Industrial Informatics

Masters in Electrical and Computer Engineering

2021/2022

Automation of a Production Line

WARNING

The customer may need to make changes to the specifications up to 21 days before the delivery deadline.

Mário de Sousa 2022-03-08 ver. 1.1

1 Introduction

This document describes the work you will be doing throughout the Industrial Informatics course. In a simplified way, it consists of:

> **Requirement 1**: automating the production of a flexible production line simulator,
>
> **Requirement 2:** development of a MES (Manufacturing Execution System) to monitor and manage the production on the plant floor
>
> **Requirement 3:** development of an ERP (Enterprise Resource Planning) module to receive orders from clients and schedule acquisition orders as well as production orders.

Typically the lowest level control (requirement 1) will use PLCs or equivalent equipment. When implementing the higher level requirements (2 and 3) it is advisable to use higher level programming languages and tools (C++, Java, Python, C#, …). At these levels the use of a DBMS (Data-Base Management System) is also mandatory (PostgreSQL, MariaDB, ...).

A separate document describes the equipment in the production line and the interfaces provided to interact with it. The document you are currently reading describes the requirements that the control, monitoring, management and scheduling applications must comply with. To better understand these requirements we suggest you first peruse the document describing the the production line equipment.

2 Overview of the Flexible Production Line

The Flexible Production Line to be automated consists of 5 cells (Fig. 1 & Fig. 2):

- a cell for receiving raw material (A),
- two automatic warehouses (B and D),
- a machining cell with 6 machines (C),
- a cell for shipping processed parts (E).

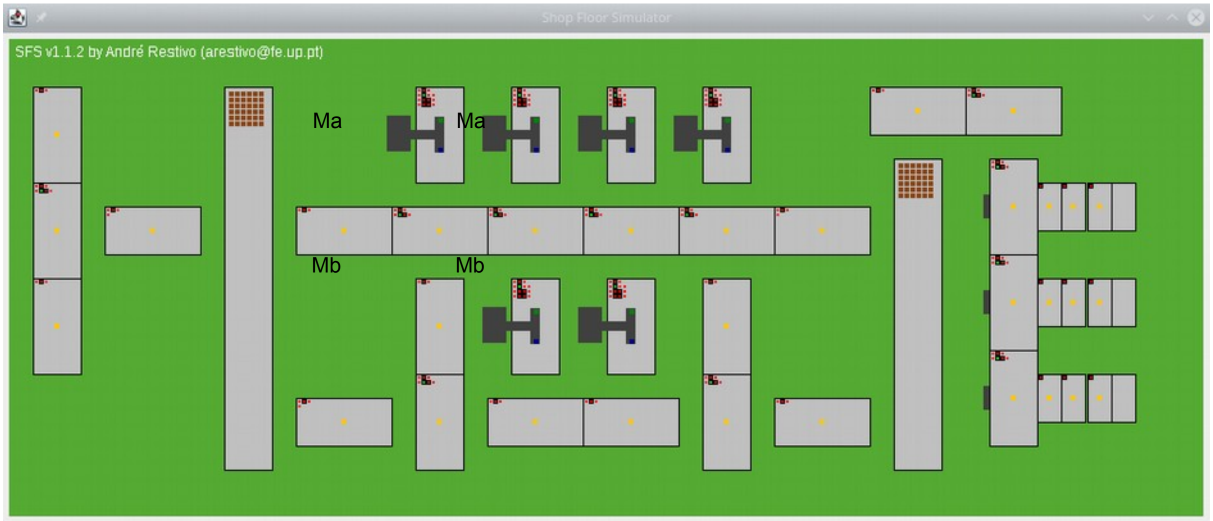

Figure 1: Flexible Production Line – *equipment layout*.

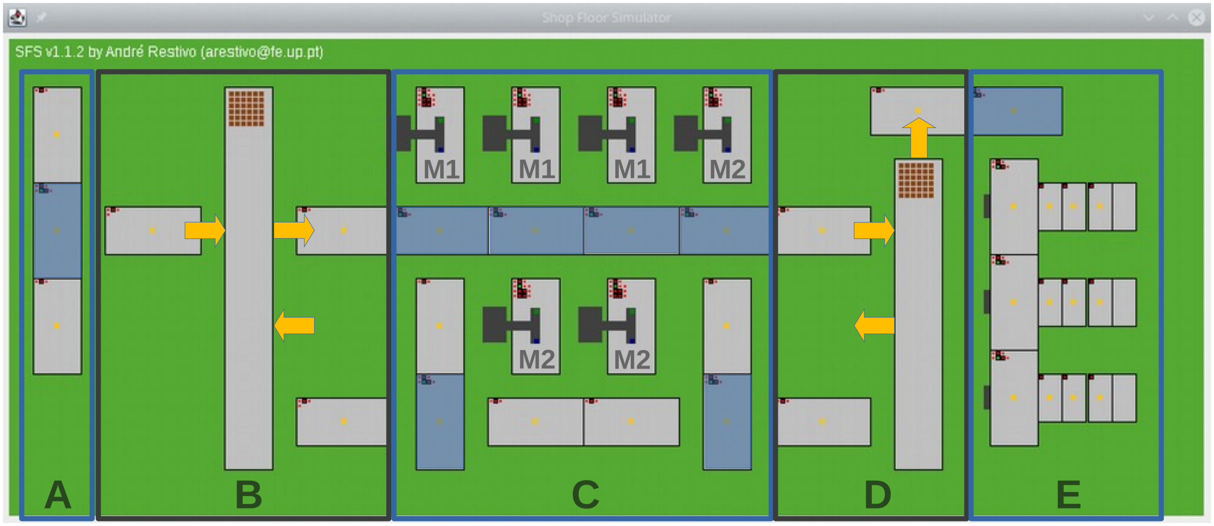

Figure 2: Flexible Production Line – annotated equipment layout.

The annotations in Fig. 2 help in understanding the equipment capabilities.

- The large vertical rectangles with brown dots/squares represent automatic warehouses;
- All conveyors (grey or bluish grey rectangles) are bi-directional. Yellow dots represent sensors that can detect the presence of a work-piece;
- Conveyors marked with a blue tinge can rotate about their vertical axis, i.e. they can rotate so as to have a vertical (North-South) or horizontal (East-West) alignment. This allows the work-pieces to come from / be sent to any direction (N, S, E or W);
- Conveyors marked with an orange arrow can insert / remove work-pieces from the respective warehouse;
- Machine types identified as M1 and M2 are configured with different sets of tools. M1 has tools T1, T3 and T4, while M2 has tools T2, T3 and T4.

This production line will be simulated by the SFS software (Shop Floor Simulator) which, being developed in Java, is compatible with Windows, Linux and OSX. Access to the logic signals (sensors and actuators) is done through the Modbus/TCP protocol. The SFS simulator implements a Modbus/TCP server, in which the sensors (Boolean signals) are mapped to Input Discrete, and the actuators (Boolean signals) on to Coils. The addresses of each signal are available in a file io.csv (format comma separated values, which can be opened as a spreadsheet). This file is generated by the simulator itself with the command:

java -jar sfs.jar –csv io.csv

The description of the signals available to control each equipment (machine, conveyor, rotating conveyor, pusher and warehouse) is described in the accompanying document.

2.1 The Production (Transformation) Process

The production line processes different types of parts/work-pieces (P1, P2, ...), represented by blocks with different colors. The color convention used is defined in the following table (same color convention as for electrical resistors):

Piece	Colour
P1	Brown
P2	Red
P3	Orange
P4	Yellow
P5	Green
P6	Blue
P7	Violet
P8	Grey
P9	White

Table 1: Mapping of work-piece type to colour .

When processing a part of a certain type on one of the machines, using a given tool, for a certain period of time, this part changes its type.

Each machine has a private warehouse of three tools (T1, T3, T4 for M1 machines, and T2, T3, T4 for M2 machines), among which you can alternate in order to produce different parts according to the raw material supplied. Changing between any two tools takes 30 seconds. The transformations of parts that may occur depend on the type of tool, and are described in the following table:

Starting Piece	Produced Piece	Tool	Processing Time
P1	P6	T1	20 s
P2	P3	T2	10 s
P2	P4	T3	10 s
P2	P5	T4	15 s
P3	P6	T1	20 s
P4	P7	T4	10 s
P6	P8	T3	30 s
P7	P9	T3	10 s

Table 2: Work-piece Transformations

If you try to perform operations that are not found in the table above (for example: processing P8 with tool T3), the part will be unusable and will be represented in black. Similarly, if you perform an operation described in the table above, but for a longer period of time than indicated, the part is also unusable. Black work-pieces should never occur in your project as machines never break down. Trying to load a black work-piece in a warehouse will result in the simulator crashing.

If you start to perform a transformation operation but stop before the necessary time has elapsed the transformation becomes incomplete, leaving the piece with the two colors (of the original piece, and of the final piece) in the correct proportion. The transformation can be completed at a later time. Work-pieces that are partially transformed may be moved out of the machines but cannot be placed in a warehouse, i.e. they must remain on the conveyors.

In order to produce a final piece it is necessary to take into account the processing sequence of the raw materials. This sequence results from the successive combination of raw material and tool, which may generate by-products for further processing. For example, if you want to produce a P8 piece from a P1, it will be necessary to apply T1 to the P1 resulting in a by-product of the type P6, followed by T3 resulting in P8.

2.2 Warehouses

Cells B and D contain warehouses to store work-pieces. Both warehouses can store any work-piece type (P1 to P9). You are free to manage what work-pieces to store in each warehouse, although initial expectations are that the warehouse on cell B will be used to store raw material (i.e. work-pieces P1 and P2), whereas the warehouse on cell D will usually store finished products (i.e. work-pieces P3 to P9).

Any loading or unloading operation executed by a warehouse takes 2 s to complete. Although the simulator does not enforce it, you should consider both warehouses as having a limited capacity:

- warehouse on cell B → capacity of 24 work-pieces
- warehouse on cell D → capacity of 24 work-pieces

The production line starts with both automated warehouse empty (no work-pieces).

2.3 The Loading Process

Cell A is used for loading raw material. Loading is carried out by the topmost conveyor (identified as Load1 by SFS) as well as the lowermost conveyor (identified as Load2). Load1 is used to load P1 work-pieces, whereas Load2 is reserved for P2 work-pieces. Both P1 and P2 work-pieces are considered raw material that the production line consumes, and cannot create.

The loading stations function as normal conveyors on which pieces will be manually deposited. These work-pieces must be transferred to the cell B warehouse. If you wish, you may later transfer these work-pieces to cell D warehouse using the conveyors on cell C.

2.4 The Unloading Process

Cell E is used for unloading work-pieces through the three unloading bays/docks identified as Dock1, Dock2 and Dock2.

Each unloading station consists of a pusher and a slider. The slider is an inclined ramp fitting a maximum of 4. The pusher is a mechanism that pushes the piece off the conveyor in line with the slider to the respective slider. The pieces will be removed manually from the far right of each slider. Each position of the slider (except the last one, to the right) has a sensor that detects the presence of a work-piece.

3 Enterprise Management (ERP)

3.1 Client Orders

The company's ERP software must accept orders sent to it by its clients. These orders will be encoded in XML files, and will be sent to port 54321 of the UDP / IP protocol. Each file may include one or more client orders, coded according to the following rules:

– **Client Order:**

The XML will contain:
<Client Name/Id="name">
<Order Number="nnn">
<WorkPiece="Px" Quantity="XX" DueDate="DD" LatePen="LP" EarlyPen="EP"/>
</Order>

In which:
name – Name identifying the client
nnn – order number
Px – work-piece type being ordered
XX – quantity being ordered
DD – Due date (integer)
LP – penalty (value in €) incurred for each day of delay in completing the order
EP – penalty (value in €) incurred for each day of advance in completing the order

The ordered work-pieces must be delivered on the due-date. They must not be delivered early, nor delivered late. The stated penalties apply when the due-dates are not satisfied.

3.2 Plans

Your ERP software is expected to generate a Master Production Schedule (MPS) into the future, **if possible** trying to maximize enterprise profit. Usually this means satisfying all client orders so as to avoid penalties, of minimizing acquisition costs of raw material, and of minimizing overall production costs. This MPS should define for each day in the future:

- the amount of work-pieces of each type you expect to have by the end of that day
- the capacity used in each warehouse

Based on the MPS you must generate a production plan as well as a purchasing plan. The production plan defines what orders to send to the MES on each day, while the purchasing plan defines when to place orders for purchasing raw material (P1, P2) from your suppliers.

You have available only 3 suppliers described in the following table:

Supplier	Piece	Minimum order	Price per piece	Delivery time
SupplierA	P1	16	€ 30	4 days
SupplierA	P2	16	€ 10	4 days
SupplierB	P1	8	€ 45	2 days
SupplierB	P2	8	€ 15	2 days
SupplierC	P1	4	€ 55	1 day
SupplierC	P2	4	€ 18	1 day

Table 3: Raw material suppliers

The ERP must provide a user interface that shows the resulting plans (MRP, Purchasing and Production plans), as well as the current date. The plan should also include a calculation of the costs incurred in satisfying each client order.

3.3 Unit Cost

The cost (per work-piece) for each order should be determined using the following formula:

$$Tc = Rc + Pc + Dc$$

Tc – Total Cost
Rc – Raw Material Cost (price of the raw material used to produce that piece)
Pc – Production Cost Cost to Produce the piece
Dc – Depreciation Cost (Cost of money invested in the piece)

$$Dc = Rc \times (Dd - Ad) \times 1\%$$

Ad – Arrival Date – date the raw material arrived at the production line
Dd – Dispatch Date – date final work-piece leaves the production line (unloaded on cell E)

$$Pc = € \, 1 \times Pt$$

Pt – Total Production time (in seconds).

Notice that Pt includes the time a work-piece spends in a machine, even if the machine is not doing a transformation (e.g. waiting for the output conveyor to become free, changing a tool).

Notice too that the work-pieces produced for the same client order may have different values of Total Cost. The raw material cost may be different (ordered from different manufacturers), as may the production cost (time spent in the machines) and the Depreciation Cost (time spent in the manufacturing line). The ERP should therefore show the average cost for each piece in the client's order.

NOTE

So as to speed up the simulation, we will consider each day to be equivalent to 60 seconds of simulation time.

NOTE

You **DO NOT** need to implement a optimization algorithms at the ERP level that try to reduce total cost to the enterprise. You only need to calculate the total cost of each client order.

A competition will be held between student groups. This competition will be won by the group that completes all orders with the minimum cost. To win this competition you **may** need to implement an optimization algorithm.

4 Managing the Plant floor (MES)

4.1 MES Orders

The Manufacturing Execution System (MES) receives production orders form the ERP and manages the execution of these orders on the plant floor. It is up to you to define how the ERP and the MES interact, both logically (i.e. what information to send/receive and when) and technologically (i.e. how that information is exchanged, e.g. communication protocol, …).

Typically the ERP will emit production and/or delivery orders once a day. The MES must also provide the ERP with sufficient information so the ERP can calculate the real production cost.

It is expected that the MES will prioritize the orders to be executed in such a way as to minimize production costs, while guaranteeing that the warehouse capacity is never exceeded.

The MES should also choose and indicate the unloading dock to use for each delivery order. Orders with more than 4 work-pieces must use more then one dock. Delivery orders should be placed on the unloading docks in the first half of the day (first 30 s of each 60 s day). The user will manually remove these work-pieces from the unloading dock on the second half of the day.

It is expected that the ERP will never ask to deliver more than 12 work-pieces per day.

The MES must provide the PLC with sufficient information so that the PLC can control the manufacturing equipment. It is up to you to define when and what information to exchange between the MES and the PLC. However, the client mandates that you use the **OPC-UA** communication protocol to exchange this data.

4.2 User Interface

The MES should provide a user interface through which the user can monitor the current production order list, as well as the status of each order received from the ERP. The user interface should also provide enough information for the user to determine the status of that algorithm. This information will depend on the execution ordering algorithm used by the MES (e.g. the list of production orders in decreasing priority).

4.3 Statistics:

To allow users to monitor the production process, the MES must provide the following information:

- **Machine Statistics**

  A list with the statistics of the operation of each machine. For each machine, provide:
  - total operating time,
  - number of operated work-pieces (total and for each type).

- **Unloaded Work-pieces**

  For each unloading dock, provide:
  - the number of unloaded work-pieces (total and by type).

The interface through which the monitoring is carried out is not specified, so you are free to implement it as you see fit. If you wish, you can extend the communication protocol so that they are made available in XML format on the UDP port.

5 Controlling the Production Equipment (PLC)

For requirement 1 (section §1) you will need to develop the software to control the equipment that is part of the cells of the production line.

You are free to use this equipment as you see fit as long as the simulator manages to produce work-pieces with the correct colour. Most notably you are free to store each work-piece on whatever warehouse you think most appropriate, or even keep it on a conveyor until needed. Work-pieces can be transported in the production line (using the conveyors) in any direction.

Please note that the simulator will have very slight timing differences depending on the computer it is running. We strongly suggest you do **not** rely on timing delays when controlling the manufacturing plant, with the sole exception when timing the work-piece transformations inside a machine. The final evaluation may occur on a computer with slight timing differences, and your code must still work reliably.

6 Persistence

The information managed by the ERP and MES must be persistent, i.e. it should not be lost if the ERP and/or MES have to be turned off. This requirement must be achieved using a database managed by a database management system (e.x .: MariaDB, MySQL, PostgreSQL, ...).

The information to be placed in the database should be such as to allow the ERP and MES to be disconnected and resumed later without loss of information regarding pending orders, or statistics produced up to that time. During the period in which the ERP and/or MES is disconnected it is acceptable to stop accepting new orders from the ERP or client.

With MES turned off, production operations may also be affected, and the level at which they are affected will depend on the way in which the architecture of the production system is organized.

7 Expected Outcomes

This project should be carried out by groups of 5 students.

All the operating logic behind the control of the production line must be analyzed, designed and implemented by the students, who must identify, in addition to the components described above, all the components that will be necessary for the operation of the line. It is recommended that you start first by not considering the possible optimization of the use of available resources. However, the optimization of resources will not fail to be valued when evaluating the project.

The expected results of the project are:

1. **Application architecture model**, composed, at least, by UML class, object, and sequence diagrams. To be delivered by **3rd April 2022**;
2. **Final Report** (50% of the project grade), to be delivered by **9 June 2022.**
3. **Demonstration** and discussion of the application (50% of the project grade), in the week of **6 to 9 of June 2022**.

<u>Distribution of effort between group elements</u>:

The project grades will be attributed to the group. The report should indicate how the group wants this grade to be distributed to each member of the group. This distribution must be made as a percentage in such a way that the sum of the percentages is 100.

final individual grade calculation	
Final Group Grade: G (0..20)	A = Max[0; SIN(pi/2 * (Px/P - 1) / (1/P - 1))]
Weight for each group member: P1, P2, P3, P4, P5	B = Min[0; (Px/P - 1)]
P = Average(P1, P2, P3, P4, P5)	C = (20-G)*G/20 * A
Individual Grade Gx = G + C + D	D = G * B

<u>Final Report</u>:

The report should be written in arial font with size 11pt and spacing between lines of 1.5 lines, and a single column, with a 2cm margin around A4 pages.

The report should also follow the following structure:

- (2%) **Introduction** (up to 0.5 pages of text)

Summary of the approach followed with identification of the problem, technologies used, some special characteristics of your solution, some results that you want to highlight and structure of the report. For this to fit in half a page, they will have to be succinct. Here it is only interesting to give a quick idea of what they did and what they will show next.

- (7%) **Structure of Code** (up to 1.5 pages of text, excluding images)

Description of the (static) structure of the code using class and / or object diagrams. The text should highlight the main components of the code and refer to the respective diagrams.

- (7%) **Code Operation** (up to 1.5 pages of text)

Description of the (dynamic) interaction between the elements of the code using interaction, activity and / or state diagrams. Once again, you should use the text to explain some more elaborate interactions or some details that you want to highlight, always referring to the respective diagrams.

- (7%) **Implementation** (up to 1.5 pages of text)

Justify the choice of technologies used, describe the way used to map classes in each technology, describe the high-level architecture of the program (infinite cycle vs threads, etc ...), explain how to invoke methods between ERP, MES and PLC (Modbus variables, OPC-UA, etc …)

- (5%) **Final vs Initial Architecture** (up to 1.5 pages of text)

Discuss the main changes introduced between the solution initially proposed and the one that was implemented.

- (7%) **Comparison with Common Design Patterns** (up to 1.5 pages of text)

Critical comparison of the architecture used in the project with the standards (design patterns) proposed by the ISA 95 standard, RAMI 4.0, and the article "Service Granularity in Industrial Automation and Control Systems".

- (5%) **Team Organization** (up to 1.0 pages of text)

Describe the assignment of tasks within the team, the development model, critical comments on the performance of team members and any changes made to the assignment of tasks. Indicate how the group wants the note to be distributed to each member of the group

- (5%) **Testing and Results** (up to 1.0 pages of text)

Describe the tests performed, in particular what complex situations were tested. Performance can also be quantified (e.g., order XXXX took xx seconds to be processed; sequence of orders X, Y and Z took yy seconds ...). Critical appreciation of these results.

- (5%) **Conclusions** (up to 1.0 pages of text)

Summary of what was said in the report, highlighting some results considered more relevant.

8 Competition

A competition will be launched at the end of the semester that will be won by the group that manages to complete a certain sequence of client orders for the lowest cost. The group that wins the competition will have a bonus of 10% in the project grade. The group in second place will have a bonus of 7.5%.

The sequence of client orders that will be used in the competition will be delivered after the project's delivery deadline. The sequence will be made available in the form of a script so that orders are launched at predetermined times (and not all at once at the start of the program).

Groups interested in competing must submit the parameters (total cost incurred). The winning groups will also have to make a live demonstration in a public session.