Implementing industry 4.0 on digital twin with real time production data processing

Geert Caberg[1], Daniel Fankort²

[1]Geert Caberg, Belgium, geert.caberg@student.pxl.be
[2]Daniel Frankort, Belgium, daniel.frankort@student.pxl.be

Keywords – Industry 4.0, Smart Factory, Glovia ERP, Digital twin, KPI analysis

Project coordinator: Hannelore Dierickx and Wim Vandormael
Project owner: Patrick Pilat

Abstract

For a research project on industry 4.0, the PXL research team has built a “Smart Factory” production line to display the different functionalities of a smart and flexible production.

This research, will focus on how the obtained knowledge on Industry 4.0 can be shared and implemented in other production facilities. For educational purposes, a digital twin is created within Factory IO which has a similar connection to the Glovia ERP software and the PLC. In addition Microsoft Power BI is used to showcase important key performance indicators (KPI) from the production.

Results have shown that, although Factory IO cannot recreate the full PXL “Smart Factory” production line, it is possible to create a more basic production which will provide a good understanding on the different connections between systems. This digital twin is a great educational tool to understand how production lines can be converted for smart and flexible production.

1 Introduction

Industry 4.0 is an upcoming movement within the manufacturing industry which strives to connect all machinery and data to create a factory which is “smart” and “flexible”. The PXL Construction & Energy research team, commissioned by Govaerts Recycling, has conducted research on how to link the enterprise resource management package (ERP) “Glovia” to the production machines.

To educate students and professionals on smart factories and industry 4.0, the request was to create a fully functional digital twin of the “Smart Factory” production line that is built in the PXL campus in Diepenbeek. The goal is to virtually recreate this machine, its functionalities, and the connections between the different systems. It was suggested to use the software package Factory IO, a 3D factory simulation, as this is commonly used for learning automation technologies. In addition it was requested to conduct a research on how the key performance indicators (KPI) of the production process can be visualized using Microsoft Power BI.

During the design and creation of the digital twin, it was decided to work according to the six sigma philosophy. When professionals opt to upgrade or renew their production to the principle of Industry 4.0, this research can be used as a reference. It is therefore important to ensure a clear and orderly structure. Since the six sigma philosophy is already widely represented within the professional world, often even down to the level of the general business philosophy, it is advisable to proceed according to this line of thought. Throughout the entire research, we have always worked according to the DMIAC principle, which stands for define, measure, improve, analysis and control.

The hypothesis is that when using the 3D simulation software, Factory IO, there will be some limitations that may affect the final result. As a result, it is possible that an exact copy of the “Smart Factory” production line cannot be made and alternative solutions will have to be found. Looking at visualising the KPIs, a major factor for success will be how the existing data will be prepared and processed.

2 Method

The research is separated in five phases. The first phase consists of a literature study where all the new tools are studied. The results of this research are used as a basis for phase two, in which the digital twin will be designed. In phase three and four, the design will be transformed into a fully functional system where all the connections are made between the different systems. Finally in phase five, a Power BI program will be created to visualise the process data.

Phase one – Literature study: Due to the fact that Power BI and Glovia are intricate packages, they will require time to learn. To learn Power BI, online courses will be followed to grasp the basic functionalities, afterwards the data from the physical pre-existent machine will be visualized. To learn Glovia, videos recorded during lectures were filtered and used to grasp the functionalities of the package and how to use it.

Phase two – Design of the digital twin: When a good grasp of both Glovia and Power BI is acquired, the focus will shift to realizing the digital twin. The digital twin will be created within a software package named Factory IO, which is commonly used to simulate 3D factories when learning PLC programming. The goal is to create a digital twin that mimics the pre-existent machine as closely as possible. Since Factory IO will have certain limitations in available production elements, it is important to find a good compromise between the functionalities

Phase three – Writing the PLC program: After finishing the design in Factory IO, the production line has to be programmed. For this a virtual Beckhoff PLC will be used which is programmed in Twincat Studio. Since this project will be used as a reference in the future, it is of the utmost importance that the PLC code is clear and structured. The decision is made to work according to the same programming agreements as used in the PXL research teams project.

Phase four – Connections: Within the Glovia ERP system, new workorders for production and maintenance cycles need to be created. To send these workorders to the production line and get an update on the status, a two way connection needs to be established. A second connection is needed to log all the production details in a dedicated event log which will be stored in a MySQL database. For this, knowledge about MySQL and Node-Red will have to be acquired. Since similar connections were already established in the previous research, much information can be gathered from there.

Phase five – Power BI: As soon as the connection between the production line and the MySQL event log database is established, the data can be analysed to calculate the most important KPIs. One of these KPIs will be the overall equipment effectiveness (OEE) of the system. Since this is a significant and important calculation, it is important to determine how this value can be correctly determined. After determining the desired KPIs to visualise and having ensured that sufficient data is available for these calculations, the program can be created.

3 Basics and Related Literature

A lot of the tools required to bring this research project to a success, require some primary research. In this section, a summary of the acquired information will be discussed.

3.1 Fujitsu Glovia G2

Glovia G2 is an ERP package developed by Fujitsu which allows for enterprise management and real-time insights and statistics to be followed and inspected [1]. At the beginning of the project, several videos which were recorded during a Glovia course, were provided by the project leader of the PXL research team.

Due to the intricacy of Glovia and the many options it offers it was decided to focus on only learning the basic features which were required. Namely creating sales orders (SO), work orders (WO), service work orders (SWO) and releasing work orders. A sales order is an order made by a customer, thus if a customer orders a product, a sales order will have to be created within Glovia. Subsequently a work order can be made out of this sales order in the project resource planning (PRP) screen of Glovia. Afterwards these work orders are to be released into the system. These new sales- and work orders will be created into the pre-existent Glovia setup. The machine will know what to produce based on the order's item name which is given within the work order. These items are created by renaming a duplicate of a pre-existent item from the Smart Factory project.

3.2 MQTT Brokers

In this project an MQTT broker is utilized, this is software hosted on a device such as a Raspberry Pi. It is used to communicate between different devices and sensors. The broker used in this project is "Eclipse Mosquitto", which is a lightweight open source message broker [2].

The way brokers allow for communication between devices is through a publish-subscribe pattern.

One or more devices can publish process data to a "topic" on the broker. A topic can best be described as a free space where data can be stored. When other devices need this data for their process, they can subscribe to this topic. If there is new data available, this will be pushed towards the subscribed devices.

3.3 Databases: MySQL and Microsoft SQL

Two types of databases were used in this project. Glovia uses a Microsoft SQL database to store all the data, whilst a MySQL database is used to store any process related data such as the event log. Both types of SQL databases are very comparable and use the same code. A database is a repository of data which consists of tables. These tables will contain rows which hold data that is sorted under different columns.

To obtain data, different SQL specific commands can be used. These commands can range from fairly simple to long and complex commands. For this project, only a few, simple commands are required. There is the "SELECT" command, which selects certain data from a table. This command often requires the help of specific filter-commands such as the "WHERE" command. This helps the main command to select data which complies to certain criteria. Secondly the "ORDER" command is used for ordering the data according to one of the variables within the table. Finally another main-command "INSERT" is used to add a new row of data within a table [3].

3.4 Node-RED

Node-RED is a programming tool used to virtually connect various hardware devices, Application Programming Interfaces (API) and online services with each-other [4]. The software will set these links by connecting nodes together with wires, creating projects which are called "flows". As Node-RED does not require high specifications from the system where it is executed, it can run on several small and low-cost devices such as a Raspberry Pi or an Android-powered device. In addition, Node-RED allows for JavaScript functions and code to be executed, however a selection of many useful functions is already present within a function library.

The flow from the Smart Factory project shows that the connections within Node-RED are fairly basic. First, Node-red will have to subscribe to the correct MQTT topic with the broker so that it will receive an update as soon as new data becomes available. Secondly this string of data needs to be reformatted through a JSON node. Thirdly, a piece of JavaScript code is used to change the incoming data to a MySQL insertion command. Lastly, this command will be sent to a MySQL database which inserts the data into the correct table.

3.5 Factory IO

Factory IO is a software package developed by Real Games, which is used to 3D simulate automation processes [5]. This package can be linked to actual and virtual PLCs, so that the designed production processes can be programmed and automated. This package will be used to create a digital twin. Besides replicating a working production line, it is also necessary that random errors can be simulated. Thus, the software had to be researched to determine to what extent the production line of the "Smart Factory" project could be replicated.

It was learned that within Factory IO there were some major limitations which would impact the final result. A first limitation relates to the actions that can be performed within a process. For example, it is not possible to perform a drilling or milling operation [5], as is present in the "Smart Factory" process. Secondly, the simulation of errors is only limited to manually putting elements in a state of being broken [5]. Besides the issue of only manually activating an error, it is also not a useful type of error for this project. A component in "error" state will not perform any action, whilst within this project it is mainly necessary to simulate incorrect actions.

Additionally there is also a limitation in products which can be used to perform any sort of processing action [5]. It was learned that only so-called "products" could be used. These are available in three basic forms, two colours and two material types. The basic forms were an unprocessed "raw" block of material, a pre-processed base and a covering lid. The material of these basic forms can be plastic, which is available in green or blue, or metal which is only available in a grey, metal colour.

Picture 1: Examples of the products [5]

3.6 PLC Beckhoff - Twincat

Since this research will be used for educational purposes, the decision is made to work according to some programming agreements. For the determination of the map structure and the structure of the code, the PLC program of the “Smart Factory” production line was examined in advance. Additionally the required function blocks required to establish communication from the PLC were also examined.

As the program for the digital twin will be large and complex, a few ways of standardizing the program are researched in order for the program to be easier to alter and read. First, the programs of the different process stations are split up and sorted into respective folders. Additionally “Actions” are used within the program organization units (POU) to compress long lines of code and distribute them more clearly. These actions are subprograms which are executed when called upon. Lastly, to make the code easier to read, “Enumerations” are applied where possible. These will especially contribute towards a better readability in case steps.

In order for the PLC to communicate with databases, certain function blocks are to be used. All of the required function blocks are found within the Smart Factory program. Reading data from the Glovia database requires the “SQLDatabaseEvt”, “SQLCommandEvt” and “SQLResultEvt” function blocks, these are respectively used to connect to the database, create the read command for the database and to save the result of the command in an array within the PLC. Writing to the Glovia database requires the “DBRecordInsert_EX” and “ND_StartProcess” function blocks. Firstly, the “DBRecordInsert_EX” is used to write the scrap value of the process into the Q_SFWOAS table and the produced value into the Q_SFWCMP table of the Glovia database, this is done with a delay of 2,5 seconds in between. Afterwards the “ND_StartProcess” function block is used to run two batch files on the server that hosts Glovia, these write the data from the priorly mentioned tables to the WO table.

3.7 Microsoft Power BI

Besides the request of creating a similar, virtual production line which bears similarities to the “Smart Factory” production line, there was also the request to research if it is possible to visualise important process data based on data which was already available. The preference of the project-owner went to using “Power BI”, a software package by Microsoft which makes it possible to process raw data and turn it into a visualisation [6] which can show the various KPIs of the production process. This software package is comparable to Microsoft Excel, but distinguishes itself with its thoroughgoing functionalities and the possibility to process large quantities of data in a short time period. The basic functionalities of Power BI were learned by following a Udemy course by Maven Analytics [7] and online research for the more specific functions [8] [9] [10] [11] [12].

The data used in the analysis of a Power BI program can come from many different sources such as MySQL databases, webpages or Excel worksheets. This project will derive the data from a MySQL database. To prepare the data and create extra columns or tables, Power Queries and DAX code can be used [13] [14]. Power Queries are pre-programmed actions within Power BI which can be used when only simple equations need to be performed. For complex actions, DAX code will be the better option. These are manually written lines of code which resemble Excel formulas and Visual Basic code. This same DAX code will also be used when creating measures, which are momentary calculations that return only one value. These measures are used to eventually calculate the predefined KPIs.

The different pages where information can be visualized in Power BI are called dashboards. Determining how and which information will be shown in the different dashboards is crucial to the quality of the program [15]. These dashboards need to be uniform, clear and easy to read. It is therefore recommended to clearly define in advance how the program will be used and which data is crucial for a good analysis.

Within a production environment, there are some common KPIs which are generally used in a process’ analysis. For example, when considering the turnaround time of orders, reoccurring KPIs are order turnaround time, the total waiting time, average production time and the ratio between successful and failed cycles. Within industry 4.0 it is also very important to make a good and detailed analysis of the errors that have occurred. A trending from occurred errors can help technicians identify potential component failure early on, which will help in preventing production losses.

Lastly, there is one KPI which provides a great indication of the current state of the process, the OEE. This value will provide an important insight on how well your production process performs. The goal is to determine how much time the process is truly productive within a period of time. E.g. an OEE score of 100% means you are manufacturing only good parts, as fast as possible and without any losses [16].

Picture 2: Overview of the OEE calculation [16]

To go from the total time available to the fully productive time, all possible losses must be taken into account. The first loss is the “Scheduled Loss”, this is all the time were there is no intention of running production, E.g. plant shutdowns, lunch breaks, time in which there is no order or planned maintenance. Thereafter the “Availability Loss” is taken into account. This includes any event that stopped the production for an appreciable length of time, long enough for an operator to log a reason. Lastly, the “Performance Loss” and “Quality Loss” is subtracted. The performance loss accounts for anything that causes the manufacturing process to run at less than the maximum possible speed. Quality losses are equal to any losses resulting from manufactured parts that do not meet quality standards [16].

4 Results

This section will discuss the results of the five research phases which were discussed earlier in the method.

4.1 Phase one – Literature study

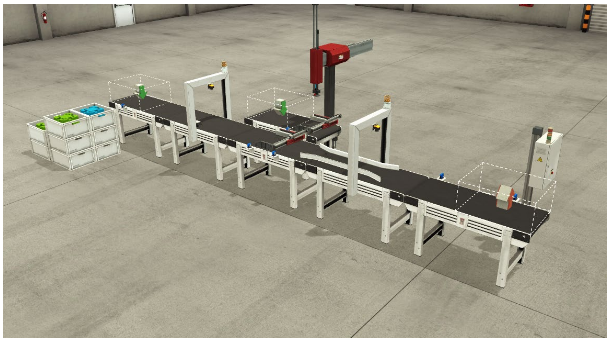
The online course which was followed showed how Power BI works and what the basic functionalities were. Although the online course focused on processing and analysing financial data, many opportunities were seen on how this could be used to visualize certain KPIs. The existing “Smart Factory” production data was used to create a first test program. Here the basic functionalities were transformed into a fully functional Power BI program that displayed all the KPIs which were requested. For the dashboards in the visualisation, a uniform layout was designed based on the corporate identity of the PXL college university. The goal was to create a simple and easy to read template based on the black and white color scheme used by the PXL. The result of this test program and the designed dashboard template are later used as a fundamental basis for the creation of the Power BI file of the digital twin.

To learn Glovia, many videos from courses were followed to develop a know-how for Glovia. However it turned out that the package was too complex to learn in such a short time period. Setting up Glovia requires experts who have experience with the software package, this is due to its extremely broad functionality. Thus in this project a member of the PXL Construction & Energy research team, who already acquired expertise on this software, helped out with the creation of the Glovia setup. The Glovia setup of the Smart Factory machine was utilized as a basic fundament, however to create sales and work orders to queue production within the digital twin, a new item was made. As an end result nine items were created within the Glovia system. Four of these items were utilized to order products whilst the other five were used to plan maintenance tasks.

4.2 Phase two – Design of the digital twin

Phase two focused on the creation of the digital twin. The goal was to mimic the “Smart Factory” production as closely as possible.

After conducting research on the functionalities of the Factory IO 3D factory simulation software, the conclusion was made that there were some limitations in the software which have certain consequences in the final design of the production process. After a thorough analysis, the design was made as shown below [Picture 3: Final design of the digital twin].


Picture 3: Final design of the digital twin

This production is divided in five process locations, each having their own specific task.

“EMIT” station: This is the first process location which is in charge of supplying the base products. This is comparable with the operator who will place new raw materials on the production line in the "Smart Factory" setup.

"CAM1" station: Next the emitted product will pass a vision camera, which will check the product's colour. Additionally a capacitive sensor checks the product's material. This action is quite similar to the first control station of the "Smart Factory" production line, where it is checked whether the supplied material has the right colour. The material check is unique for the digital twin and is only used to obtain one extra possible error situation.

"ROB" station: During this step, the base material will receive a lid. It is noteworthy that this action will only be executed if the result of the CAM1 check is positive. Even though this is a fully virtual production line which will only be used for educational purposes, it is important to still design with a view on sustainable productions in which waste is limited as much as possible. This step is chosen as an alternative for the operating action. Although this action is only comparable with the placement of the candle in the "Smart Factory" process, it can still produce similar data to the drill and mill actions of the "Smart Factory" process.

"CAM2" station: Second to last there is a second controlling station. This station will check whether the lid, which is placed upon the base product, has the correct colour. Again the attempt was made to mimic the similar station in the "Smart Factory" production where the products get a final check to determine if all of the process' actions were executed correctly.

"REMO" station: At the end of the production line a remover station is added. If a product is marked as "scrap", the product will be removed immediately. When a product was created successfully, the operator will be asked to make the decision whether the product is effectively correct or if it should still be disposed as scrap.

Since Factory IO also has some limitations on simulation errors, the decision was made to include these in the corresponding PLC code. This will be explained in phase 3 of the results.

4.3 Phase three – Writing the PLC program

4.3.1 Programming agreements

After finishing the design in Factory IO, the PLC code has to be written. To ensure that the code of the program is clear with a logical structure, the decision was made to work according to some pre-determined programming agreements.

A first agreement was to create a separate program for each process location in the production process. By separating these programs, long sections of code will be avoided, creating a program which is easier to read. Within these separate programs, actions are used to extract long sections of code, dedicate to specific or reoccurring tasks and exclude these from the main program. These actions can be called upon when needed in the main programs.

Secondly, enumerations are used where possible. By using this functionality, many objects within the code are easier to read. Especially with use of cases. The addition of an enumeration as a replacement for the case steps, can help with navigating through the different steps of the process quicker. The enumerations will also contribute to a faster understanding of the code when a technician wishes to change or add elements afterwards.

Finally, a uniform map structure is used which has been carried through to the full construction of the project. For example, this folder structure has been used under "DUTs", "GVLs" and "POUs". This makes it easy to determine where certain data of a specific process element can be found.

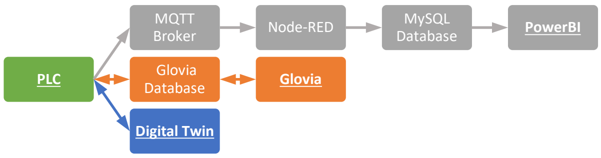
4.3.2 Error simulation

Since factory IO has some limitations with regards to the simulation of errors, it has been decided to build in possible errors within the PLC code. These errors will be called upon at random. The magnitude of the errors can be set in the PLC program, which allows the user to control how successful the production round should be.

There are two types of errors built-in. First there are errors which are product-related such as a wrong colour or material. These types of errors can occur on the base part as well as on the lids. Secondly, there are time-related errors such as waiting times and process delays. All possible faults which are processed in the PLC program, are thoroughly explained in a separate video which clarifies the full PLC code.

4.4 Phase four – Connections

To establish communication between Glovia, the PLC, the digital twin and Power BI, several different connections had to be established. To give a better understanding on how all these systems were connected to each-other, a network tree was created [Picture 4].



Picture 4: Communication tree

As can be seen on this image, the PLC will serve as a central organ where there is a connection with three separate branches. This allows for easy processing and distribution of data within the PLC.

The first branch: The first branch, dedicated to the event log of the process, is the only branch in which data moves in one direction. At the start, the data is sent to the MQTT broker. Since NODE-Red is subscribed on the topic of the MQTT broker, it will immediately receive any new information. This new information is then forwarded and inserted into the MySQL database. This allows for Power BI to retrieve this data and process it to execute the necessary measures. It was found that the MQTT broker and Node-RED were not vital to write data to the MySQL database. Within the PLC there are certain functionalities which enable the PLC to have a direct communication with the MySQL database. Nevertheless, the decision was made to use an MQTT broker instead of creating a direct communication. This decision was made since MQTT is a technology which was never investigated before by the researcher and there was a desire to test the capabilities an MQTT broker would add to the system.

As the process of the created digital twin is very fast, an additional buffer had to be implemented to prevent data to be lost in transition. This buffer will send one line of the event log's data every 200ms, making sure every line of data is registered.

The second branch: In this branch the connection with the Glovia ERP system is established. First, the PLC will import workorders (WO) from the Glovia database. As soon as the WO is finished, feedback will be given to Glovia by the PLC. This feedback contains information on the amount of scrapped products and amount of correct products which have been produced. These WOs will then no longer appear in the operator's interface.

For flexible production processes, Glovia provides the functionality to enable a "product configurator" when a customer wants to order a specific product. This very interesting when there are many variations possible of one product. The process created in this research is an example of flexible production. However, because of the limitations within Factory IO, there are only four different product compositions possible. Therefor the decision was made to not use the "product configurator". Instead four different products were created with specific item names. The items created in Glovia were as follows:

- GOV-BB: Blue base with blue lid
- GOV-GG: Green base with green lid
- GOV-BG: Blue base with green lid
- GOV-GB: Green base with blue lid

When a WO is sent to the PLC, the system knows, based on the specific item names, which product compositions have to be produced.

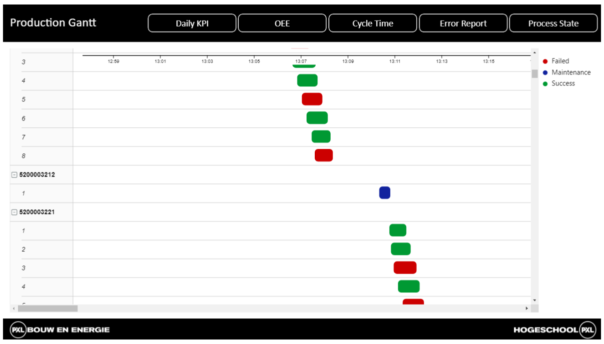
It is important to know that the communication to send feedback back to the Glovia system is slow, therefor extra time needs to be accounted for whilst writing data to the database.

The third branch: This is the branch where the connection to the digital twin is made. In this branch the PLC can control the digital twin and is able to tell it to produce the requested products from the imported work orders.

Overall the communication worked very well. However with fast production processes it might be possible that writing data requires some extra attention. As data-transmissions will be done much faster than data-writing, this will result in the loss of data. Therefore it is advisable to carefully consider how data will be written to the databases. To make sure data would come through, these data transmissions were tested with the digital twin running at a 400% of the normal speed.

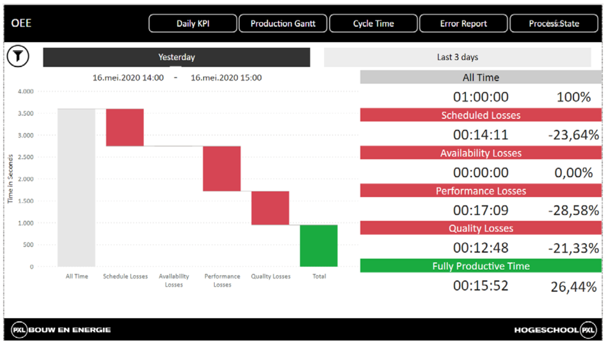
4.5 Phase five – Power BI

Before developing the Power BI program for the digital twin, all the required KPIs were determined. Since it is important that the various KPIs are presented clearly and in a structured manner over the various dashboards, a clear layout was developed in advance. The decision has been made to work with different dashboards, each of which will display a composite group of KPIs aimed at a specific part of the analysis. The first dashboard will specifically focus on the process' data of the current production day. This data will be continuously updated during the production process. On a second dashboard, a full size Gantt chart is used to provide a visual representation of the progress of production. When was production carried out, which orders were these, what was their result and how do they relate to each other in function of time.



Picture 5: Preview of the Gantt chart in Power BI

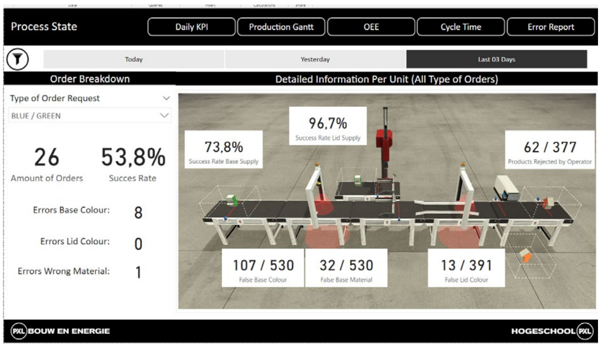
The calculation of the overall equipment effectiveness (OEE) is located on the third dashboard [Picture 5]. This is presented both numerically and visually with a graph. Since this KPI is very important when analysing a system, the calculation of it must be very precise. During this calculation, different situations that may occur have always been taken into account. For example, production cycles that overlap a working day are analysed according their effective influence on the OEE of the selected period. This determination in particular proved to be very difficult in practice. To clarify how this complex calculation has been conducted, a dedicated section is provided within the created Power BI manual.



Picture 6: Preview of the OEE analysis in PowerBI

To get a better insight on the cycle times, a fourth dashboard is added where all the KPIs regarding production cycle time are included. Here information like order cycle times, production results and a trending on waiting times is presented. This information is based off of data from finished cycles and can be individually filtered to specific periods.

The fifth and sixth dashboard can be combined for a thorough analysis on the occurred errors. Here an overview is given on the errors which were detected and how these errors behave over a period of time. One dashboard is focused on providing a trending of the reoccurring errors. This can help technicians identify potential component failures early on, which will help in preventing production losses. The other dashboard will provide a more visual approach on where most errors occur [Picture 6]. This page will provides the data as an overlay on an image of the production unit. As before, it is possible to filter on specific periods, but also on specific order requests. This has been done to have an analysis where certain failures can be linked to a specific order request.



Picture 7: Preview of the second error analysis dashboard

As there are many calculation within the Power BI program, the decision was made to have a separate manual where all the used measures, tables and presented KPIs are explained in further detail.

4.6 EDU version

After the original digital twin was finished, the proposal was made to develop a second version, aimed at educational purposes. The idea was to create a program that can simulate a full production day within one hour, without any manual interactions. This makes it possible to start the system at the beginning of a workshop. Whilst the production is up and running, the teacher can then take a theoretical approach to analysing production data. When at least one hour has elapsed, a practical analysis can be performed based on the data generated by the simulation.

To create such a program, some adjustments were needed within the PLC code and the Power BI program. For example, within the PLC program the connection with the Glovia ERP system is removed and replaced by an extra piece of PLC code which ensures that new orders are continuously generated. Different production orders and maintenance orders are generated completely at random. The adjustments that had to be made within Power BI were mostly related to the time-related calculation. Since a full production day is now equal to one hour in reality, some additional challenges presented themselves. Particularly when drawing up the relationships between different tables.

5 Conclusion

In this research, the goal was to create a fully functional digital twin of the existing "Smart Factory" production process which was created in previous research on Industry 4.0. After a thorough literature review on the different tools used, a digital twin was created in the 3D factory simulation program Factory IO. Because of certain limitations in the software tool, it was not possible to recreate an exact copy of the existing production line. However, after a thorough analysis, a production line has been designed which closely matches the process that is carried out within the "Smart Factory". In addition a connections with the Glovia ERP software and a MySQL database, used for logging the process data, were established. To get a realistic simulation with randomly occurring errors, extra lines of PLC code were added to simulate several different errors or time related losses. Lastly a Power BI program was created which accurately visualises the different, requested key performance indicators (KPIs).

Since it was possible to recreate a digital version of a smart production line, it can be concluded that all the information which is present from the previous research, conducted by the PXL research team, is sufficient to use as a reference to optimize old systems or even create completely new processes. For the Power BI program there was no existent information. However, the programs and accompanying documentation from this study can be used as a reference in the future.

Although machine learning is also an important element within industry 4.0, it was not possible, due to the limited time, to implement this into the digital twin. A separate document describes how this could possibly be implemented within the existing project. These recommendations can be used as a basis for subsequent research.

Acknowledgements

First and foremost, we would like to thank our project owner, Mr. Pilat, for providing us with the tools necessary to conduct this research. In addition, we would like to give our special thanks to Miss. Dierickx and the full PXL – construction & energy research team for their continuous contribution of knowledge and assistance to bring this research to fruition.

References

[1] Fujitsu, "Glovia," Fujitsu, [Online]. Available: https://www.glovia.com/.

[2] Eclipse Mosquitto, "Mosquitto," [Online]. Available: https://mosquitto.org/.

[3] w3schools.com, "SQL Tutorial," [Online]. Available: https://www.w3schools.com/sql/.

[4] Node-RED, "Node-RED," [Online]. Available: https://nodered.org/.

[5] R. Games, "Factory IO," Real Games, [Online]. Available: https://factoryio.com/.

[6] Microsoft, "Power BI," Microsoft, [Online]. Available: https://powerbi.microsoft.com/nl-nl/.

[7] C. Dutton, "Udemy," Maven Analytics, 3 2020. [Online]. Available: https://www.udemy.com/course/microsoft-power-bi-up-running-with-power-bi-desktop/. [Accessed 03 2020].

[8] A. Saxton, "Make navigation easier with Power BI buttons," Guy in a Cube, 24 03 2020. [Online]. Available: https://www.youtube.com/watch?v=nCEMM6Ryees. [Accessed 03 2020].

[9] Microsoft, "Data Analysis Expressions (DAX) Reference," Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/dax/dax-function-reference.

[10] P. Leblanc, "Power BI Tutorial: Dynamically Filter By Today's Date," Guy in a Cube, 08 08 2018. [Online]. Available:

https://www.youtube.com/watch?v=lkHFpmA4SJ4. [Accessed 03 2020].

[11] S. K. Anandan, "Time Period Slicer for Last 7 Days, Last 30 Days,...," Analytics Avenue, 7 05 2016. [Online]. Available: http://analyticsavenue.com/power-bi-timeperiod-slicer-for-last-7-dayslast-30-days/. [Accessed 03 2020].

[12] B. Elite, "Organize Measures with Measure Table," BI Elite, 6 03 2018. [Online]. Available: https://www.youtube.com/watch?v=6IHNYDy_Ynw. [Accessed 03 2020].

[13] P. LeBlanc, "The How and Why of Power BI Aggregations," Guy in a Cube, 4 03 2020. [Online]. Available: https://www.youtube.com/watch?v=EhGF372t0sU. [Accessed 03 2020].

[14] P. LeBlanc, "Configure Power BI Aggregations," Guy in a Cube, 11 03 2020. [Online]. Available: https://www.youtube.com/watch?v=kytYzjTXUHs. [Accessed 03 2020].

[15] A. Sinh, "Top 5 Power BI Beginner Mistakes," PowerBIPro, 12 07 2018. [Online]. Available: https://www.youtube.com/watch?v=PnPP-46oolc&t=386s. [Accessed 03 2020].

[16] Vorne, "OEE made easy," Vorne, [Online]. Available: https://www.oee.com/oee-factors.html.