manufacturing, *Who we are*, <https://www.criticalmanufacturing.com/company/who-we-are/>, Accessed: Jun. 27, 2026, 2025.
- [15] Critical Manufacturing, *Complete modular solution*, <https://www.criticalmanufacturing.com/mes-for-industry-4-0/complete-modular-solution/>, Accessed: Jun. 27, 2026, 2025.
- [16] Critical Manufacturing, *Manufacturing execution system (mes) for industry 4.0*, <https://www.criticalmanufacturing.com/mes-for-industry-4-0/>, Accessed: Jun. 27, 2026, 2025.
- [17] Critical Manufacturing, *Material logistics*, <https://help.criticalmanufacturing.com/tutorials/modules/material-logistics/materiallogistics/>, Accessed: Jun. 27, 2026, 2025.
- [18] Critical Manufacturing, *Mes migration*, <https://www.criticalmanufacturing.com/mes-for-industry-4-0/mes-migration/>, Accessed: Jun. 27, 2026, 2025.
- [19] Traveller iMES. “iMES – Manufacturing Execution System as simple as tap.” Accessed: 2026-01-06. [Online]. Available: <https://imes.app/>.
- [20] Spruik Technologies. “Libre: Open Source Manufacturing Execution and Performance Monitoring.” Version 1.1.0, accessed: 2026-01-06. [Online]. Available: <https://github.com/Spruik/Libre>.
- [21] Apache Software Foundation. “Apache OFBiz® Project.” Accessed: 2026-01-06. [Online]. Available: <https://ofbiz.apache.org/>.
- [22] Apache Software Foundation. “Apache OFBiz® Framework Repository.” Accessed: 2026-01-06. [Online]. Available: <https://github.com/apache/ofbiz-framework>.
- [23] frePPLe bv. “frePPLe – Open-Source Advanced Planning and Scheduling Software.” Accessed: 2026-01-06. [Online]. Available: <https://frepple.com/>.
- [24] MDCplus. “Top free & open-source MES software for 2025.” Accessed: 2026-01-05. [Online]. Available: <https://mdcplus.fi/blog/top-free-mes-systems-manufacturing-execution/>.
- [25] GE Vernova. “Manufacturing Execution Systems (MES) Software.” Accessed: 2026-01-06. [Online]. Available: <https://www.gevernova.com/software/products/manufacturing-execution-systems>.
- [26] S. Mantravadi, C. Møller, C. LI, and R. Schnyder, “Design choices for next-generation iiot-connected mes/mom: An empirical study on smart factories,” *Robotics and Computer-Integrated Manufacturing*, vol. 73, 2022, Cited by: 59; All Open Access, Green Open Access. DOI: 10.1016/j.rcim.2021.102225. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85112385889&doi=10.1016%2fj.rcim.2021.102225&partnerID=40&md5=1ec90d29d421739a9519f4b224c3db68>
- [27] C. Li, S. Mantravadi, and C. Moller, “Aau open source mes architecture for smart factories-exploiting isa 95,” Cited by: 11, vol. 2020-July, 2020, pp. 369–373. DOI: 10.1109/INDIN45582.2020.9442130. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111133722&doi=10.1109%2fINDIN45582.2020.9442130&partnerID=40&md5=8b7143c4a2409b44cdb16d9491fb65a9>.

- [28] R. Cupek, A. Ziebinski, L. Huczala, and H. Erdogan, “Agent-based manufacturing execution systems for short-series production scheduling,” *Computers in Industry*, vol. 82, pp. 245–258, 2016, Cited by: 87; All Open Access, Hybrid Gold Open Access. DOI: 10.1016/j.compind.2016.07.009. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84995771983&doi=10.1016%2fj.compind.2016.07.009&partnerID=40&md5=7284dc3013423b719a20adc8af0856a6>.
- [29] L. Yue, P. Niu, and Y. Wang, “Guidelines for defining user requirement specifications (urs) of manufacturing execution system (mes) based on isa-95 standard,” 3, Cited by: 11; All Open Access, Gold Open Access, vol. 1168, 2019. DOI: 10.1088/1742-6596/1168/3/032065. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063878848&doi=10.1088%2f1742-6596%2f1168%2f3%2f032065&partnerID=40&md5=0367c9855d4e699fa70aeadc0fd2f541>.
- [30] A. Restivo, *sfs: Shop Floor Simulator*, GitHub repository. [Online]. Available: <https://github.com/arestivo/sfs>.

## Appendix A

# ISA-95 Data Models

### A.1 Process Segment

#### A.1.1 Process Segment

A Process Segment defines a distinct production step within the manufacturing process. It serves as a bridge between the product definition and the physical capabilities of the shop floor. In our implementation, this entity encapsulates the operational logic, ensuring that the scheduling engine can trigger a task without requiring hard-coded routines for every machine type.

Table A.1: Process Segment Attributes [6]

| Attribute                | Description                                                                                                                                  |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| ID                       | A unique identification of a specific material class, within the scope of the information exchanged.                                         |
| Description              | Additional information about the process segment.                                                                                            |
| Operations Type          | Describes the category of the activity.<br>Required attribute.<br>Defined values are: Production, Maintenance, Quality, Inventory, or Mixed. |
| Hierarchy Scope          | Identifies where the exchanged information fits within the role based equipment hierarchy.                                                   |
| Duration                 | Duration of process segment, if known.                                                                                                       |
| Duration Unit of Measure | The units of measure of the duration, if defined.                                                                                            |

#### A.1.2 Equipment Segment Specification

Equipment resources that are needed for a process segment.

Table A.2: Equipment Segment Specification Segment Attributes [6]

| Attribute                | Description                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------|
| Equipment Class          | Identifies the associated equipment class or set of equipment classes of the capability.            |
| Equipment                | Identifies the associated equipment or set of equipment of the capability.                          |
| Description              | Contains additional information and descriptions.                                                   |
| Equipment Use            | Defines the expected use of the equipment class or equipment in the context of the process segment. |
| Quantity                 | Specifies the amount of resources required, if applicable.                                          |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                      |

For equipment segment specifications, we can use equipment class or equipment based on what is necessary for the process segment.

### A.1.3 Equipment Segment Specification Property

Specific properties that are required to perform the process segment for the respective equipment or equipment class, e.g Type Power - electric.

Table A.3: Equipment Segment Specification Property Attributes [6]

| Attribute                | Description                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated equipment property or equipment class property. |
| Description              | Contains additional information and description.                                                  |
| Value                    | The value, set of values, or range of the property.                                               |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                              |
| Quantity                 | Specifies the amount of resources required.                                                       |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                    |

### A.1.4 Physical Asset Segment Specification

Physical asset resources needed for the process segment.

Table A.4: Physical Asset Segment Specification Attributes [6]

| Attribute                | Description                                                                                                   |
|--------------------------|---------------------------------------------------------------------------------------------------------------|
| Physical Asset Class     | A Identifies the associated material class or set of physical asset classes of the capability.                |
| Physical Asset           | Identifies the associated material definition or set of physical assets of the capability.                    |
| Description              | Contains additional information and descriptions.                                                             |
| Physical Asset Use       | Defines the expected use of the physical asset class or physical asset in the context of the process segment. |
| Quantity                 | Specifies the amount of resources required, if applicable.                                                    |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                                |

### A.1.5 Physical Asset Segment Specification Property

Properties that are required to perform the process segment for the respective physical asset or physical asset class, e.g Run clock - 1200 hours.

Table A.5: Physical Asset Segment Specification Property Attributes [6]

| Attribute                | Description                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated material property or material class property. |
| Description              | Contains additional information and description.                                                |
| Value                    | The value, set of values, or range of the property.                                             |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                            |
| Quantity                 | Specifies the amount of resources required.                                                     |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                  |

### A.1.6 Material Segment Specification

Material resources necessary for the process segment.

Table A.6: Material Segment Specification Attributes [6]

| Attribute                | Description                                                                                                                                                                                                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Material Class           | A Identifies the associated material class or set of material classes of the capability.                                                                                                                                                                                  |
| Material Definition      | Identifies the associated material definition or set of material definitions of the capability.                                                                                                                                                                           |
| Description              | Contains additional information and descriptions.                                                                                                                                                                                                                         |
| Assembly Type            | Optional: Defines the type of the assembly.<br>Types:<br>Physical – The components of the assembly are physically connected or in the same area.<br>Logical – The components of the assembly are not necessarily physically connected or in the same area.                |
| Assembly Relationship    | Optional: Defines the type of the relationship.<br>Types:<br>Permanent – An assembly that is not intended to be split during the production process.<br>Transient – A temporary assembly using during production, such as a pallet of different materials or a batch kit. |
| Material Use             | Defines the material use. For production defined values are: Consumable, Material Consumed, and Material Produced.                                                                                                                                                        |
| Quantity                 | Specifies the amount of resources required.                                                                                                                                                                                                                               |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                                                                                                                                                                                            |

For material segment specifications, we can use material class or material definition based on what is necessary for the process segment.

### A.1.7 Material Segment Specification Property

Specific properties that are required to perform the process segment for the respective material definition or material class, e.g Sterilized - true.

Table A.7: Material Segment Specification Property Attributes [6]

| Attribute                | Description                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated material property or material class property. |
| Description              | Contains additional information and description.                                                |
| Value                    | The value, set of values, or range of the property.                                             |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                            |
| Quantity                 | Specifies the amount of resources required.                                                     |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                  |

### A.1.8 Process Segment Parameter

Specific parameters needed to perform the process segment.

Table A.8: Process Segment Parameter Attributes [6]

| Attribute             | Description                                                          |
|-----------------------|----------------------------------------------------------------------|
| ID                    | Identificatio of the process parameter.                              |
| Description           | Contains additional information.                                     |
| Value                 | The value, set of values, or range.                                  |
| Value Unit of Measure | The unit of measure of the associated property value, if applicable. |

### A.1.9 Process Segment Dependency

Process segment dependency defines relationships and dependencies between different process segment. Understanding them is very important for effective planning, scheduling and execution of production.

Table A.9: Process Segment Dependency Attributes [6]

| Attribute         | Description                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------|
| ID                | The identification of the unique instance of the process segment dependency.                   |
| Description       | Contains additional information and descriptions of the process segment dependency definition. |
| Dependency Type   | Defines the execution dependency constraints of one segment by another segment.                |
| Dependency Factor | Factor used by dependency.                                                                     |
| Unit of Measure   | The units of measure of the dependency factor, if defined.                                     |

## A.2 Operations Definition

### A.2.1 Operations Definition

This entity represents the master template for a specific product manufacturing workflow. It organizes various Process Segments into a coherent, executable sequence or operations. By centralizing the product definition within this entity, our MES allows for rapid product changeovers.

The system can switch between production flows simply by referencing a different definition file, rather than modifying the underlying control logic.

Table A.10: Operations Definition Attributes [6]

| Attribute             | Description                                                                                                                               |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| ID                    | Uniquely identifies the operations definition.                                                                                            |
| Version               | An identification of the version of the Operations definition.                                                                            |
| Description           | Contains additional information and descriptions of the Operations definition.                                                            |
| Operations type       | Describes the category of operation.<br>Required attribute.<br>Defined values are: Production, Maintenance, Quality, Inventory, or Mixed. |
| Hierarchy Scope       | Identifies where the exchanged information fits within the role based equipment hierarchy.                                                |
| Bill of Materials ID  | Identification of the external Bill Of Material associated with this Operation Definition.                                                |
| Work of Definition ID | Identification of the external Work Definition associated with this Operations Definition.                                                |
| Bill of Resources ID  | Identification of the external Bill Of Resource associated with this Operation Definition.                                                |

### A.2.2 Operations Segment

The information needed to quantify a segment for a specific operation. An operations segment identifies, references, or corresponds to a process segment.

Table A.11: Operations Segment Attributes [6]

| Attribute                | Description                                                                                                                                 |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| ID                       | A unique identification of a specific segment within the scope of the information exchanged.                                                |
| Description              | Contains additional information and descriptions of the Operations definition.                                                              |
| Hierarchy Scope          | Identifies where the exchanged information fits within the role based equipment hierarchy.                                                  |
| Duration                 | Duration of segment, if known.                                                                                                              |
| Duration Unit of Measure | The units of measure of the duration, if defined.                                                                                           |
| Process Segment          | Identifies the associated Process segments. There may be multiple alternate process segments that could be used for the operations segment. |
| Operations type          | Describes the category of operation.<br>Required attribute.<br>Defined values are: Production, Maintenance, Quality, Inventory, or Mixed.   |
| Work of Definition ID    | Identification of the external Work Definition associated with this Operations Segment.                                                     |

### A.2.3 Parameter Specification

An operations segment may have an associated set of parameters specifications. The parameter specification contains the names and types of the values that may be sent to the Level 3 systems to parameterize the operation.

Table A.12: Parameter Specification Attributes [6]

| Attribute             | Description                                              |
|-----------------------|----------------------------------------------------------|
| ID                    | Identification of the parameter for a specific segment.  |
| Description           | Contains additional information of the parameter.        |
| Value                 | The value, set of values, or range of acceptable values. |
| Value Unit of Measure | Unit of measure of the values, if applicable.            |

### A.2.4 Equipment Specification

An identification to an equipment capability. An equipment specification may specify either an equipment class or a piece of equipment. This identifies the specific equipment capability that is associated with the segment.

Table A.13: Equipment Specification Attributes [6]

| Attribute                | Description                                                                              |
|--------------------------|------------------------------------------------------------------------------------------|
| Equipment Class          | Identifies the associated equipment class or set of equipment classes of the capability. |
| Equipment                | Identifies the associated equipment or set of equipment of the capability.               |
| Description              | Contains additional information and descriptions.                                        |
| Equipment Use            | Defines the expected use of the equipment class or equipment.                            |
| Quantity                 | Specifies the amount of resources required, if applicable.                               |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                           |

For equipment specifications, we can use equipment class or equipment based on what is necessary for the process segment.

### A.2.5 Equipment Specification Property

Specific properties that are required to perform the operation segment for the respective equipment or equipment class, e.g drill size - 5mm.

Table A.14: Equipment Specification Property Attributes [6]

| Attribute                | Description                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated equipment property or equipment class property. |
| Description              | Contains additional information and description.                                                  |
| Value                    | The value, set of values, or range of the property.                                               |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                              |
| Quantity                 | Specifies the amount of resources required.                                                       |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                    |

### A.2.6 Physical Asset Specification

An identification to a physical asset capability. A physical asset specification may specify either a physical asset or a physical asset class. This identifies the specific physical asset capability that is

associated with the segment.

Table A.15: Physical Asset Specification Attributes [6]

| Attribute                | Description                                                                                    |
|--------------------------|------------------------------------------------------------------------------------------------|
| Physical Asset Class     | A Identifies the associated material class or set of physical asset classes of the capability. |
| Physical Asset           | Identifies the associated material definition or set of physical assets of the capability.     |
| Description              | Contains additional information and descriptions.                                              |
| Physical Asset Use       | Defines the expected use of the physical asset class or physical asset.                        |
| Quantity                 | Specifies the amount of resources required, if applicable.                                     |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                 |

### A.2.7 Physical Asset Specification Property

Properties that are required to perform the operation segment for the respective physical asset or physical asset class.

Table A.16: Physical Asset Specification Property Attributes [6]

| Attribute                | Description                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated material property or material class property. |
| Description              | Contains additional information and description.                                                |
| Value                    | The value, set of values, or range of the property.                                             |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                            |
| Quantity                 | Specifies the amount of resources required.                                                     |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                  |

### A.2.8 Material Specification

An identification to a material capability. A material specification specifies a material definition or a material class. This identifies the specific material specification that is associated with the identified product segment.

Table A.17: Material Specification Attributes [6]

| Attribute                | Description                                                                                                                                                                                                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Material Class           | A Identifies the associated material class or set of material classes of the capability.                                                                                                                                                                                  |
| Material Definition      | Identifies the associated material definition or set of material definitions of the capability.                                                                                                                                                                           |
| Description              | Contains additional information and descriptions.                                                                                                                                                                                                                         |
| Material Use             | Defines the material use. For production defined values are: Consumable, Material Consumed, and Material Produced.                                                                                                                                                        |
| Quantity                 | Specifies the amount of resources required.                                                                                                                                                                                                                               |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                                                                                                                                                                                            |
| Assembly Type            | Optional: Defines the type of the assembly.<br>Types:<br>Physical – The components of the assembly are physically connected or in the same area.<br>Logical – The components of the assembly are not necessarily physically connected or in the same area.                |
| Assembly Relationship    | Optional: Defines the type of the relationship.<br>Types:<br>Permanent – An assembly that is not intended to be split during the production process.<br>Transient – A temporary assembly using during production, such as a pallet of different materials or a batch kit. |

For material specifications, we can use material class or material definition based on what is necessary for the process segment.

### A.2.9 Material Specification Property

Specific properties that are required to perform the operation segment for the respective material definition or material class.

Table A.18: Material Specification Property Attributes [6]

| Attribute                | Description                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated material property or material class property. |
| Description              | Contains additional information and description.                                                |
| Value                    | The value, set of values, or range of the property.                                             |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                            |
| Quantity                 | Specifies the amount of resources required.                                                     |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                  |

### A.2.10 Operations Segment Dependency

The operations segment dependencies can be used to describe dependencies that are operation or product specific.

Table A.19: Operations Segment Dependency Attributes [6]

| Attribute         | Description                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------|
| ID                | The identification of the unique instance of the operation segment dependency.                    |
| Description       | Contains additional information and descriptions of the operations segment dependency definition. |
| Dependency Type   | Defines the execution dependency constraints of one segment by another segment.                   |
| Dependency Factor | Factor used by dependency.                                                                        |
| Unit of Measure   | The units of measure of the dependency factor, if defined.                                        |

## A.3 Operations Schedule

### A.3.1 Operations Schedule

This entity acts as the container for the overall production plan for a defined time frame. Links the external demand (customer orders) with the internal capacity (shop-floor availability). In this MES implementation, the schedule is not static. It is updated dynamically as the scheduling algorithm processes new requests and reacts to real-time machine status updates.

Table A.20: Operations Schedule Attributes [6]

| Attribute       | Description                                                                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID              | A unique identification of the operations schedule and could include version and revision identification.                                                                                                               |
| Description     | Contains additional information and descriptions of the Operations schedule.                                                                                                                                            |
| Operations type | Describes the category of operation.<br>Required attribute.<br>Defined values are: Production, Maintenance, Quality, Inventory, or Mixed.                                                                               |
| Start Time      | The starting time for the associated operations schedule, if applicable.                                                                                                                                                |
| End Time        | The ending time for the associated operations schedule, if applicable.                                                                                                                                                  |
| Published Date  | The date and time on which the operations schedule was published or generated.                                                                                                                                          |
| Hierarchy Scope | Identifies where the exchanged information fits within the role based equipment hierarchy.                                                                                                                              |
| Schedule State  | Indicates the state of the operations schedule.<br>Defined values are: Forecast and Released.<br>Forecast - The requirements have not been released for use.<br>Released - The requirements have been released for use. |

### A.3.2 Operations Request

An Operations Request represents a specific production order or task within the schedule. Translates customer-level requirements into actionable data for the MES engine. Each request references an Operations Definition and defines the target quantities, triggering the allocation of the required Material Lots and Equipment resources.

Table A.21: Operations Request Attributes [6]

| Attribute               | Description                                                                                                                                                                                                            |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID                      | A unique identification of the operations request.                                                                                                                                                                     |
| Description             | Contains additional information and descriptions of the Operations request.                                                                                                                                            |
| Operations type         | Describes the category of operation.<br>Required attribute.<br>Defined values are: Production, Maintenance, Quality, Inventory, or Mixed.                                                                              |
| Start Time              | The starting time for the associated operations schedule, if applicable.                                                                                                                                               |
| End Time                | The ending time for the associated operations schedule, if applicable.                                                                                                                                                 |
| Priority                | The priority of the request, if applicable.                                                                                                                                                                            |
| Hierarchy Scope         | Identifies where the exchanged information fits within the role based equipment hierarchy.                                                                                                                             |
| Operation Definition ID | Identifies the associated Operations definition to be used, if applicable.                                                                                                                                             |
| Request State           | Indicates the state of the operations request.<br>Defined values are: Forecast and Released.<br>Forecast - The requirements have not been released for use.<br>Released - The requirements have been released for use. |

### A.3.3 Segment Requirement

Each segment requirement will correspond to an identified operations segment or process segment. The segment requirement identifies or references the segment capability to which the associated personnel, equipment, materials, and segment parameters correspond.

Table A.22: Segment Requirement Attributes [6]

| Attribute                | Description                                                                                                                               |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| ID                       | A unique identification of the segment requirement within the scope of an operations request.                                             |
| Description              | Contains additional information and descriptions of the segment requirement.                                                              |
| Operations type          | Describes the category of operation.<br>Required attribute.<br>Defined values are: Production, Maintenance, Quality, Inventory, or Mixed. |
| Segment                  | An identification of the segment associated with the segment requirement, if applicable.                                                  |
| Earliest Start Time      | The expected earliest start time of this segment requirement, if applicable.                                                              |
| Latest End Time          | The expected latest end time of this segment requirement, if applicable.                                                                  |
| Duration                 | The expected duration of this segment requirement, if applicable.                                                                         |
| Duration Unit of Measure | The unit of measure of the duration, if applicable.                                                                                       |
| Hierarchy Scope          | Identifies where the exchanged information fits within the role based equipment hierarchy.                                                |
| Operation Definition ID  | Identifies the associated Operations definition to be used, if applicable.                                                                |

#### A.3.4 Segment Parameter

Information contained in the enterprise system that is required by the operation system for correct manufacturing.

A segment parameter should include a set of limits that apply to any change to the value, such as quality limits and safety limits.

Table A.23: Segment Parameter Attributes [6]

| Attribute             | Description                                                         |
|-----------------------|---------------------------------------------------------------------|
| ID                    | Identification of the segment parameter.                            |
| Description           | Contains additional information of the segment parameter.           |
| Value                 | The value, set of values, or range of acceptable values.            |
| Value Unit of Measure | The engineering units in which the value is defined, if applicable. |

### A.3.5 Equipment Requirement

The identification of the number, type, duration, and scheduling of specific equipment and equipment classifications needed to support the current operations request.

Table A.24: Equipment Requirement Attributes [6]

| Attribute                | Description                                                                                                                                              |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Equipment Class          | Identifies the associated equipment class or set of equipment classes of the requirement for a specific segment requirement.                             |
| Equipment                | Identifies the associated equipment set of equipment of the requirement for a specific segment requirement.                                              |
| Description              | Contains additional information and descriptions.                                                                                                        |
| Equipment Use            | Defines the expected use of the equipment class or equipment.                                                                                            |
| Quantity                 | Specifies the amount of resources required, if applicable.                                                                                               |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                                                                           |
| Equipment Level          | A definition of the level of the associated element of the equipment model. For example: enterprise, site, area, unit, Equipment module, Control Module. |

### A.3.6 Equipment Requirement Property

Specific properties that are required to perform the segment requirement for the respective equipment or equipment class.

Table A.25: Equipment Requirement Property Attributes [6]

| Attribute                | Description                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated equipment property or equipment class property. |
| Description              | Contains additional information and description.                                                  |
| Value                    | The value, set of values, or range of the property.                                               |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                              |
| Quantity                 | Specifies the amount of resources required.                                                       |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                    |

### A.3.7 Physical Asset Requirement

The identification of the number, type, duration, and scheduling of specific physical assets and physical asset class constraints needed to support the current operations request.

Table A.26: Physical Asset Requirement Attributes [6]

| Attribute                | Description                                                                                                                           |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Physical Asset Class     | Identifies the associated physical asset model or set of physical asset models of the requirement for a specific segment requirement. |
| Physical Asset           | Identifies the associated physical asset or set of physical assets of the requirement for a specific segment requirement.             |
| Description              | Contains additional information and descriptions.                                                                                     |
| Physical Asset Use       | Defines the expected use of the physical asset class or physical asset.                                                               |
| Quantity                 | Specifies the amount of resources required, if applicable.                                                                            |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                                                        |
| Physical Asset Level     | A level definition for the associated element in the hierarchy of the physical asset model.                                           |

### A.3.8 Physical Asset Requirement Property

Specific properties that are required to perform the segment requirement for the respective physical asset or physical asset class.

Table A.27: Physical Asset Specification Property Attributes [6]

| Attribute                | Description                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated material property or material class property. |
| Description              | Contains additional information and description.                                                |
| Value                    | The value, set of values, or range of the property.                                             |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                            |
| Quantity                 | Specifies the amount of resources required.                                                     |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                  |

### A.3.9 Material Requirement

An identification of a material that is expected to be used in the operations request. Material requirements contain definitions of materials that may be consumed, produced, replaced, sampled, or otherwise used in manufacturing.

Table A.28: Material Requirement Attributes [6]

| Attribute                | Description                                                                                                                                                                                                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Material Class           | Identifies the associated material class or set of material classes of the requirement for a specific segment requirement.                                                                                                                                                |
| Material Definition      | Identifies the associated material definition or set of material definitions of the requirement for a specific segment requirement.                                                                                                                                       |
| Material Lot             | Identifies the associated material lot, or set of material lots of the requirement for a specific segment requirement.                                                                                                                                                    |
| Material Sublot          | Identifies the associated material subplot, or set of material sublots of the requirement for a specific segment requirement.                                                                                                                                             |
| Description              | Contains additional information and descriptions.                                                                                                                                                                                                                         |
| Material Use             | Defines the use of the material.                                                                                                                                                                                                                                          |
| Storage Location         | Identifies the proposed location of the material, if applicable.                                                                                                                                                                                                          |
| Quantity                 | Specifies the amount of resources required.                                                                                                                                                                                                                               |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                                                                                                                                                                                            |
| Assembly Type            | Optional: Defines the type of the assembly.<br>Types:<br>Physical – The components of the assembly are physically connected or in the same area.<br>Logical – The components of the assembly are not necessarily physically connected or in the same area.                |
| Assembly Relationship    | Optional: Defines the type of the relationship.<br>Types:<br>Permanent – An assembly that is not intended to be split during the production process.<br>Transient – A temporary assembly using during production, such as a pallet of different materials or a batch kit. |

For material specifications, we can use material class, material definition, material lot or material subplot based on what is necessary for the segment requirement.

### A.3.10 Material Requirement Property

Specific properties that are required to perform the segment requirement for the respective material definition or material class.

Table A.29: Material Requirement Property Attributes [6]

| Attribute                | Description                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------|
| ID                       | An identification of a property of the associated material property or material class property. |
| Description              | Contains additional information and description.                                                |
| Value                    | The value, set of values, or range of the property.                                             |
| Value Unit of Measure    | The unit of measure of the associated property value, if applicable.                            |
| Quantity                 | Specifies the amount of resources required.                                                     |
| Quantity Unit of Measure | The unit of measure of the associated quantity, if applicable.                                  |

## Appendix B

# Database Table Structure

Tables structure of the implemented database based on the ISA-95 standard information models depicted in Chapter 4.

### B.1 Equipment

Based on the Role Based Equipment Model, Chapter 4, Figure 4.1.

This group of tables represents the logical and functional perspective of manufacturing resources, focusing on the roles machines play within the production lifecycle rather than their physical identity.

- `equipment` and `equipment_class`: These entities store individual logical equipment definitions and their broader categories (e.g., "Assembly Robot" or "Milling Station"). Grouping assets into logical classes enables the system to enforce uniform operational behaviors across similar machine types.
- `equipment_property` and `equipment_class_property`: These tables implement a dynamic, key-value attribute schema. This pattern allows the system to append custom, run-time variables (such as nominal velocities or maximum capacities) to specific machines or classes.
- `equipment_made_up_of`: This table establishes the functional hierarchy of the production facility using a recursive parent-child relationship. It allows the MES to structurally define that a particular logical machine (child) belongs to a specific Work Cell or Production Line (parent).

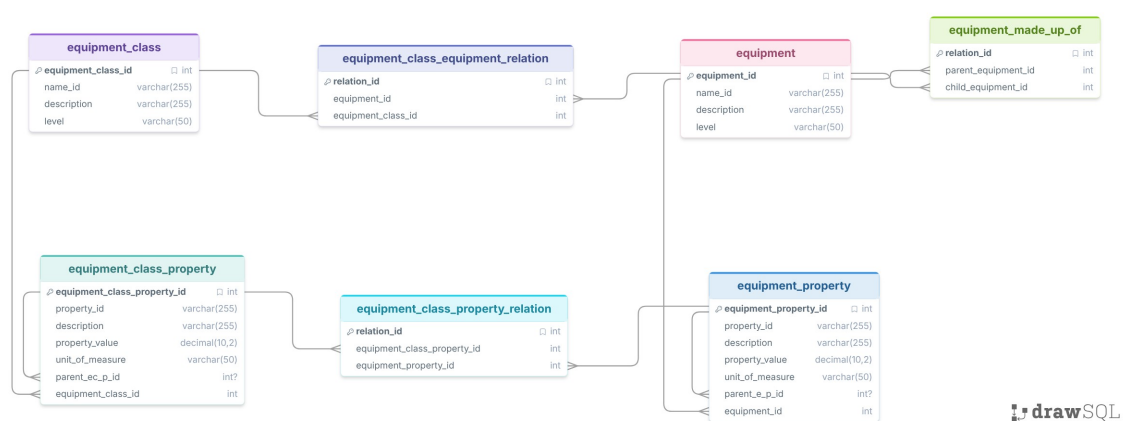


Figure B.1: Equipment Database Table Structure

## B.2 Physical Asset

Based on the Physical Asset Model, Chapter 4, Figure 4.2.

While the Equipment Model focuses on functional roles, the Physical Asset Model tracks the actual, serialized hardware instances deployed on the shop floor.

- `physical_asset` and `physical_asset_class`: These tables register the physical hardware instances, capturing asset-specific data such as manufacturing serial numbers, fixed asset IDs, vendor details, and deployment locations.
- Asset-to-Equipment Mapping (`equipment_id`): The `physical_asset` table includes a direct foreign key reference to the logical `equipment_id`. This mapping enables the MES to track precisely which piece of physical hardware is fulfilling a specific logical production role at any given moment.
- `physical_asset_property` and `physical_asset_made_up_of`: These entities mirror the key-value property pattern and recursive hierarchy of the equipment model, providing an extensible way to catalog hardware sub-components and their exact maintenance or engineering specifications.

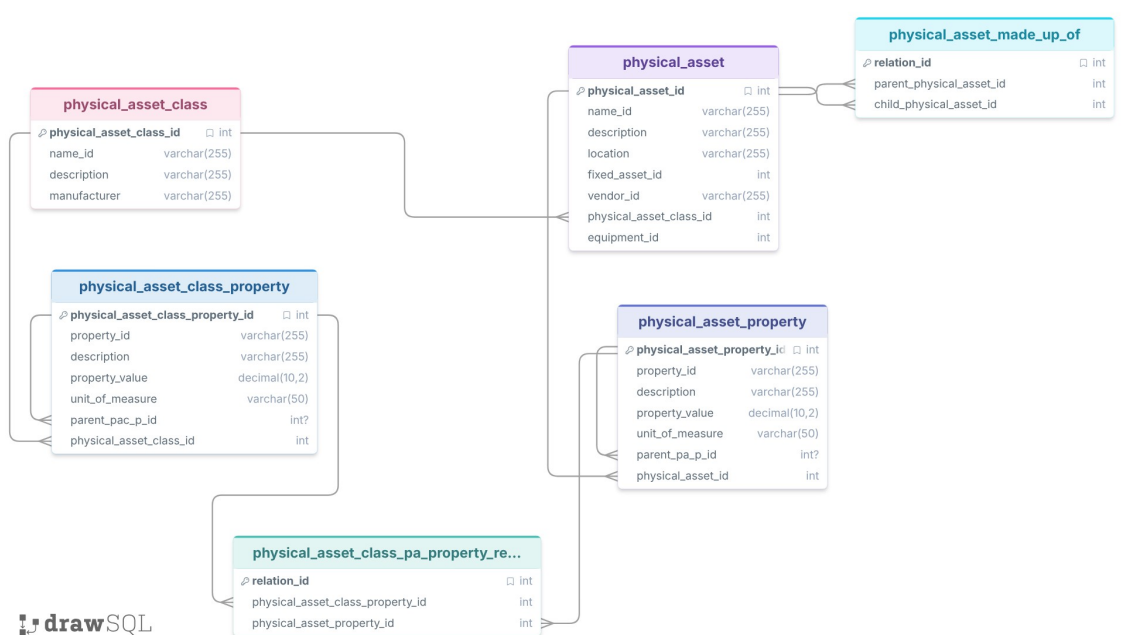


Figure B.2: Physical Asset Database Table Structure

### B.3 Material

Based on the Material Model, Chapter 4, Figure 4.4.

This domain handles the complete inventory lifecycle, tracking raw elements, intermediate sub-assemblies, and final products, while supporting Bill of Materials (BOM) definitions.

- `material_definition` and `material_class`: These tables catalog the abstract material types across the factory (e.g., wood panels, metallic frames, finished desks) alongside their assembly types and relationships.
- `material_lot`: Crucial for execution tracking, this entity represents concrete physical batches of material currently sitting in inventory. It maintains live records of quantities, units of measure (`unit_of_measure`), and storage designations (`storage_location`).
- Association Tables (`assemble_material_definition`, `assemble_lot`): These tables model recursive tree-like structures for materials to handle BOM explosions. They map dependencies both at a conceptual level (defining which material definitions compose a product recipe) and at a physical execution level (tracking which specific physical batches/lots were consumed to yield a new output lot), guaranteeing end-to-end material traceability (genealogy).



Figure B.3: Material Database Table Structure

## B.4 Process Segment

Based on the Process Segment Model, Chapter 4, Figure 4.5.

This domain serves as the operational blueprint that connects materials, physical assets, and equipment to execute a specific manufacturing job.

- **process\_segment**: Maps the discrete logical steps of the production process. It defines the operational typology, the hierarchical scope in which it executes, and the nominal durations expected for the task. This model serves as the "recipe template" that incoming ERP production schedule entries instantiate to generate concrete operations requests.

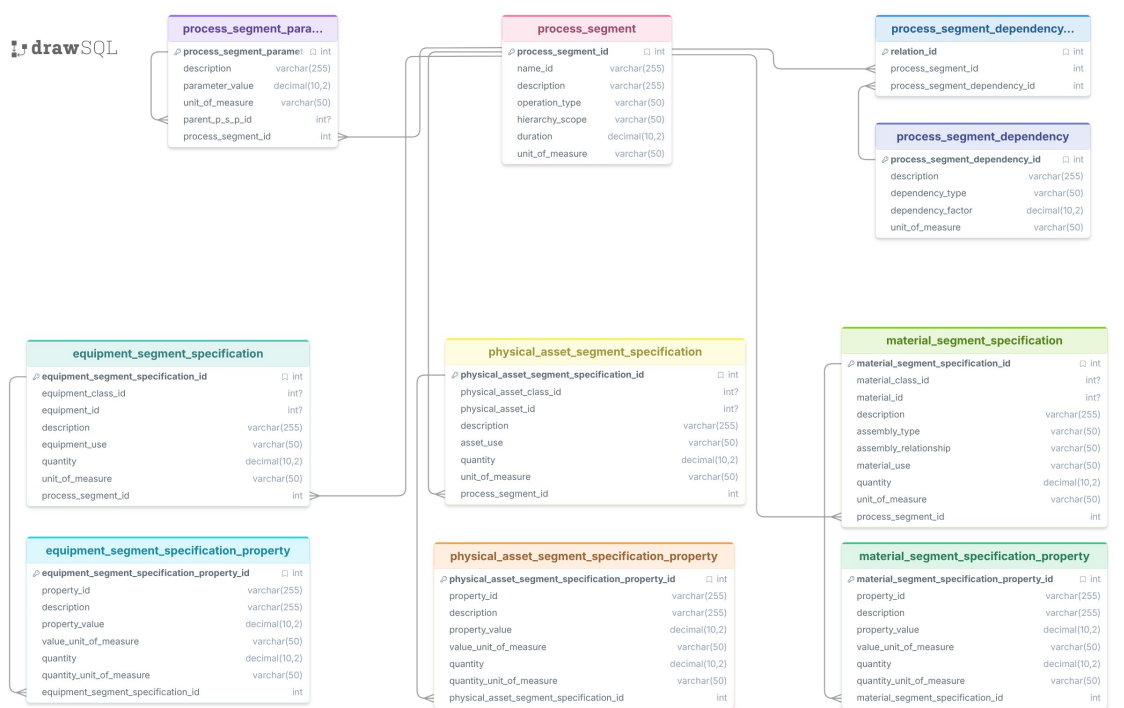


Figure B.4: Process Segment Database Table Structure

## B.5 Operations Definition

Based on the Operations Definition Model, Chapter 4, Figure 4.6.

While the Process Segment domain defines individual, isolated manufacturing steps, the Operations Definition domain acts as the comprehensive "Master Recipe" or routing matrix. It defines exactly how a specific product is made from start to finish.

- **operations\_definition**: This table represents the global blueprint for producing a specific item (e.g., the complete recipe for a "Wooden Desk"). It encapsulates the entire manufacturing routing required to transform raw materials into finished goods.
- **operations\_segment**: This entity breaks down the overarching operations\_definition into an ordered sequence of logical steps. It creates a relational bridge to the process\_segment table, sequencing individual tasks (e.g., "Cut Wood," then "Assemble Legs") and establishing the structural dependencies required to execute the full product recipe.
- **Role in the Architecture**: This model serves as the static reference data. When the ERP sends a production order for a specific product, the MES queries this domain to understand the exact sequence of segments, material ratios, and equipment classes required to fulfill the order.

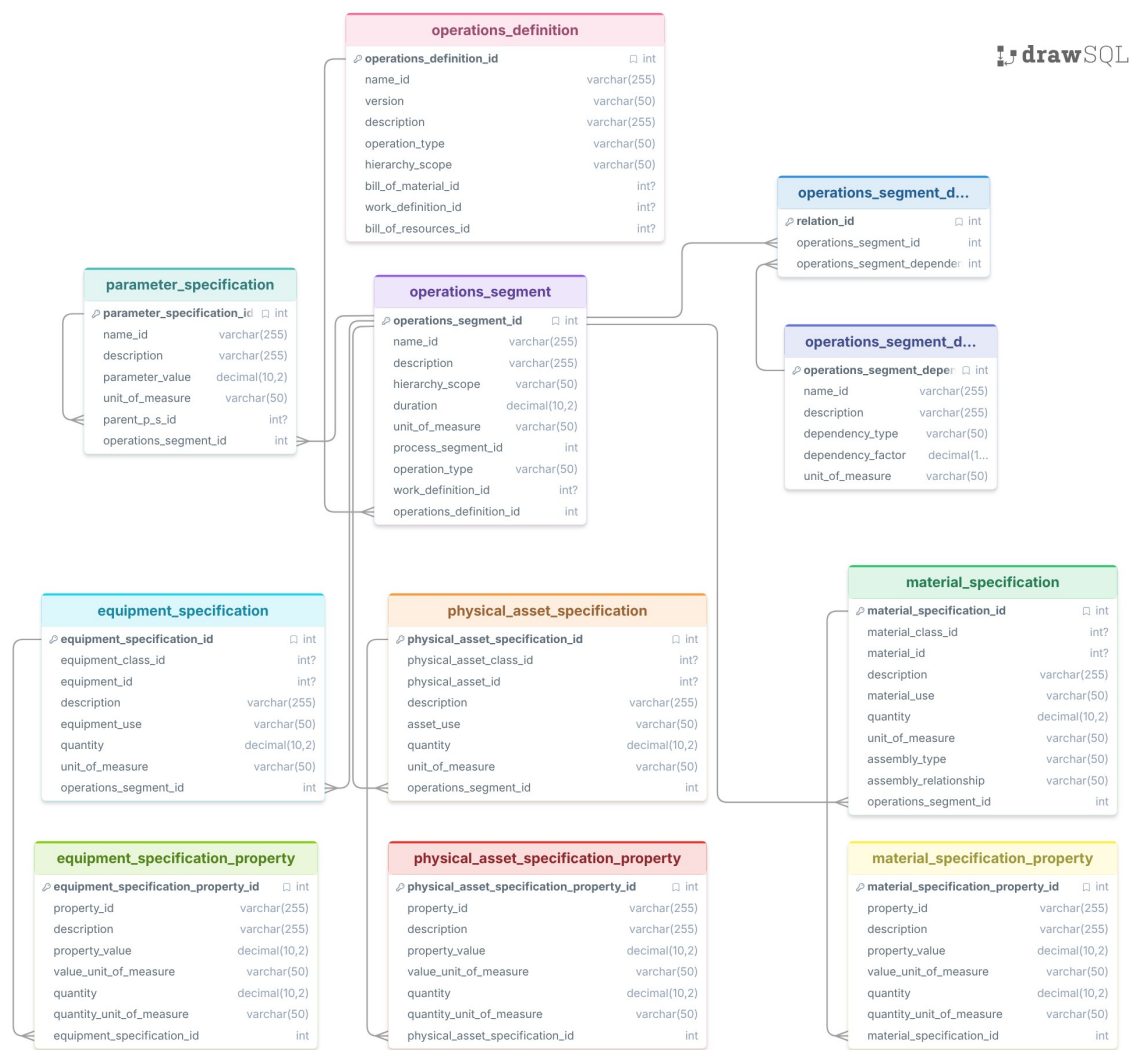


Figure B.5: Operations Definition Database Table Structure

## B.6 Operations Schedule

Based on the Operations Schedule Model, Chapter 4, Figure 4.7.

This is the dynamic, transactional core of the MES database. While all previous models represent static structural definitions or live physical states, the Operations Schedule model represents the active execution timeline. It answers the questions: What is being produced right now, by when, and in what quantity?

- **operations\_schedule**: Represents the macro-level production plan (often generated from an incoming ERP batch). It defines the overarching time horizon (*start\_time*, *end\_time*) and global state of the scheduling block.
- **operations\_request**: Represents the individual production orders nested within the global schedule. Each request targets a specific final product, quantity, and due date.

- **segment\_requirement**: This is the most granular operational table. The Python Scheduler module dynamically generates these records by instantiating the static `operations_segment` templates. Each requirement represents a specific operation that must be dispatched to the shop floor.
- **Resource Requirement Tables** (`material_requirement`, `equipment_requirement`, `physical_asset_requirement`): Linked directly to the `segment_requirement`, these tables lock in the explicit execution constraints. They declare exactly which physical lots (`material_lot_id`) will be consumed, which machine classes are authorized, and the target quantities for that specific runtime execution. This rigid structure provides the Dispatcher module with a completely resolved operational matrix, preventing unsequenced or unresourced commands from reaching the PLC layer.

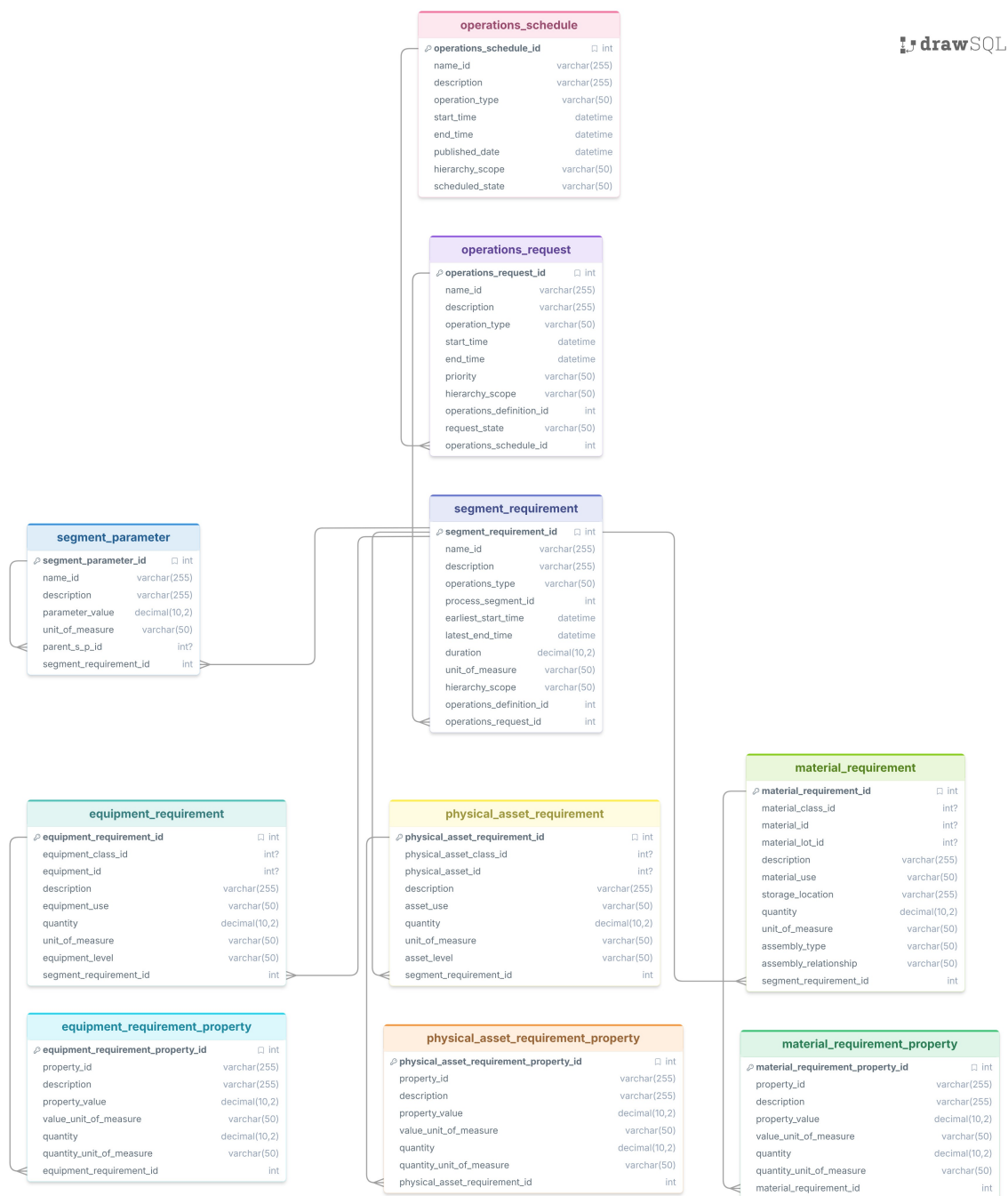


Figure B.6: Operations Schedule Database Table Structure

# Appendix C

## User Interface

### C.1 Navigation Flow

This section describes the structure, navigation flow, and layout of the developed user interface (UI). It outlines the primary dashboards accessible from the central navigation hub and details the hierarchical workflow that allows users to seamlessly manage, visualize, and drill down into the system's equipment, physical assets, materials, process segments, operations definition, and operations schedule.

#### C.1.1 Navigation

The navigation bar (navbar) serves as the central navigation hub of the user interface, enabling seamless transitions between all main sections of the system. From this global component, users can access the primary dashboards for: Equipments, Physical Assets, Materials, Process Segments, Operations Definitions, and the Operations Schedule. The general UI flow follows a hierarchical drill-down pattern, where users start at a high-level list and navigate deeper into nested details and dependencies by interacting with specific elements.

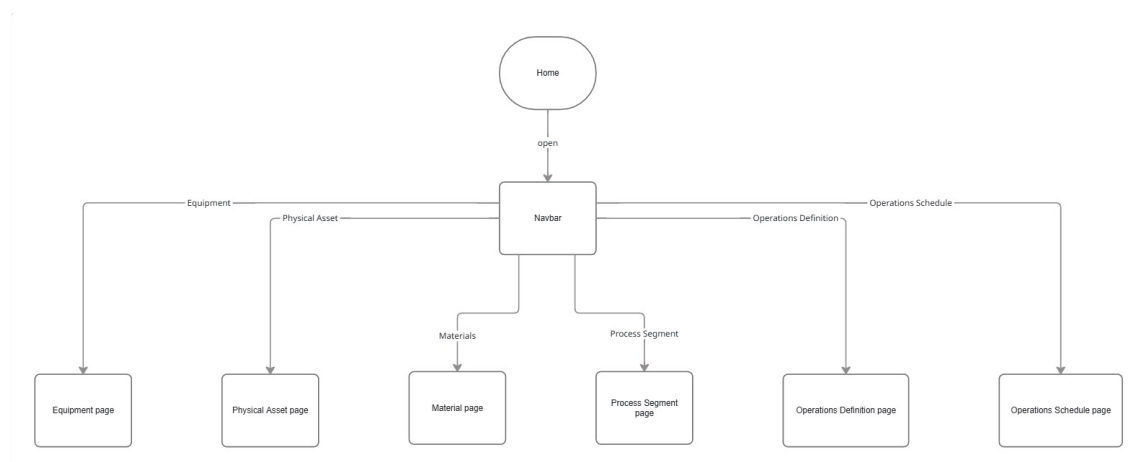


Figure C.1: Navbar Pages Diagram

**C.1.2 Equipments Tab**

The main page of the Equipments tab displays tabular lists of the implemented equipment entities, specifically Equipment Classes and individual Equipment units. By clicking on any item within these lists, the user is redirected to a dedicated details page. This view presents comprehensive information regarding the selected element, including its specific properties, operational parameters, and underlying dependencies.

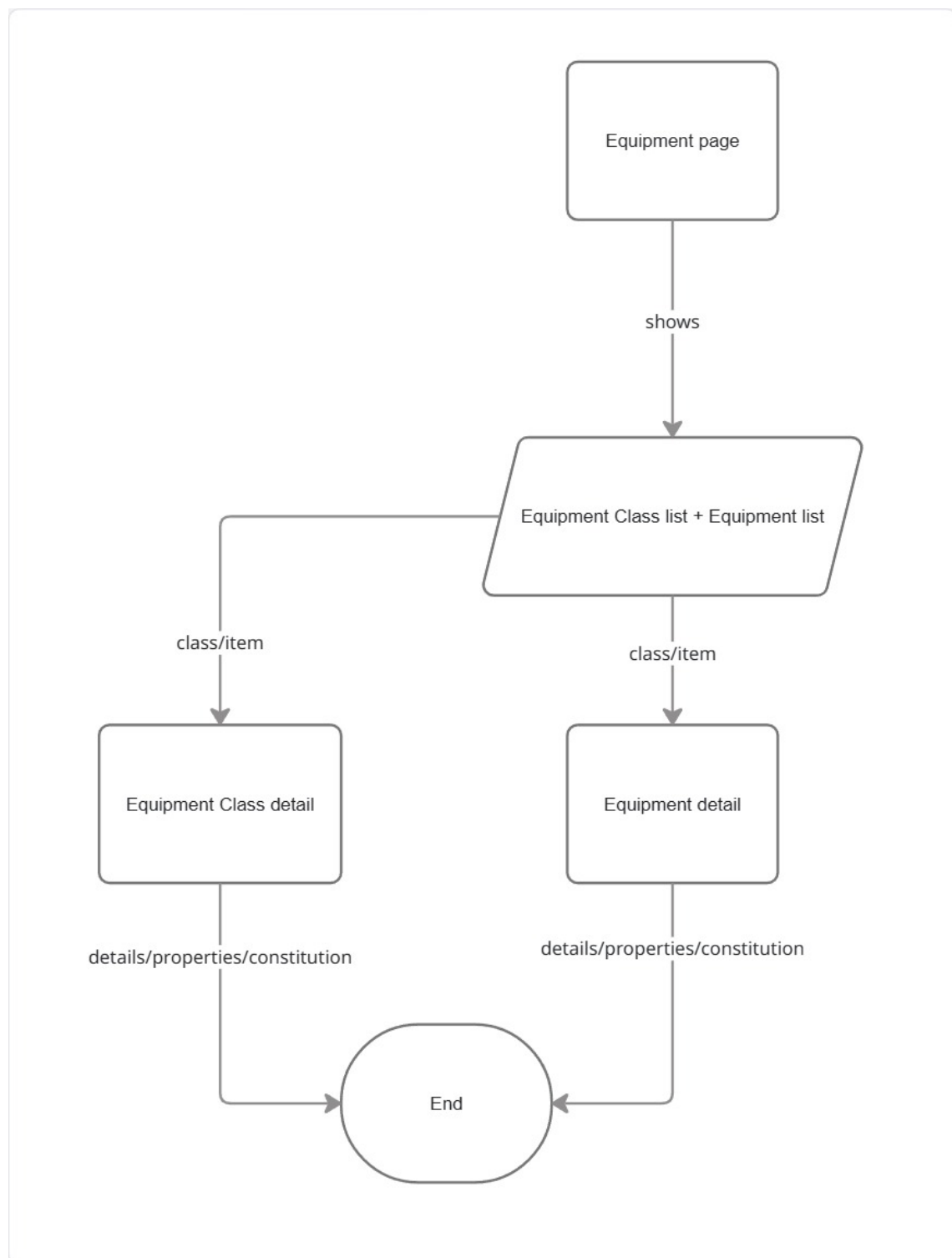


Figure C.2: Equipment Pages Diagram

### C.1.3 Physical Asset Tab

Similar to the equipment view, the Physical Asset main dashboard features lists for both Physical Asset Classes and individual physical asset elements. Selecting any item from these lists navigates

the user to its respective details page, which showcases its structural characteristics, properties, and configuration settings.

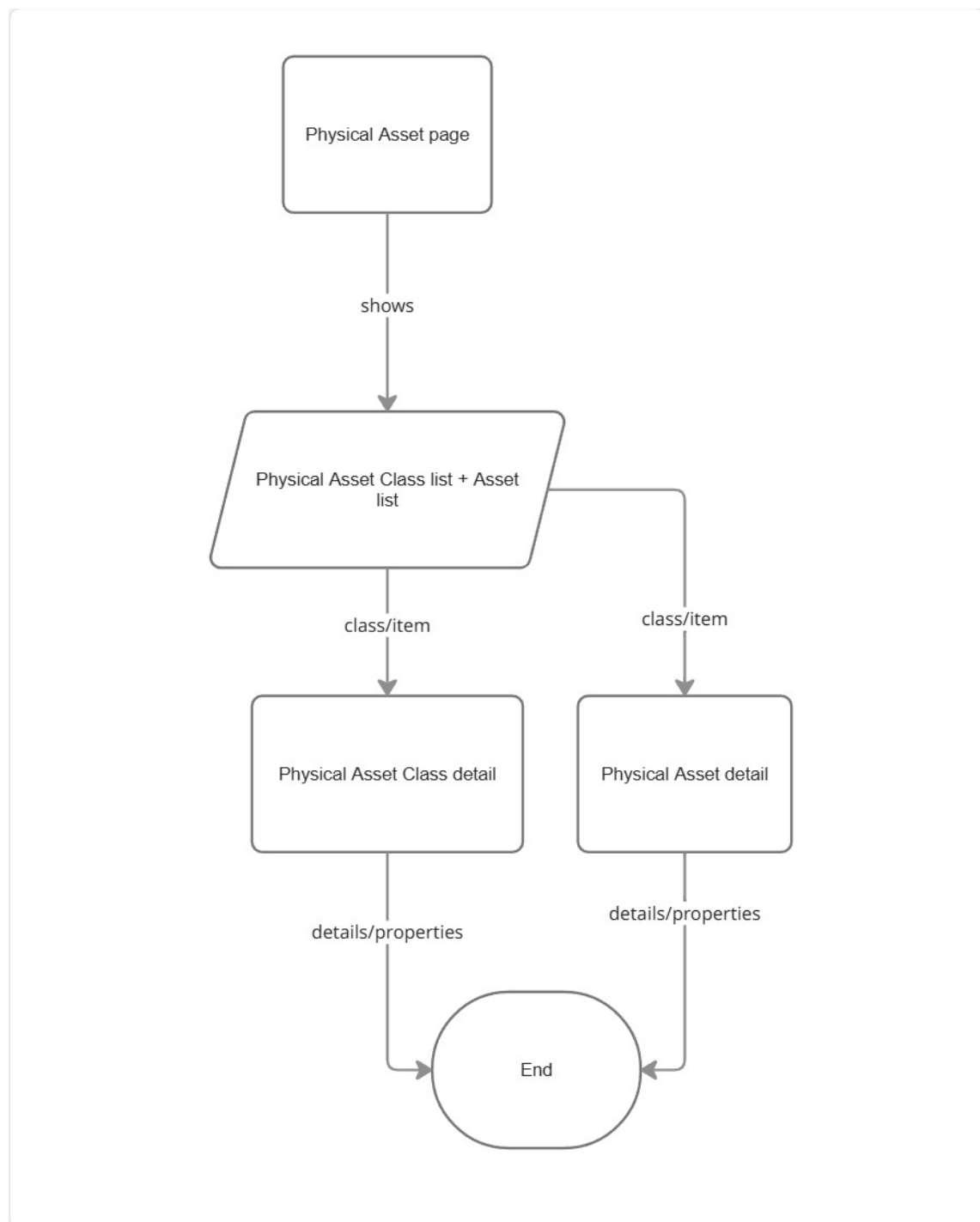


Figure C.3: Physical Asset Pages Diagram

### C.1.4 Material Tab

The Material section's dashboard provides three distinct lists: Material Classes, Material Definitions, and Material Lots. Users can drill down into the material hierarchy by clicking on any element within these lists. This action opens a dedicated view displaying the selected item's core information, specifications, and properties.

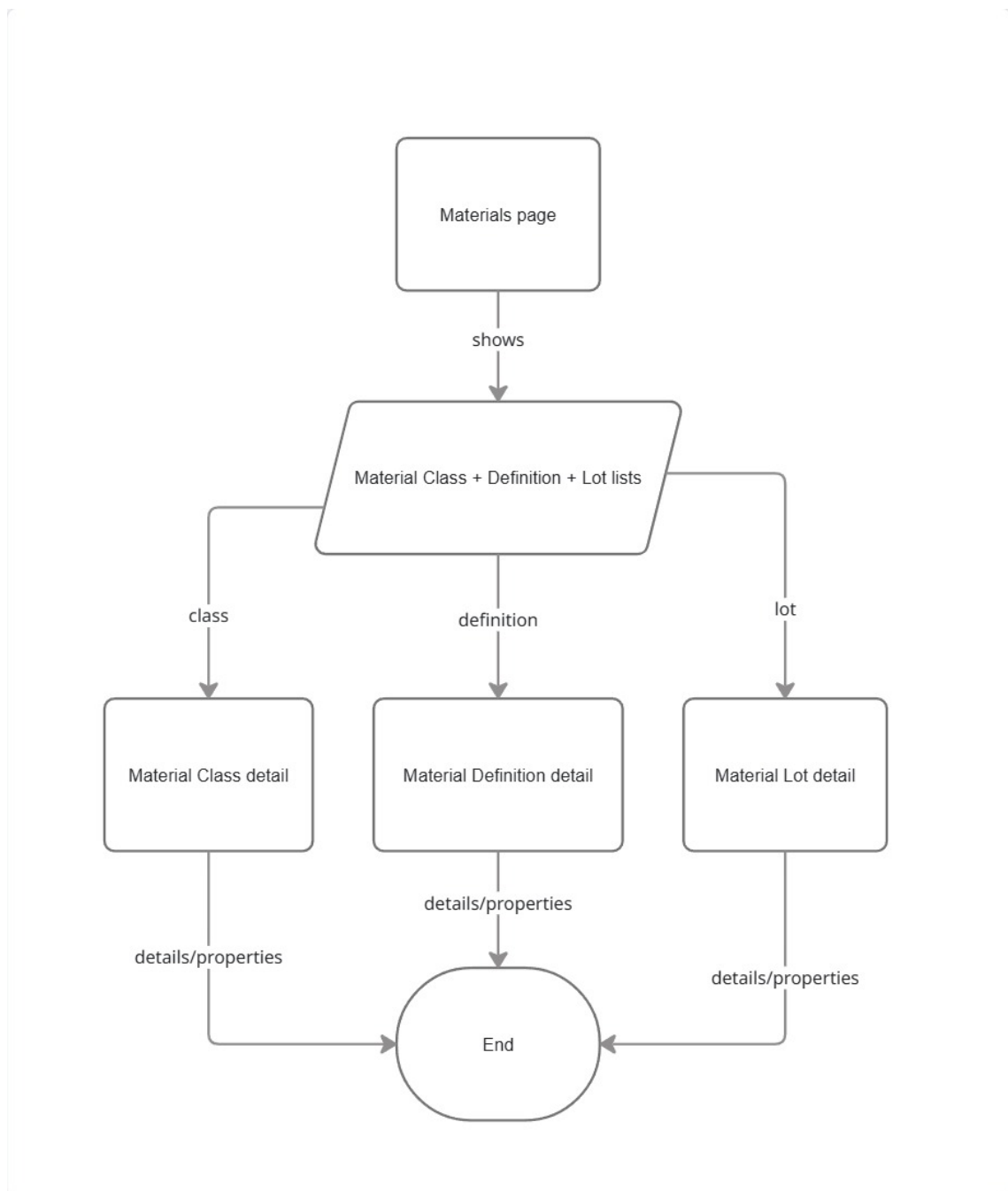


Figure C.4: Material Pages Diagram

### **C.1.5 Process Segment Tab**

This section displays a comprehensive list of all Process Segments configured in the system. Clicking on any segment redirects the user to its details page, which outlines its overall constitution. Within this view, the interface displays nested lists for Equipment Segment Specifications, Physical Asset Segment Specifications, and Material Segment Specifications. Selecting an item from any of these sub-lists opens its specific detail page, exposing its technical properties and parameters.

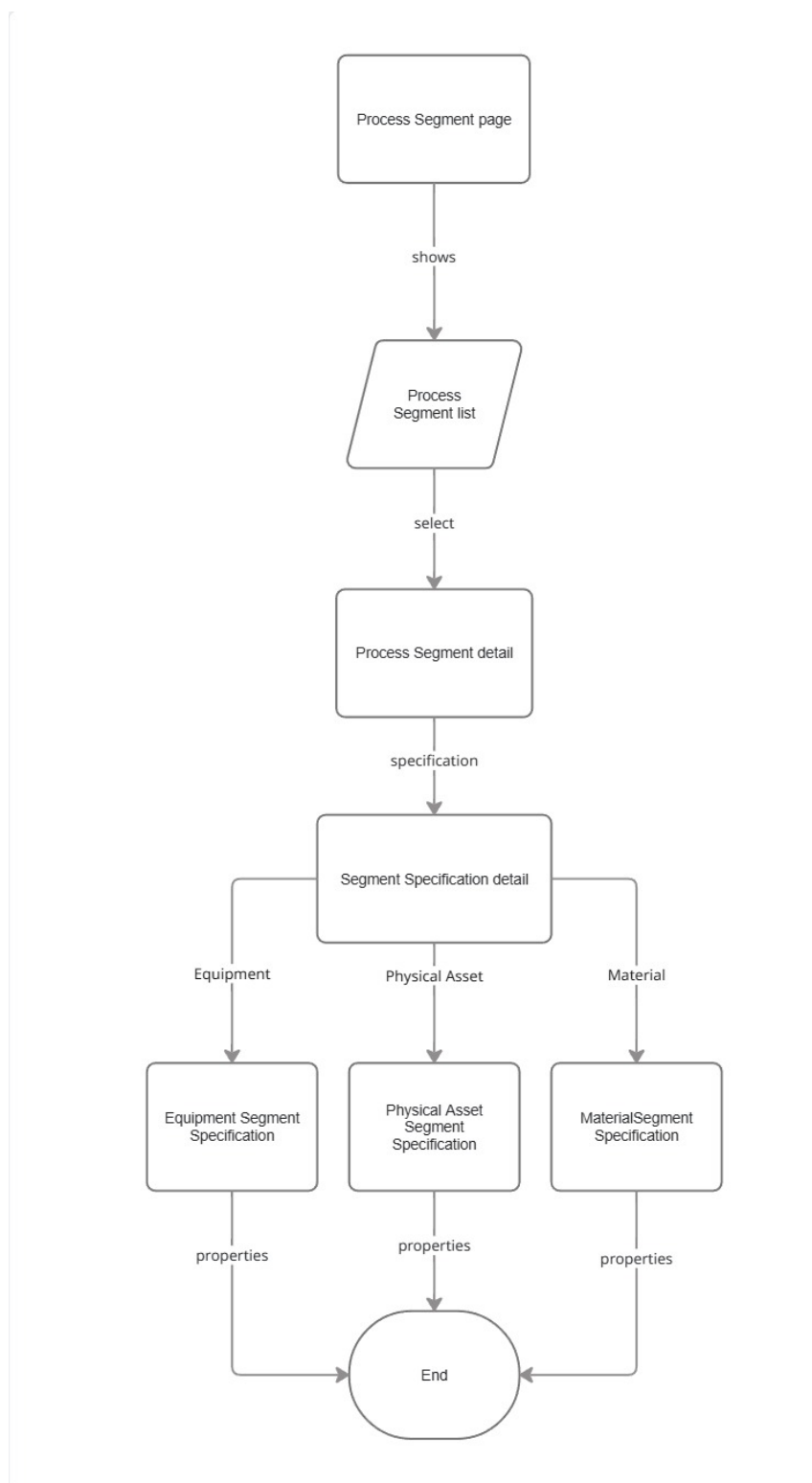


Figure C.5: Process Segment Pages Diagram

**C.1.6 Operations Definition Tab**

The Operations Definition dashboard displays a list of all defined operations within the system. Clicking an entry opens its main details view, which includes general information and a list of its associated Operations Segments. Selecting an Operations Segment navigates the user to a deeper detail level, detailing its composition across three sub-lists: Equipment Segments, Physical Asset Segments, and Material Segments. Finally, clicking on any element within these sub-lists reveals its specific properties and detailed configurations.

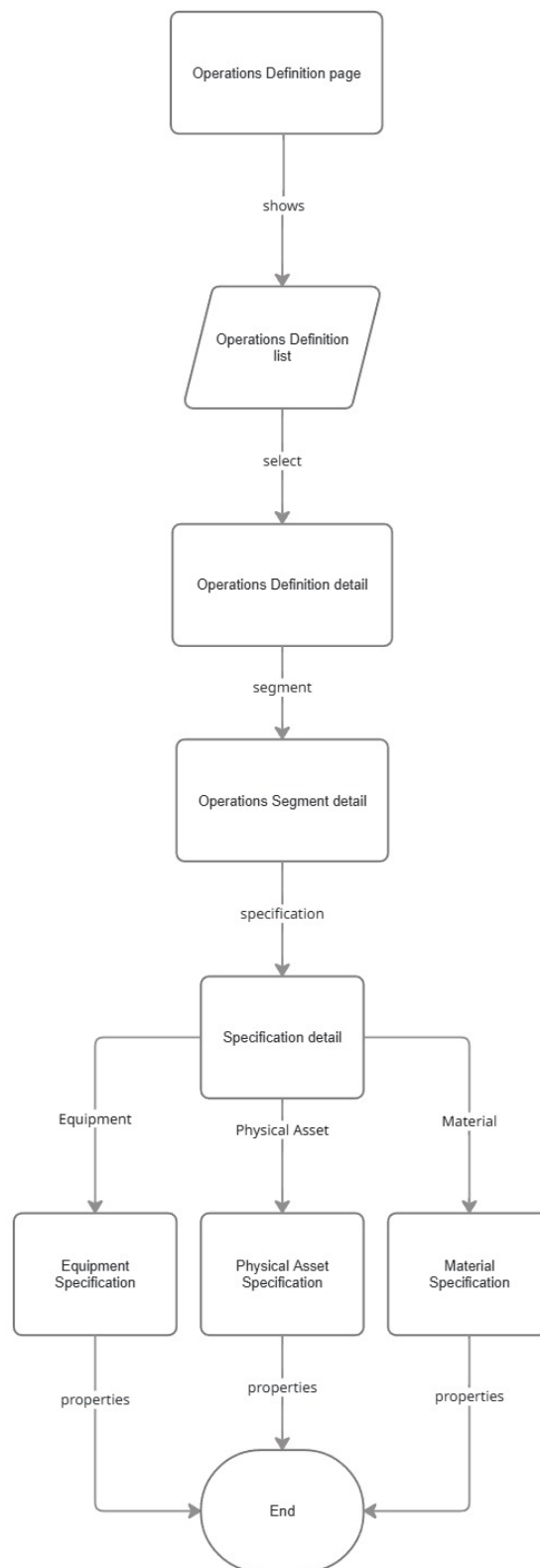


Figure C.6: Operations Definition Pages Diagram

**C.1.7 Operations Schedule Tab**

The Operations Schedule tab lists the scheduled operations in the system. Clicking on a schedule entry navigates to its details page, which displays scheduling parameters and a list of associated Operations Requests. Clicking on a specific Operations Request opens its detail view, revealing a list of its constituent Segment Requirements. Furthermore, selecting a Segment Requirement opens its dedicated page, listing the specific Equipment, Physical Asset, and Material Requirements needed. Selecting any element from these final requirement lists leads to the deepest detail page, displaying its properties and individual information.

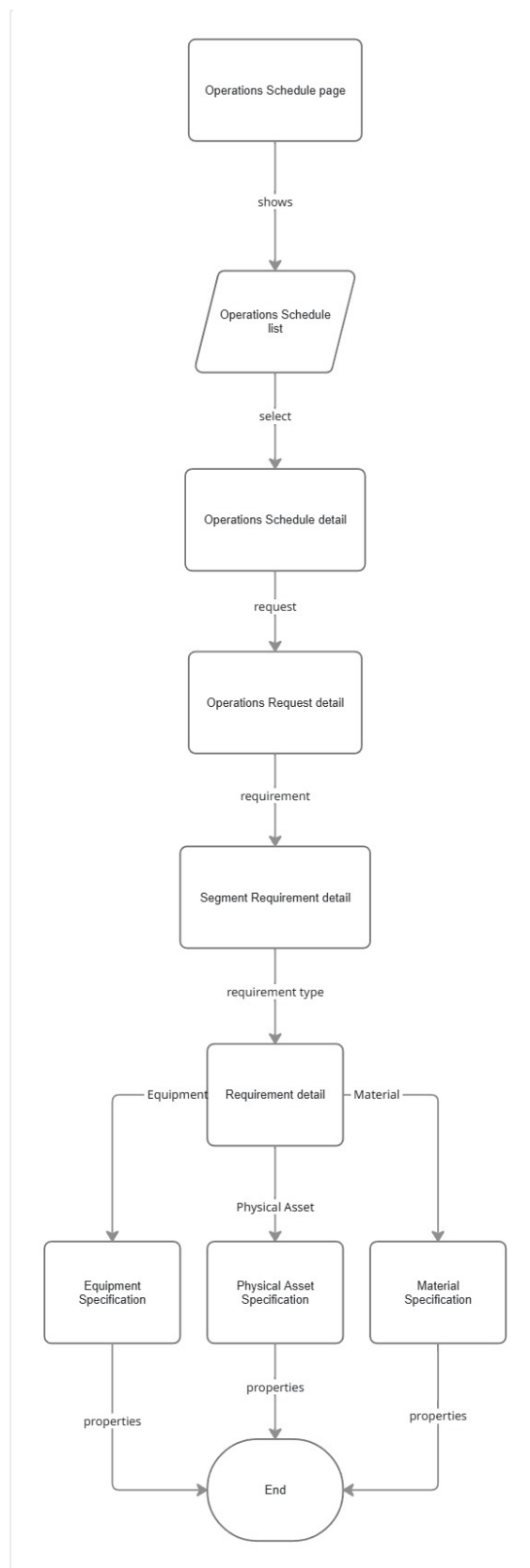


Figure C.7: Operations Schedule Pages Diagram

## C.2 Index - Home Page

The Home Page serves as the primary landing interface for the web application. It features a dynamic carousel component that provides a statistical overview of the system, displaying the current number of active records across the main ISA-95 information models stored in the relational database.

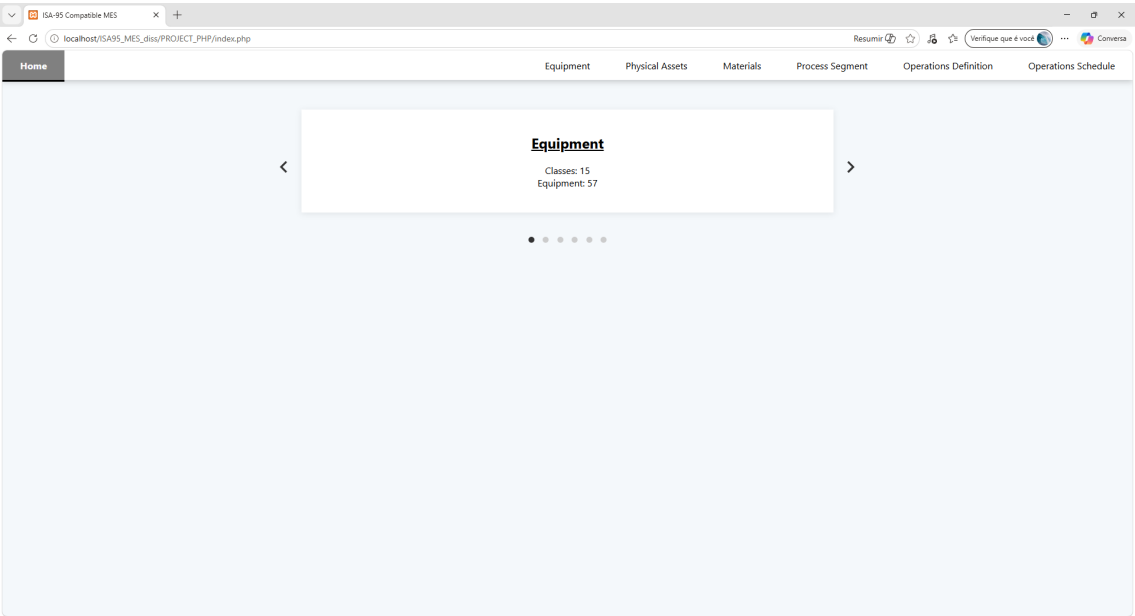


Figure C.8: Index Page - Equipment

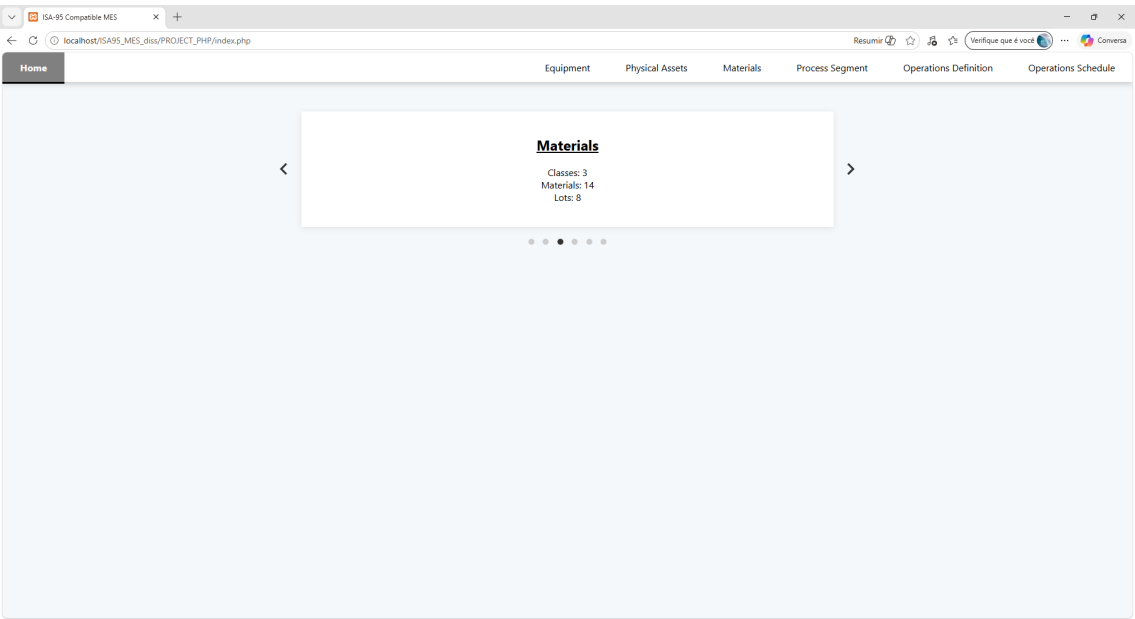


Figure C.9: Index Page - Material

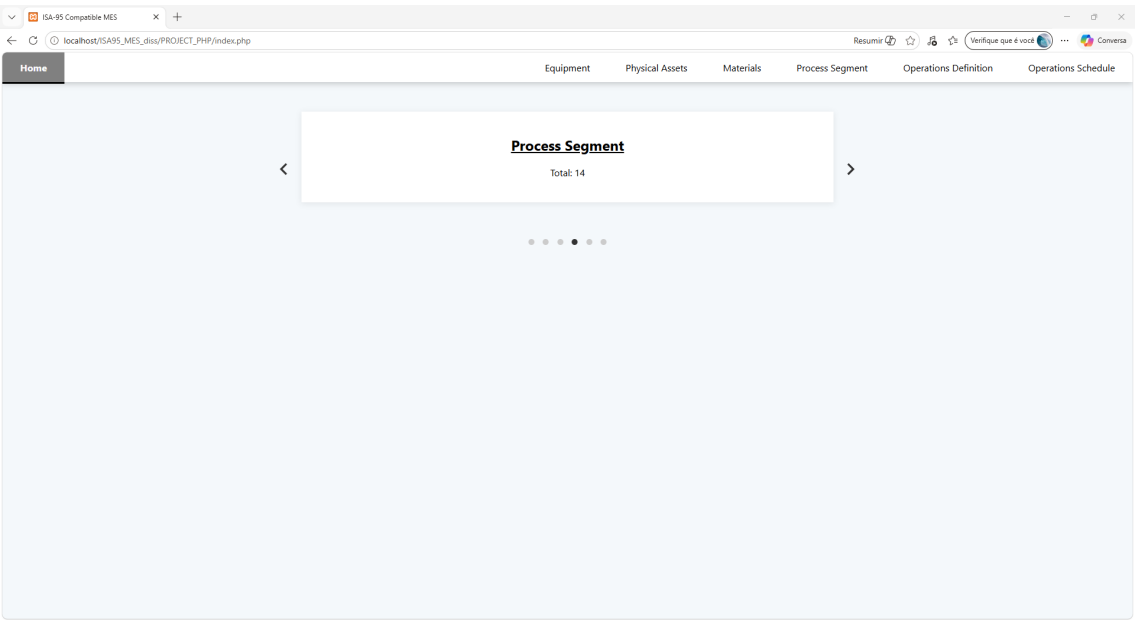


Figure C.10: Index Page - Process Segment

### C.3 Equipment

The Equipment main dashboard is structurally divided into two primary sections: Equipment Class and Equipment. Each section incorporates a dedicated data table to display previously stored records, alongside interactive options to facilitate the insertion of new assets into the system.

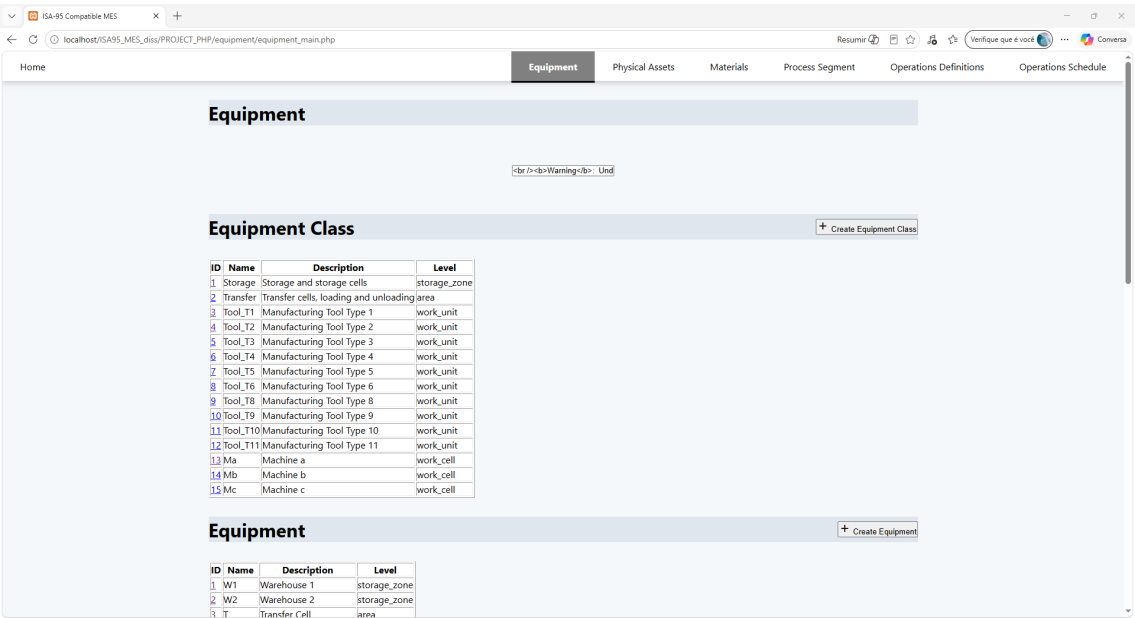
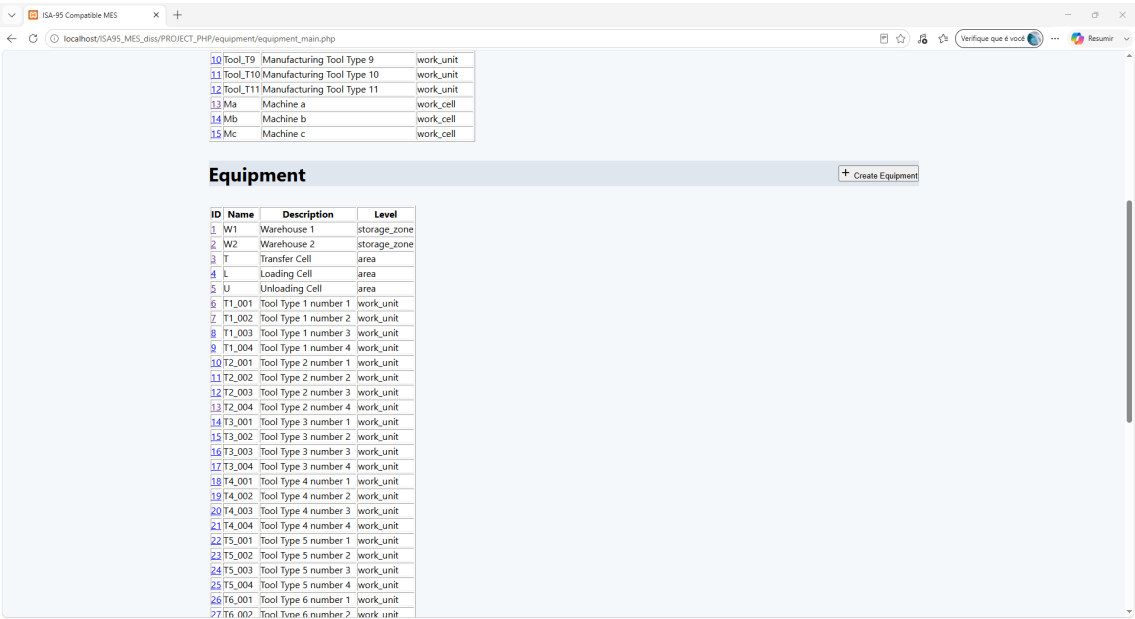


Figure C.11: Equipment Main Page 1

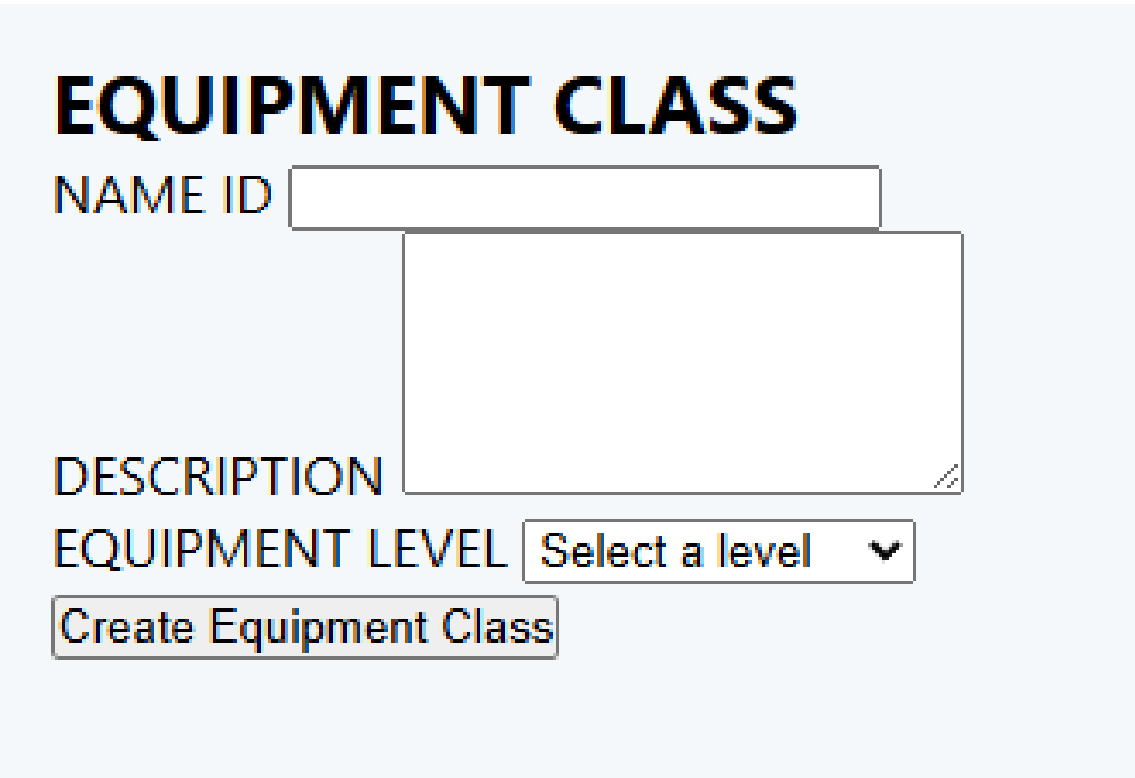


| ID | Name   | Description          | Level        |
|----|--------|----------------------|--------------|
| 1  | W1     | Warehouse 1          | storage_zone |
| 2  | W2     | Warehouse 2          | storage_zone |
| 3  | T      | Transfer Cell        | area         |
| 4  | L      | Loading Cell         | area         |
| 5  | U      | Unloading Cell       | area         |
| 6  | T1_001 | Tool Type 1 number 1 | work_unit    |
| 7  | T1_002 | Tool Type 1 number 2 | work_unit    |
| 8  | T1_003 | Tool Type 1 number 3 | work_unit    |
| 9  | T1_004 | Tool Type 1 number 4 | work_unit    |
| 10 | T2_001 | Tool Type 2 number 1 | work_unit    |
| 11 | T2_002 | Tool Type 2 number 2 | work_unit    |
| 12 | T2_003 | Tool Type 2 number 3 | work_unit    |
| 13 | T2_004 | Tool Type 2 number 4 | work_unit    |
| 14 | T3_001 | Tool Type 3 number 1 | work_unit    |
| 15 | T3_002 | Tool Type 3 number 2 | work_unit    |
| 16 | T3_003 | Tool Type 3 number 3 | work_unit    |
| 17 | T3_004 | Tool Type 3 number 4 | work_unit    |
| 18 | T4_001 | Tool Type 4 number 1 | work_unit    |
| 19 | T4_002 | Tool Type 4 number 2 | work_unit    |
| 20 | T4_003 | Tool Type 4 number 3 | work_unit    |
| 21 | T4_004 | Tool Type 4 number 4 | work_unit    |
| 22 | T5_001 | Tool Type 5 number 1 | work_unit    |
| 23 | T5_002 | Tool Type 5 number 2 | work_unit    |
| 24 | T5_003 | Tool Type 5 number 3 | work_unit    |
| 25 | T5_004 | Tool Type 5 number 4 | work_unit    |
| 26 | T6_001 | Tool Type 6 number 1 | work_unit    |
| 27 | T6_002 | Tool Type 6 number 2 | work_unit    |

Figure C.12: Equipment Main Page 2

C.3.1 Equipment Class Create Form

This form allows operators to define new equipment classes, establishing broad categories for the factory’s elements before individual machines are registered.



**EQUIPMENT CLASS**

NAME ID

DESCRIPTION

EQUIPMENT LEVEL

Figure C.13: Equipment Class Form

C.3.2 Equipment Create Form

This interface is used to register individual pieces of equipment, assigning them to a specific class and defining their primary identification attributes.

EQUIPMENT

NAME ID

DESCRIPTION

EQUIPMENT LEVEL

Select a level

CLASS

☐ Storage

☐ Transfer

☐ Tool\_T1

☐ Tool\_T2

☐ Tool\_T3

☐ Tool\_T4

☐ Tool\_T5

☐ Tool\_T6

☐ Tool\_T8

☐ Tool\_T9

☐ Tool\_T10

☐ Tool\_T11

☐ Ma

☐ Mb

☐ Mc

MADE UP OF

☐ No

☒ Yes

EQUIPMENT

☐ W1

☐ W2

☐ T

☐ L

☐ U

☐ T1\_001

☐ T1\_002

☐ T1\_003

☐ T1\_004

☐ T2\_001

☐ T2\_002

☐ T2\_003

☐ T2\_004

☐ T3\_001

☐ T3\_002

☐ T3\_003

☐ T3\_004

☐ T4\_001

☐ T4\_002

☐ T4\_003

☐ T4\_004

☐ T5\_001

☐ T5\_002

☐ T5\_003

☐ T5\_004

☐ T6\_001

☐ T6\_002

☐ T6\_003

☐ T6\_004

☐ T8\_001

☐ T8\_002

☐ T8\_003

☐ T8\_004

☐ T9\_001

☐ T9\_002

☐ T9\_003

☐ T9\_004

☐ T10\_001

☐ T10\_002

☐ T11\_001

☐ T11\_002

☐ C1

☐ M1a

☐ M1b

☐ M1c

☐ C2

☐ M2a

☐ M2b

☐ M2c

☐ C3

☐ M3a

☐ M3b

☐ M3c

☐ C4

☐ M4a

☐ M4b

☐ M4c

Create Equipment

Figure C.14: Equipment Form

C.3.3 Equipment Detail Page

The detail page provides an aggregated view of a specific equipment entity, rendering its structural dependencies, hierarchical parent-child relationships, and associated technical properties.

Home

Equipment

Physical Assets

Materials

Process Segment

Operations Definitions

Operations Schedule

Home

Equipments

M1a

M1a

Edit Equipment

Delete Equipment

NAME

M1a

LEVEL

work\_cell

DESCRIPTION

Machine 1a - Cell 1

Properties

Add Equipment Property

No equipment properties found.

Classes List

| ID | Name | Description | Level     |
|----|------|-------------|-----------|
| 13 | Ma   | Machine a   | work_cell |

Made Up Of

| ID | Name   | Description          | Level     |
|----|--------|----------------------|-----------|
| 6  | T1_001 | Tool Type 1 number 1 | work_unit |
| 10 | T2_001 | Tool Type 2 number 1 | work_unit |
| 14 | T3_001 | Tool Type 3 number 1 | work_unit |

Figure C.15: Equipment Detail Page

C.3.4 Equipment Property Form

This dedicated form enables the parameterization of specific technical properties and metrics for a previously registered piece of equipment.

# EQUIPMENT PROPERTY

NAME ID

DESCRIPTION

VALUE

UNIT OF MEASURE

MAPS TO

☐ Tool\_Change\_Time

Add Equipment Property

Figure C.16: Equipment Property Form

## C.4 Process Segment

The Process Segment main page provides a comprehensive tabular view of all logical manufacturing steps registered in the system. From this interface, users can initialize a new process segment or select an existing one to inspect its detailed specifications.

HomeEquipmentPhysical AssetsMaterialsProcess SegmentOperations DefinitionsOperations Schedule

### Process Segments

<br /><b>Warning</b><br /> Und

Process Segment

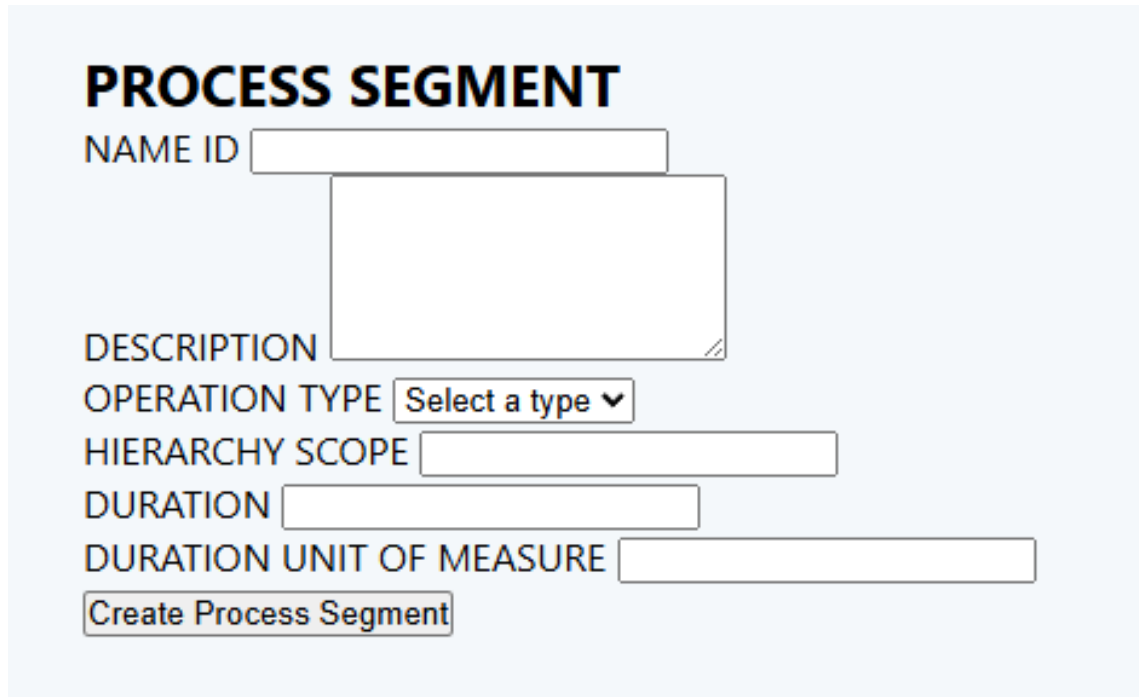
+ Create Process Segment

| ID | Name                | Description                          | Operations Type | Hierarchy | Scope | Duration | Duration Unit | Unit of Measure |
|----|---------------------|--------------------------------------|-----------------|-----------|-------|----------|---------------|-----------------|
| 1  | PROC_WOOD_RTOW_T1   | Wood to RtopW using Tool T1          | production      | Machine   |       | 30.00    | s             |                 |
| 2  | PROC_WOOD_STOPW_T2  | Wood to StopW using Tool T2          | production      | Machine   |       | 20.00    | s             |                 |
| 3  | PROC_WOOD_LEGW_T3   | Wood to LegW using Tool T3           | production      | Machine   |       | 10.00    | s             |                 |
| 4  | PROC_WOOD_LEGW_T10  | Wood to LegW using Tool T10          | production      | Machine   |       | 30.00    | s             |                 |
| 5  | PROC_METAL_RTOW_T4  | Metal to RtopM using Tool T4         | production      | Machine   |       | 35.00    | s             |                 |
| 6  | PROC_METAL_STOPM_T6 | Metal to StopM using Tool T6         | production      | Machine   |       | 25.00    | s             |                 |
| 7  | PROC_METAL_LEGM_T5  | Metal to LegM using Tool T5          | production      | Machine   |       | 30.00    | s             |                 |
| 8  | PROC_METAL_LEGM_T11 | Metal to LegM using Tool T11         | production      | Machine   |       | 40.00    | s             |                 |
| 9  | PROC_RWW_T8         | RtopW + 2x LegW to RWW using Tool T8 | production      | Machine   |       | 10.00    | s             |                 |
| 10 | PROC_SWW_T8         | StopW + 2x LegW to SWW using Tool T8 | production      | Machine   |       | 10.00    | s             |                 |
| 11 | PROC_RWM_T9         | RtopW + 2x LegM to RWM using Tool T9 | production      | Machine   |       | 10.00    | s             |                 |
| 12 | PROC_SWM_T9         | StopW + 2x LegM to SWM using Tool T9 | production      | Machine   |       | 10.00    | s             |                 |
| 13 | PROC_RMM_T8         | RtopM + 2x LegM to RMM using Tool T8 | production      | Machine   |       | 10.00    | s             |                 |
| 14 | PROC_SMM_T8         | StopM + 2x LegM to SMM using Tool T8 | production      | Machine   |       | 10.00    | s             |                 |

Figure C.17: Process Segment Main Page

### C.4.1 Process Segment Form

An input form dedicated to the creation of new process segments, establishing the core identifier for a new manufacturing transformation.

The image shows a web form titled "PROCESS SEGMENT" in bold black text. Below the title, there are several input fields: "NAME ID" followed by a text box, "DESCRIPTION" followed by a larger text box with a diagonal line in the bottom right corner, "OPERATION TYPE" followed by a dropdown menu showing "Select a type" with a downward arrow, "HIERARCHY SCOPE" followed by a text box, "DURATION" followed by a text box, and "DURATION UNIT OF MEASURE" followed by a text box. At the bottom of the form is a button labeled "Create Process Segment".

**PROCESS SEGMENT**

NAME ID

DESCRIPTION

OPERATION TYPE

HIERARCHY SCOPE

DURATION

DURATION UNIT OF MEASURE

Figure C.18: Process Segment Form

### C.4.2 Process Segment Detail Page

This page offers a detailed breakdown of a selected process segment, dynamically aggregating and displaying its associated equipment requirements and material flow logic.

Home

EquipmentPhysical AssetsMaterialsProcess SegmentOperations DefinitionsOperations Schedule

Home > Process Segment > PROC\_WOOD\_RT0PW\_T1

PROC\_WOOD\_RT0PW\_T1

Edit Process SegmentDelete Process Segment

NAME

PROC\_WOOD\_RT0PW\_T1

OPERATION TYPE

production

HIERARCHY SCOPE

Machine

DURATION

30.00

DURATION UNIT OF MEASURE

s

DESCRIPTION

Wood to RtopW using Tool T1

Equipment Segment Specification

Add Equipment Segment Specification

| ID | Equipment Class ID | Equipment ID | Description | Equipment Use  | Quantity | Unit of Measure |
|----|--------------------|--------------|-------------|----------------|----------|-----------------|
| 1  | 3                  |              | Tool T1     | Transformation | 1.00     | unit            |

Physical Asset Segment Specification

Add Physical Asset Segment Specification

No physical asset segment specification found.

Material Segment Specification

Add Material Segment Specification

| ID | Material Class ID | Material Definition ID | Description  | Assembly Type | Assembly Relation | Material Use | Quantity | Unit of Measure |
|----|-------------------|------------------------|--------------|---------------|-------------------|--------------|----------|-----------------|
| 1  |                   | 1                      | Input Wood   | physical      | permanent         | consumed     | 1.00     | unit            |
| 2  |                   | 3                      | Output RtopW | physical      | permanent         | produced     | 1.00     | unit            |

localhost/GASS\_MES\_dss/PROJECT\_PHP/index.php

Figure C.19: Process Segment Main Page

C.4.3 Equipment Segment Specification Form

This specialized form is used to bind specific tool and machinery requirements, as well as processing times, to a target process segment.

EQUIPMENT SEGMENT SPECIFICATIONS

☒ Equipment Class

☐ Equipment

EQUIPMENT CLASS

☐ Storage ☐ Transfer ☐ Tool\_T1 ☐ Tool\_T2 ☐ Tool\_T3 ☐ Tool\_T4 ☐ Tool\_T5 ☐ Tool\_T6 ☐ Tool\_T8 ☐ Tool\_T9 ☐ Tool\_T10 ☐ Tool\_T11 ☐ Ma ☐ Mb ☐ Mc

DESCRIPTION

EQUIPMENT USE

QUANTITY

UNIT OF MEASURE

Create Equipment Segment Specifications

Figure C.20: Equipment Segment Specification Form

C.4.4 Material Segment Specification Form

This interface enables the operator to define the material logic of a process segment, specifying the required raw material inputs and the resulting product outputs.

## MATERIAL SEGMENT SPECIFICATIONS

☒ Material Class ☐ Material Definition

MATERIAL CLASS

☐ Raw Material ☐ Intermediate ☐ Final Product

DESCRIPTION

ASSEMBLY TYPE 

Select a type ▼

ASSEMBLY RELATIONSHIP 

Select a relationship ▼

MATERIAL USE 

Select the use ▼

QUANTITY

UNIT OF MEASURE

Create Material Segment Specification

Figure C.21: Material Segment Specification Form

## Appendix D

# MES Core Program

### D.1 Package Structure and Modular Organization

|                            |                                                                 |
|----------------------------|-----------------------------------------------------------------|
| MES/                       | Main project folder                                             |
| ├─ main.py                 | Application entry point                                         |
| ├─ send_xml.py             | Simulation script for sending XML files                         |
| ├─ erp_orders.xml          | XML file containing ERP test orders                             |
| ├─ communication/          | Communication module                                            |
| │   └─ erp_listener.py     | Listens and receives XML orders via<br>Sockets                  |
| ├─ core/                   | Core business logic                                             |
| │   └─ mes.py              | Central orchestrator of the MES behavior                        |
| │   └─ dispatcher.py       | Task distribution/dispatching to work<br>cells                  |
| │   └─ material_planner.py | Stock and raw material planning and<br>validation               |
| │   └─ scheduler.py        | Production scheduler and plan/agenda<br>creator                 |
| ├─ data/                   | Data access layer                                               |
| │   └─ db_connect.py       | MySQL database connection management                            |
| │   └─ queries.py          | SQL queries and commands (ISA-95 mapping)                       |
| │   └─ test_db_connect.py  | Database connection test script                                 |
| │   └─ test_queries.py     | Query validation and test script                                |
| ├─ opcua/                  | OPC UA communication module                                     |
| │   └─ plc_client.py       | Communication with the PLC via OPC UA to<br>send the final plan |
| ├─ parser/                 | Parsing module                                                  |
| │   └─ xml_parser.py       | Parses the received ERP XML into Python<br>dictionaries         |

## Appendix E

# Validation and Results

### E.1 Transformations Diagrams

Wood to RtopW

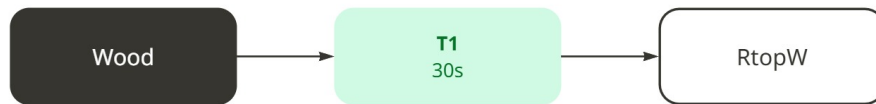


Figure E.1: Transformation - Wood to RtopW

Wood to StopW

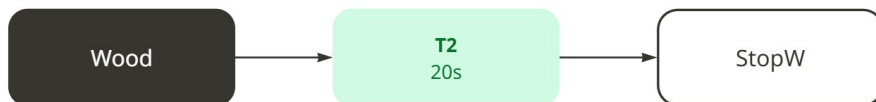


Figure E.2: Transformation - Wood to StopW

Wood to LegW (T3)

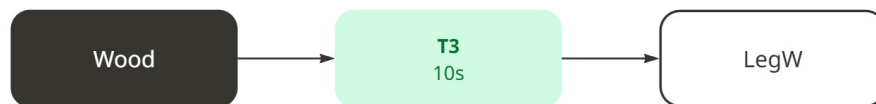


Figure E.3: Transformation - Wood to LegW

Wood to LegW (T10)

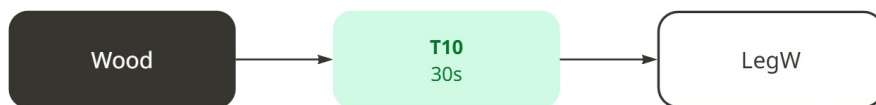


Figure E.4: Transformation - Wood to LegW

Metal to RtopM

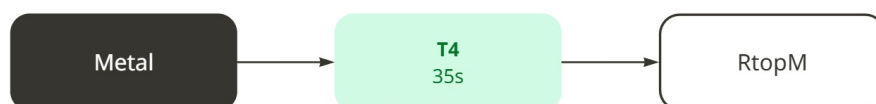


Figure E.5: Transformation - Metal to RtopM

Metal to StopM

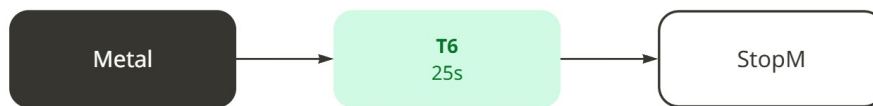


Figure E.6: Transformation - Metal to StopM

Metal to LegM (T5)

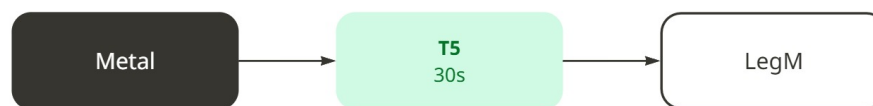


Figure E.7: Transformation - Metal to LegM

Metal to LegM (T11)



Figure E.8: Transformation - Metal to LegM

RtopW + 2x LegW to RWW

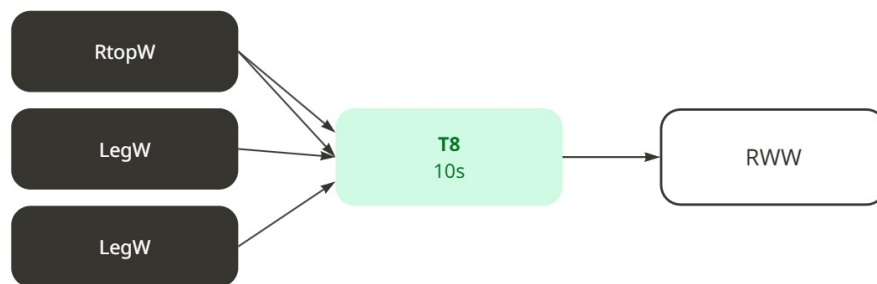


Figure E.9: Transformation - Assemble to RWW

StopW + 2x LegW to SWW

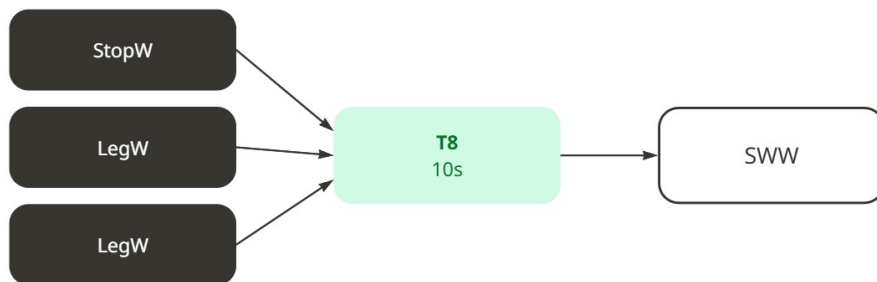


Figure E.10: Transformation - Assemble to SWW

RtopW + 2x LegM to RWM

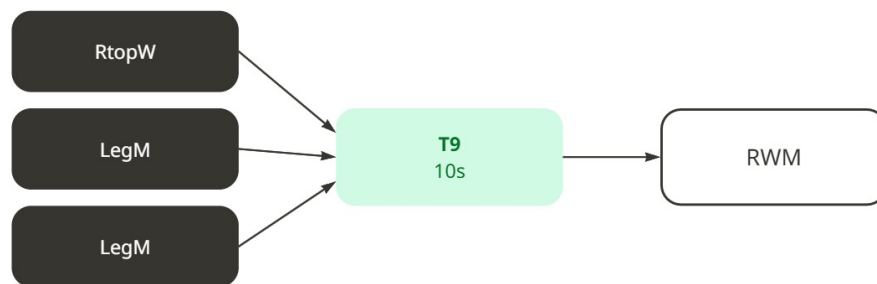


Figure E.11: Transformation - Assemble to RWM

StopW + 2x LegM to SWM

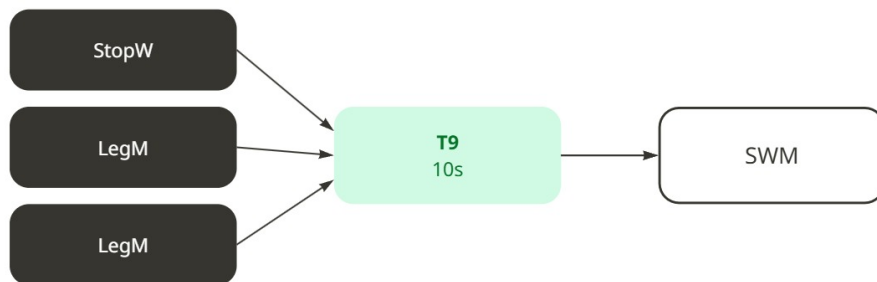


Figure E.12: Transformation - Assemble to SWM

RtopM + 2x LegM to RMM

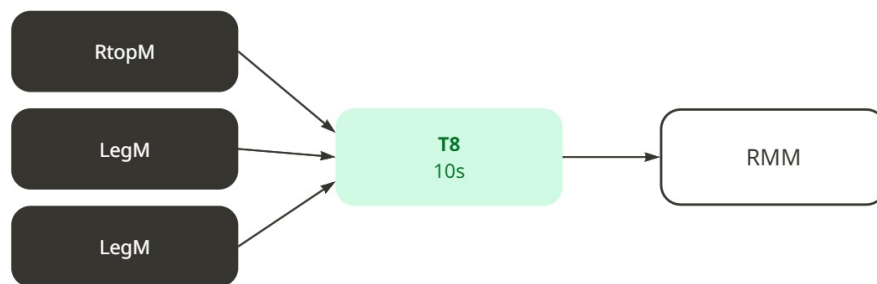


Figure E.13: Transformation - Assemble to RMM

StopM + 2x LegM to SMM

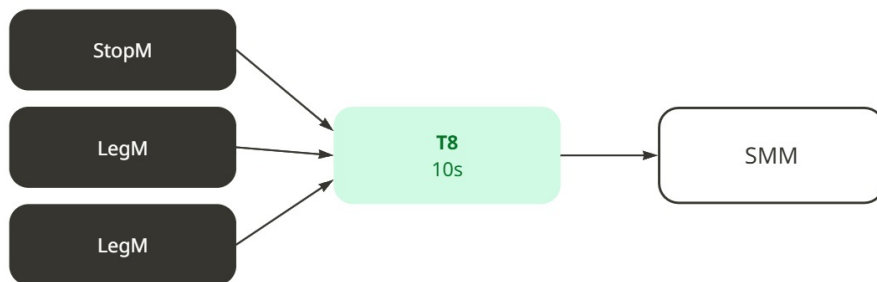


Figure E.14: Transformation - Assemble to SMM