Int j simul model 17 (2018) 1, 30-41
ISSN 1726-4529 Review scientific paper

USING 4-LAYER ARCHITECTURE TO SIMULATE PRODUCT AND INFORMATION FLOWS IN MANUFACTURING SYSTEMS

Saez-Mas, A.; Garcia-Sabater, J. P. & Morant-Llorca, J.
ROGLE, Department of Business Organisation, Universitat Politècnica de València,
Camino de Vera s/n, 46022 Valencia, Spain
E-Mail: adsems@upv.es, jpgarcia@omp.upv.es, joamollo@upv.es

Abstract
This work illustrates the application of novel simulation architecture with two case studies where the proposed architecture, the so-called 4-layer, allowed us to address the complexity of the analysed systems. The fundamental objective of this work is to show the structure of layers, how layers interact with one another and with the user, and what benefits this separation proposes. The first case study deals with moving car bodies from the paint plant to the assembly line through a sequencing system that involves distributed decision-making processes in an ASRS. The second case study focuses on analysing a layout of a section used to assemble the engine and transmission set, where the quality of the material flow is evaluated. The work highlights some of the advantages of modelling with 4-layer architecture, and explains the key processes that connect different elements.
(Received in June 2017, accepted in December 2017. This paper was with the authors 2 months for 2 revisions.)

Key Words: Discrete Event Simulation (DES), Material Handling System (MHS), Manufacturing System, Automobile Assembly Plant, Simulation Approach

1. INTRODUCTION

In today's changing manufacturing world with its new paradigms like mass customisation and global manufacturing operations and competition, companies need better capabilities to respond more quickly to market dynamics and varying demands [1]. This is a challenge, especially for increasingly complex industries, such as the automotive industry, where it is difficult to adapt the production plant to demand requirements while its operations continue to be efficient [2].

The logistics in an industrial enterprise covers all the activities related to material flows, including the material flow from suppliers, warehousing, the material flow in production, delivery and distribution. To summarise, logistics can be understood as a way to manage these activities that contributes substantially to the enterprise's efficiency [3].

The management of the materials flow through trolleys and other moving elements is not an aspect that is usually considered while designing layouts. Not only is it difficult because of its dynamic nature, but it is also considered of little relevance in classical assembly line manufacturing environments, where more interest is shown in machine capacity than in the previous and subsequent movements of both future vehicles and their components. However, the variety of information interaction and material transportation in current Manufacturing Systems (MS) [4], which involves up to five car models on the same line in automobile assembly lines, hinder the analysis of this relevant flow. For that purpose, the development of a virtual layout using simulation modelling is worthwhile during the decision-making process of MS reconfigurations [5].

Likewise, the system's complexity requires the coexistence of data with a high level of uncertainty and a set of possible scenarios that must be combined with multiple design alternatives. This requirement favours data-driven simulation, which Pidd [6] defined as "a generic model that is designed to be applied to a wide range of systems that have structural

https://doi.org/10.2507/IJSIMM17(1)408

similarities" to complement those circumstances in which many similar models have to be evaluated [7]. Although data-driven simulation was developed to enhance the ability to simulate alternative scenarios, separating simulation model elements from data provides a step forward in how to build simulation models.

After testing several approaches to solve the problem of analysing the material flow that feeds an assembly line, the need to separate material flow (vehicles and trucks) from different information flow emerged. However, given the complexity and importance of material handling systems (MHS) for their impact on the performance of MS [8], it is necessary to go a step further with data-driven simulation. For this purpose, in this paper we suggest the conception of simulation models as a 4-layer architecture (i.e. network, logic, database and visual). By doing so, we not only abstract the data of the model to allow the handling of a greater volume of data, but also we propose distinguishing the logical processes (those that make decisions) from physical processes (those that “add value”) in modelling. This approach seeks to decompose the problem in order to achieve a better design through facing independently each layer and combining with each other. This way of modelling is suitable for complex problems such as layout (re)design, the management of the material flow or the high volume of data to handle, and favours the detection of problems that otherwise might have been overlooked. It also allows a modular growth of the simulation model, which allows us to make changes in layers without altering the entire simulation model, in addition to reusing the layers in future simulation models.

The purpose of this paper is to describe layers, the connections among them, and the advantages of acting in this way. In order to do so, we identify generic layer connections and present two case studies from the automobile sector where it was necessary to apply this architecture.

This work is structured as follows. In the next section we propose a brief review of relevant works in the literature devoted to data-driven simulation and its applications, as well as other simulation approaches. The third section presents the 4-layer architecture of simulation used and the relationship between its layers generically. The fourth section describes the practical application of this architecture to two case studies in the automotive sector. The fifth section presents conclusion and future research lines.

2. LITERATURE REVIEW

As variability in requirements increases, alternative approaches are needed that cut the time required to develop models and, at the same time, are able to incorporate the increasing complexity of the relationship between stages [9]. Clark and Cash [10] suggest that a model may be considered data-driven “when users can apply the model to different situations by changing input that only requires problem-domain knowledge with a minimal modelling knowledge requirement’’. It also allows a wide variety of scenarios to be created so they can be explored relatively quickly and without having to acquire the level of simulation experience that is normally required [7]. These models can also be fully parameterised by providing data through a set of forms, tables, spreadsheets or templates [9]. So it is not necessary to create different simulation models, except if special logics is not considered in common parts [11]. Many researchers have applied this and other simulation approaches according to different objectives, such as reducing modelling times, automatic modelling, or integrating different systems of the production system into the simulation model, e.g., the MHS.

Lim and Seo [11] propose a generic simulation model that is used to cut modelling times and to make simulation models more reliable by employing parameters to automatically execute simulation models. The generic simulator concept includes extracting the storage of common domain parts, e.g., scheduling logics or the operation rules of MHS in manufacturing simulations. Kim et al. [12] suggest a layout and data-driven generic simulation modelling framework. The framework consists of a layout modelling software called the AutoLay modelling software, and a generic simulation model called AutoLogic. With this framework, an integrated simulation model of production processes and material handling processes in a short time period can be developed. Son and Wysk [13] present a structure and architecture for automatic simulation model generation based on a shop floor resource model and a shop floor control model. The static and dynamic information in simulation is derived from a shop floor resource model and a shop level control model, respectively. They apply it to six MS with material processors, material handlers and AS/RS. Wy et al. [14] introduce a generic simulation modelling framework (AutoLay and AutoLogic-Assembly™) to cut the simulation model build time, which is based on data-driven simulation. AutoLay converts the layout data in the CAD file format into simulation models. AutoLogic-Assembly is a generic simulation model that has been developed to consider generalised logistics-embedded assembly lines. Wang et al. [15] report a data-driven simulation methodology to automatically model a production system to be rapidly modified on an automobile assembly line. The system logic is outsourced through the Application Programming Interface (API) from the simulation software. This confers the model flexibility when faced with different scenarios and production data.

These works pose the problem of how to model Information Systems logics without separating the material flow according to data-driven or similar simulation approaches. However, nowadays it is common to find research works that focus on simulation modelling as a combination of modules. This is due to the complex manufacturing reality, which involves operations planning and control, material handling and warehouse systems, among others. Debevec et al. [16] propose the innovative PoVEIR approach, which creates a virtual factory by combining a virtual physical system and a real information system. They present the developed method and simulation models, which proves practical applicability in small companies with an individual or small quantity production type. Seebacher et al. [17] present a feasible and practical approach to evaluate the efficiency of production logistics processes. To this end, they offer a system that allows communication among the products to be processed, internal transport vehicles, and the ERP-system. Kehris [18] offers a prototype web-based MS simulator that consists in three modules: the simulation module, the groupware module, and the project management and workflow module.

As it can be seen in the literature, researchers use several simulation approaches with different objectives. However, to the best of our knowledge, we have not found other research work that provides a modelling approach to deal with the complexity of material handling systems in the automotive sector. To fill this gap, we propose to facilitate the MHS´s simulation modelling by separating the simulation model in 4-layers (network, logic, database and visual). In the following section we introduce each one of the layers and explain how layers relate to each other.

3. SIMULATION MODELLING

3.1 The 4-layer architecture

Generally, in discrete event simulation models, the decision making that is triggered (or activated) by triggers is embedded in the different elements that outline the system model. In this way, the complexity of decision-making is limited by the ability to program complex routines.

As it is necessary to incorporate mobile elements into simulation, the need arises to generate a system that autonomously governs the flow, for example the movement of forklifts in a plant or a warehouse. The same applies when sequencing processes of units are complicated by the increasing complexity of the range of products to be sequenced or the paths that they must follow.

In practice, one need to control both the flow of product there are at least two parallel operations of planning and control systems, one for products and another for handling equipment, which are not connected to each other. In terms of constructing simulation models, it is difficult to consider the aspects discussed above, so that it arises the need of separating the problem in layers to address the whole problem.

The 4-layer concept is especially relevant in this context as it distinguishes the collection and storage of the information from each domain, and also separates the model's main modules. In the proposed architecture, four blocks are used to construct the model: Network, Logic, Database, and Visual. The layers' idea was suggested by [19] and used in [20]. This paper consolidates the idea and discusses the advantages of using a 4-layer model. Moreover, the paper focuses on the interactions and connections between them. The four proposed layers are:

- Network Layer. This layer includes what is for most users the "Simulation model". It includes the machines, buffers, paths and products to be transformed. Explicitly, we remove the data that controls performance, the logics that makes decisions, and, if possible, the external representation. Isolating this layer from others allows us to concentrate on the layout and the structure of flow, which facilitates using simulation parts in different contexts.
- Logic layer. This layer includes decision-making processes. The different procedures are programmed as standalone using standard packages. These procedures may activate other decision procedures, or even elements in the Network Layer. In a regular factory, this layer would represent the operations planning and control systems, which would include planning and sequencing processes and the material handling equipment (MHE) control system.
- Database Layer. In most operational settings, it may help to consider ERPs as a transactional database that contains all the information needed to perform processes. The Database Layer follows this idea. The Database Layer is a database that feeds simulations with the data needed to perform activities and to make decisions. The information stored is not only the parameters needed for the Network Layer, but also the results of the calculations, and the performance indicators. But building the Database Layer separately requires designing the system carefully in order to avoid including data in the Logic and Network Layers.
- Visual Layer. The Visual Layer is included in most simulation packages with the Network Layer, and this feature eases communication with users. Yet different users in the main organisation might require distinct visual aids. A specific feature of our proposal includes blackboards and other coloured *Andon* elements in simulation. Based on Wallace [21], Andon elements represent states or problems and their locations, which gives engineers the opportunity to intervene and resolve before it gets out of hand. Andon allows colouring machines to inform about their status, but also may illustrate the congestion at a given aisle, the state of a warehouse or the queue's size at the tow trucks station.

The Logic Layer, together with the Database Layer, configures the Information System of a real setting. Meanwhile, the Network and Visual layers are related mostly with Physical Systems. We can conclude that the information and material flows with this configuration are explicitly represented to allow a better understanding of real processes.

3.2 Simulation architecture

The basis of simulation is that our layers architecture is strongly interconnected. The explicit split of layers requires planning the simulation processes and links among layers. Fig. 1 illustrates a representation of the structure and its connections, which are presented hereinafter.

The Network Layer performs physical activities; e.g., machines transform products at a rate that might be defined internally with parameters, but these parameters might also be stored at the Database Layer, which facilitates scenario experimentation. Conversely, most of the results and indicators of a given simulation are stored at the Simulator Level. If transferred to the Database, this information might be used to make decisions based on the plant's actual situation.

In most simulation packages, the Network Layer is capable of making scheduling decisions; e.g., where the next available tow truck has to go. However, the system is much more flexible and robust if such decisions are made by customized procedures. The Network Layer connects with the Logic Layer by sending triggers that inform about the situation, and by requesting and receiving explicit orders. One of the advantages of doing this externally is that the logics might be as complicated as it is in most real settings. Moreover, the effects of uncertainty on executing the process might also be evaluated uncertain can be handled in rather easily manner by simulation models.

The Visual Layer is a key part in the model and it has become a relevant part of any DES package. Normally this is the layer that connects the users to the simulation tool. In explicit terms, using images that look like real elements (the Google Sketchup Library might help to do that) is normally pretty well accepted. However, the Visual Layer might represent other views of the system, such as *Andon* boards. If the Visual Layer is capable of interacting with users, we can add a very interesting feature by, for instance, stopping machines or closing aisles.

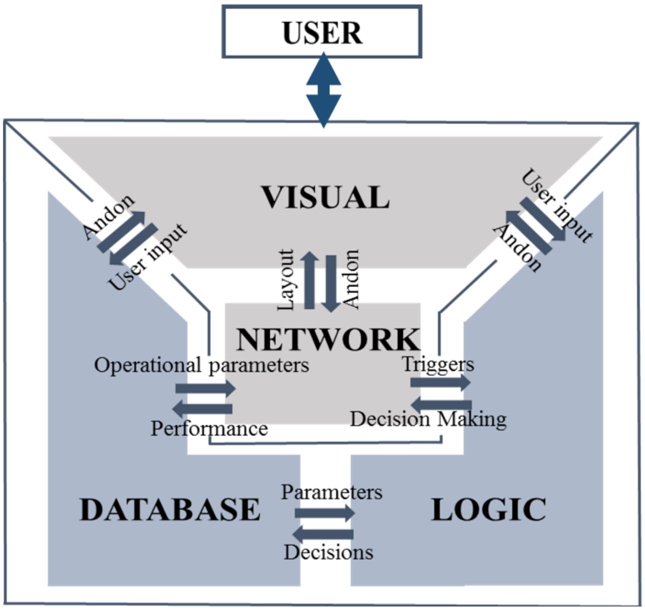

Figure 1: Connections among simulation model layers, users and engineers.

The capability of accepting user inputs through the Visual Layer might also be used to change the parameters at the Database Layer, and not just to show them, and to also answer the specific questions made by the Logic Layer to make decisions and to see their immediate effects.

Last but not least, the Logic layer uses the data stored at the Database Layer to support decision-making processes. The Logic layer needs to know relevant information, e.g., the location of each reference in the warehouse, the assembly sequence, the number of elements travelling in a specific aisle, or the quantity of available stock at the Point of Fit on the Assembly Line. The Logic Layer is also capable of inserting data into the database on the decision made, the obtained results and interactions with users through the Visual Layer.

As we expect most of the sharing to be done through the Database, which happens with ERPs in real settings, and which is the case in such a setting, machines might have their own intelligence (PLC-like systems), and visual boards might be directly filled by users on non-integrated boards. As simulation attempts to replicate reality, the involved team will obtain a much profounder understanding of real processes.

4. APPLYING THE 4-LAYER ARCHITECTURE TO TWO INDUSTRIAL CASE STUDIES

4.1 Automatic Storage and Retrieval System (ASRS) and an extension of the assembly plant

The factory under study has, at the Paint Shop exit and before entering the Assembly Line, a large Automatic Storage and Retrieval System (ASRS), which can hold more than 500 units in four parallel aisles handled by four cranes. The facility is used to manage around 1,700 units on three different small-sized car lines. From this storage device, units have to be delivered to the assembly line in a levelled manner, which is called Heijunka [22], despite the non-regular supply of units from the Paint Shop. Dozens of assembly line stations, but some hundred sequencing cells and external suppliers, work at the pace defined by the sequence [23]. This storage is managed with algorithms, which are designed to work with low complexity products by our research team [24, 25], and were implemented years ago by an external consulting firm.

A change in the global assignment policy forecasted an increase of production up to 2,100 units/day from five different car lines due to the closure of a firm factory. The top management had concerns about the new system’s capability to cope with the new requirements. On the one hand, the original factory that was to be closed was assembling many more complex units, and had an ASRS twice the size to manage its own capacity. On the other hand, some of the new units were much larger and wider, so the current capacity of cells was not enough to store them. Some cells had to be enlarged (not all of them could be).

Furthermore, some variants were going to require a new facility configuration. As the cycle times of some tasks for some units multiplied the assembly line’s actual takt time by 4, a new and parallel facility is to be built to prepare approximately 30 % of the units before sending them to the assembly line. This mix of sequences has implications for the cycle time of some elements of the facilities; e.g., elevator and merger tables. Therefore, these workloads also have to be balanced.

Several issues affect the design process for simulation. As expected, the visual elements of simulation will be helpful as they clarify the focus of the problem. Most decision makers would love to see the factory as it would be, with cars moving in it. Simulating a 500-unit storage with so many different models simultaneously is not feasible in practical. It is impossible for our system to simulate an ASRS of nearly 500 different units at a reasonable speed, each in its cell that was to be sequenced. Therefore, the team had to learn to find visual elements and graphs to help understand ASRS performance.

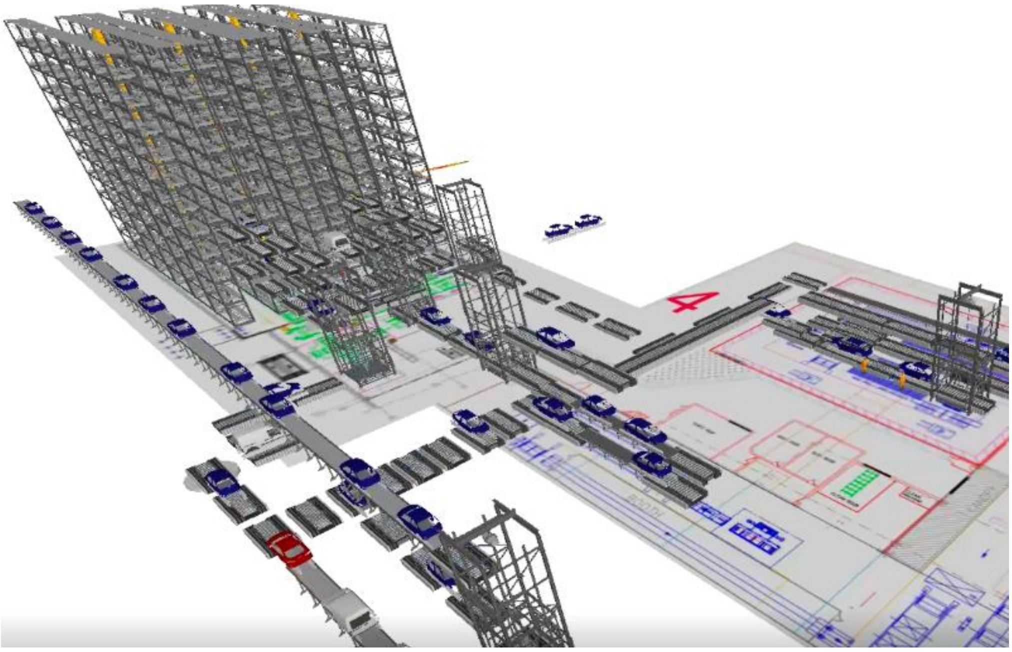

Figure 2: The ASRS and an assembly plant simulation view.

The Network Layer is constituted by transfer and rotary tables as parameterizable elements. Speeds, as well as Mean Failure Times and other parameters are in the Database to facilitate the performance of externally governed experiments. Neither robots nor maintenance elements were implemented as they were not relevant in this simulation. The network is made up of elements that were in the implementation phase, although their definite speeds were not known. The Network was used in later simulation works to improve the operational efficiency by eliminating the Logic Layers as it consumed resources and provided no benefits.

Mechanical engineers wanted to know if the combination of the different elements would deliver the expected Cycle Time. The algorithm that sequences units is the core of the system. Since the system’s utilisation level is very high, it is necessary to thoroughly manage the use of cranes. Yet the Cycle Time was to be achieved only if the sequence of units allowed the used of the resources appropriately. However, its necessity was not clear to most users from the very beginning of the work. Therefore, the sequencing algorithm was created because it was necessary, but unfortunately it was unknown and not considered by staff to evaluate simulation until much later when they realised that it was the core of the simulation.

Having simplified the ASRS, it was possible to accurately simulate the rest of the system: entry and exit transfer tables, elevators, and the whole assembly subsystem. These elements form part of the network layer. Meanwhile, all the decision processes of the elements, except for the most basic, are implemented into the separate Logics Layer.

The most representative aspect of simulation was the ASRS. It is worth mentioning that the ASRS was not simulated in the Network Layer, but used the Logic Layer since the simulation model could not handle this warehouse filled with car bodies. It is impossible for our system to simulate an ASRS at a reasonable speed. Therefore, the first separation was generated. The physical system is modelled with two basic subsystems (this split corresponds to the real separation): storage itself (cells and cranes) and the entry and exit transfer tables. Storage becomes a database connected with the Logic Layer. The network reduces the system of transfer tables that manages input and output. Thus when the Logic Layer requests a unit from the storage, simulation delivers a unit directly to the exit transfer tables and delays it with a simplified queuing system, which considers all the operation times in the ASRS. Units are individually stored at the Database Layer.

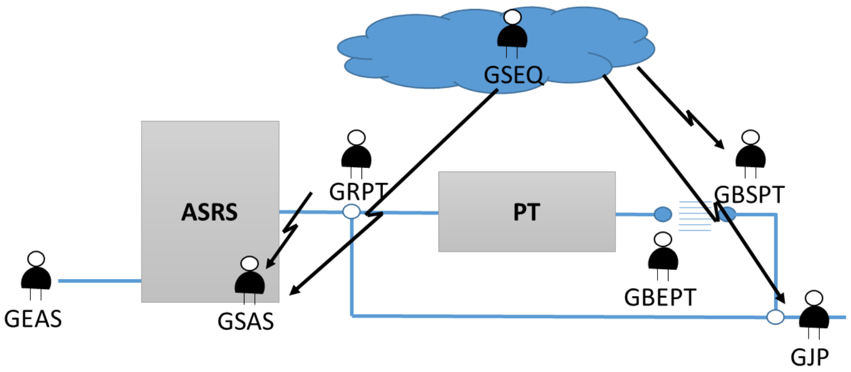

Figure 3: The ASRS and an assembly plant Logics and Network connection points.

The Logic Layer is a set of algorithms (coded in C#) that are activated when requested by the different system elements. Originally, the Logic Layer was a replication of the algorithms found in the real system. The original algorithms centralised the decision at the ASRS output by sequencing the next unit every time, and by assuming that the rest of the system would attempt to produce the predicted sequence following a push strategy. Once the poor performance of such a system was unveiled, the new Logic Layer to be implemented into the real system had to be developed. Thanks to the separation of the Physical and Logic Layers, the new, and much more complex, scheduling system was programmed without changing the rest of the system.

The Database includes four substructures: a) static data about the physical system and the decision rules of end users; b) online data about the status of the different elements (restricted to those variables that could be accessed in real life); c) data about the ASRS content, which replicates the real database structure; d) data about the system's performance and its elements.

The Visual Layer (see Fig. 2) had to fit the facilities according to the design that was already working, and had to convince those who had previously been at the real premises and those who were designing the new ones.

The 4-layer structure, and specifically the explicit separation of the Logic Layer, allows taking a step forward. A new strategy was designed following a pull strategy where the different agents attempt to deliver the best possible unit to the next station to improve the next agent's situation. This same Logic Layer (translated into Java) was implemented into the real system, and today cars are sequenced to the assembly plant using these algorithms, which lead to the user interface being developed. In addition, the network layer was used later in other material flow analysis simulations, but detaching it from the initial logical layer and providing it with a simpler logic, since the previous one consumed resources and was not necessary for the following analyses.

4.2 Layout redesign in an engines and transmissions assembly plant

This second case study focuses on the process of assembling engines and transmissions from a car assembly plant, as described in Saez-Mas et al. [20]. The main problem lies in a layout redesign requirement due to an increase in both the variety and daily production volume of the engine and transmission set. The new system has to be disconnected from the assembly line where it is inserted and the engine and transmission sets are supplied in sequence from a plant located several hundreds of metres away. The new facilities cover more metres, but incorporate a complex system of supply logistics and material flow, which used a wide variety of conveyors, cranes, pallet trucks, tow trucks or forklift trucks, among others. The purpose of the study is to assist the decision-making process for redesigning the logistics supply process. Thus, an attempt was made to validate the different configurations for installations and for programming operations.

It is worth mentioning that, the main concern of the project's owner was not the cost of the operation, but the risk that moving such a large number of handling equipment in such a small space would cause. This safety objective requires developing a set of measures and their corresponding representation to facilitate users' interpretation.

In our model, the Network Layer comprises the layout with its different sections, such as the warehouse, assembly cells, assembly line or picking stations, among others. The Logic Layer comprises the operations planning and controlling system and the MHS. The Database contains data needed for logical operations. The Visual Layer is used mainly to relate to stakeholders through realistic visualisations and *Andon* elements. Fig. 4 shows the new warehouse to be implemented into the production process, and also includes a link to a simulation video.

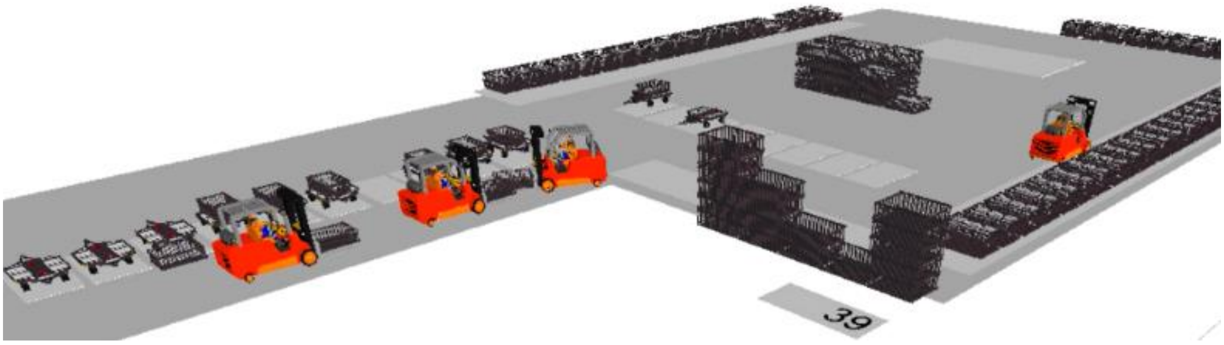

Figure 4: Warehouse simulation view.

The Network Layer is found in the simulation software itself, SIMIO®, which is usual when the discrete event simulation tool is equipped with such functionality. Each element has its own control parameters to activate the triggers that connect to the Logical Layer. In this instance, the Logical Layer, that includes the management system of handling equipment and the operations planning and control are programmed using C# (through SIMIO® API). This external part of the software is responsible for making decisions about achieving the tasks to be performed by elements, and is based on the plant's requirements transmitted by triggers. Once decisions are made, the Logic Layer transmits the execution commands to the Network Layer, where the selected operator and the operator executes the action. This type of modelling allows simulating more realistic systems, where the information system communicates to the operator the orders to be made through screens. The decision about how to route deliveries is not made by the internal logic but by the trolley itself since in fact it is the trolley itself that knows the facilities and decides the way to take.

The Logic Layer is able to read all the necessary information from the database; i.e., from both the requirements side (demands, sequences, bills of materials) and the situation of the different elements (actions taken, status of resources etc.). Furthermore, as well as the logic layer, it was decided to remove the database from the simulator given the complexity of the system and thus simplify the simulation model as such. This separation allows hosting the data as an ERP structure, as is the case in the real facility. MySQL is the application that supports the database layer.

The relationships between layers are generally given by parameters. The Network Layer transmits to the Visual Layer through parameters and state variables, the information to show during a certain period. This relationship is possible thanks to the simulation software's own functionality. In this case, the user can modify the parameters of the network elements including both internal elements and layouts at the beginning of simulation. This simulation model offers the option of choosing where to locate the different resources. In particular, this MS contains repeated sections in the plant such as several picking, storage, or market place points. To simplify the construction of the model, we built independent submodels that were added as many times as necessary to represent the real plant. Through the database, we assign to each submodel which resource (network element) it represented and how it should be managed (logic).

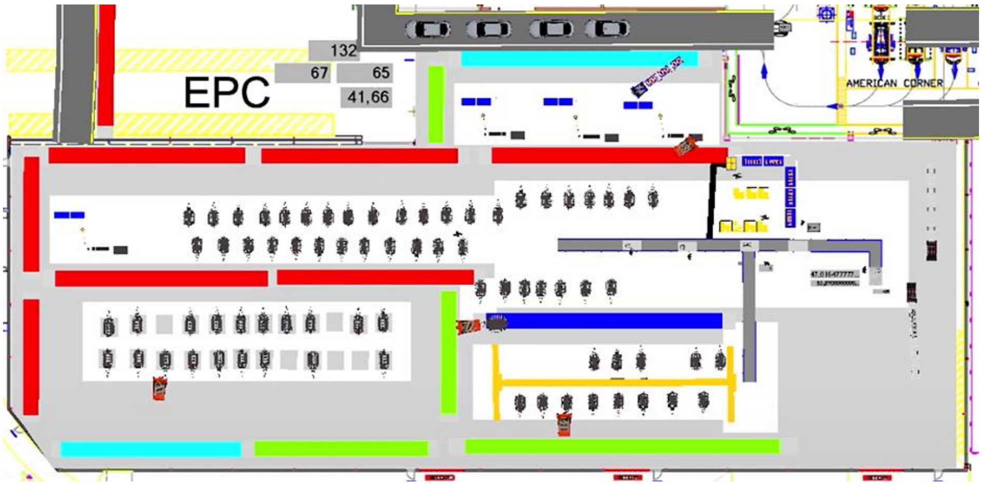

Figure 5: The engines and transmission assembly plant simulation.

The decision outputs can also be changed; e.g., what kind of sequence to create, the production mix or the forklifts strategy. Visual elements are also sensitive to changes, such as the capacity intervals against which to compare the results of the output, the image of the different resources according to their state (working, idle, blocking etc.). Fig. 5 depicts the assembly plant with some of the discussed elements, and also includes a link to one of the simulation footages. For instance, thanks to the visual elements, we could identify design problems before the solution was implemented. The number of market place points between the warehouse and the picking stations were not well sized according to the assembly sequence and mix production. This area had to be expanded with more submodels. Thanks to the modular growth allowed by the 4-layers, this type of changes only required minor modifications in the network and database layers, without interfering in the design of the rest of the layers.

To summarize, with this specific approach for MS, it is possible to separate the problem in order to achieve a better overall design with different combinations between layers.

5. CONCLUSIONS AND FUTURE RESEARCH

This research work proposes the structure, connections and some benefits of a 4-layer architecture, network, logic, database and visual, to simulate real manufacturing systems. In addition, we present two case studies where the 4-layer architecture was applied to deal with the complexity of the systems.

4-layer modelling makes the real separation of the physical system and the logical system explicit in simulation. The first benefit of this separation is that the modelling team acquires a much better understanding of the real problem because it has to face this as it actually is in reality: a material and information flow. A second benefit can be obtained if we consider that a team of engineers with different capabilities might work simultaneously on the same model (in our case an industrial engineer was responsible for the Network and the Visual Layer and, at the same time, a computer science engineer focused on the Logic and Database Layers).

Moreover, the explicit split of the Physical and Logic Layers facilitates the generation of alternatives in both the physical world and the logical one. It also makes the independent management of certain system elements easier, such as workers and handling equipment, or even the routing file. In most complex systems, like those involved in the manufacture and assembly of automobiles, it is relevant to consider the sequencing of operations during the analysis. Separating the Physical Layer and the Logic Layer facilitates the connection of the models that sequence the products and operations of activities. It could also be used to validate the system's robustness in case of non-compliance with orders received. Designing the Logic Layer separately from the Physical Layer would allow the direct use of the algorithms developed and tested as applications for the actual management of the plant or part of it.

The separation of the Logic and Database layers allows information to be exported from simulation, which makes easier analyses that use spreadsheets or other statistical tools. The use of *Andon* elements in the simulation model improves communication with end users as they are accustomed to having statistical values that are visually represented in the same place where operations occur. The ability to interact with the model during simulation (e.g., by stopping a machine) allows the user to visualise the effect of his action on the simulation.

This simulation approach provides other advantages over conventional modelling. On the one hand, more realistic simulation is obtained by decoupling the information flow and the material flow. On the other hand, it allows identifying problems which would otherwise have been overlooked since modelling involves much more modular thinking. This feature allows users to reuse layers with certain similarities in future models. It is also possible to combine a Logic Layer with multiple network layers of a layout design, and *vice versa*. These advantages can significantly cut down modelling efforts, and can avoid having to construct a different model for each alternative.

From a more general standpoint, the two cases presented herein illustrate the simulation methodology employed, and also present a structured procedure to describe a model whose use is proposed for similar cases. We can therefore conclude that the proposed model construction approach provides enough flexibility as to be used when analysing capacity or distribution alternatives in MS, and when proposing and evaluating improvements in the planning and control procedures of operations.

The present work opens up several research lines, two of which we would like to highlight. Firstly, to develop a methodology to facilitate real-time assessments of the plant situation. Secondly, the simulation model will also be used as a training tool for both middle managers and workers.

REFERENCES

[1] Mourtzis, D.; Doukas, M.; Bernidaki, D. (2014). Simulation in manufacturing: Review and challenges, *Procedia CIRP*, Vol. 25, 213-229, doi:10.1016/j.procir.2014.10.032

[2] Tjahjono, B.; Fernández, R. (2008). Practical approach to experimentation in a simulation study, *Proceedings of the 40th Winter Simulation Conference*, 1981-1988, doi:10.1109/wsc.2008.4736292

[3] Korecký, M. (2012). Risk management in logistics, *Proceedings of the Carpathian Logistics Congress 2012*, 26-32

[4] Li, Y.; Yao, X.; Zhou, J. (2016). Multi-objective optimization of cloud manufacturing service composition with cloud-entropy enhanced genetic algorithm, *Strojniski vestnik – Journal of Mechanical Engineering*, Vol. 62, No. 10, 577-590, doi:10.5545/sv-jme.2016.3545

[5] Centobelli, P.; Cerchione, R.; Murino, T.; Gallo, M. (2016). Layout and material flow optimization in digital factory, *International Journal of Simulation Modelling*, Vol. 15, No. 2, 223-235, doi:10.2507/IJSIMM15(2)3.327

[6] Pidd, M. (1992). Guidelines for the design of data driven generic simulators for specific domains, *Simulation*, Vol. 59, No. 4, 237-243, doi:10.1177/003754979205900403

[7] Tannock, J.; Cao, B.; Farr, R.; Byrne, M. (2007). Data driven simulation of the supply-chain – insights from the aerospace sector, *International Journal of Production Economics*, Vol. 110, No. 1-2, 70-84, doi:10.1016/j.ijpe.2007.02.018

[8] Negahban, A.; Smith, J. S. (2014). Simulation for manufacturing system design and operation: Literature review and analysis, *Journal of Manufacturing Systems*, Vol. 33, No. 2, 241-261, doi:10.1016/j.jmsy.2013.12.007

[9] McLean, C.; Jones, A.; Lee, T.; Riddick, F. (2002). An architecture for a generic data-driven machine shop simulator, *Proceedings of the 2002 Winter Simulation Conference*, Vol. 2, 1108-1116, doi:10.1109/WSC.2002.1166364

[10] Clark, G. M.; Cash, C. R. (1993). Data-driven simulation of networks with manufacturing blocking, *Proceeding of the 1993 Winter Simulation Conference*, 662-669, doi:10.1109/WSC.1993.718305

[11] Lim, D.-E.; Seo, M. (2014). A generic simulation framework for efficient simulation analyses for semiconductor manufacturing: A case study, *International Journal of Control & Automation*, Vol. 7, No. 2, 75-84, doi:10.14257/ijca.2014.7.2.08

[12] Kim, B. I.; Jeong, S.; Shin, J.; Koo, J.; Chae, J.; Lee, S. (2009). A layout- and data-driven generic simulation model for semiconductor fabs, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 22, No. 2, 225-231, doi:10.1109/TSM.2009.2017624

[13] Son, Y. J.; Wysk, R. A. (2001). Automatic simulation model generation simulation-based, real-time shop floor control, *Computers in Industry*, Vol. 45, No. 3, 291-308, doi:10.1016/S0166-3615(01)00086-0

[14] Wy, J.; Jeong, S.; Kim, B. I.; Park, J.; Shin, J.; Yoon, H.; Lee, S. (2011). A data-driven generic simulation model for logistics-embedded assembly manufacturing lines, *Computers & Industrial Engineering*, Vol. 60, No. 1, 138-147, doi:10.1016/j.cie.2010.10.011

[15] Wang, J.; Chang, Q.; Xiao, G.; Wang, N.; Li, S. (2011). Data driven production modeling and simulation of complex automobile general assembly plant, *Computers in Industry*, Vol. 62, No. 7, 765-775, doi:10.1016/j.compind.2011.05.004

[16] Debevec, M.; Simic, M.; Herakovic, N. (2014). Virtual factory as an advanced approach for production process optimization, *International Journal of Simulation Modelling*, Vol. 13, No. 1, 66-78, doi:10.2507/IJSIMM13(1)6.260

[17] Seebacher, G.; Winkler, H.; Oberegger, B. (2015). In-plant logistics efficiency valuation using discrete event simulation, *International Journal of Simulation Modelling*, Vol. 14, No. 1, 60-70, doi:10.2507/IJSIMM14(1)6.289

[18] Kehris, E. (2009). Web-based simulation of manufacturing systems, *International Journal of Simulation Modelling*, Vol. 8, No. 2, 102-113, doi:10.2507/IJSIMM08(2)4.126

[19] Saez-Mas, A.; Garcia-Sabater, J. P.; Morant-Llorca, J.; Maheut, J. (2016). Data-driven simulation methodology using DES 4-layer architecture, *Working Papers on Operations Management*, Vol. 7, No. 1, 22-30, doi:10.4995/wpom.v7i1.4727

[20] Saez Mas, A.; Garcia-Sabater, J. P.; Morant Llorca, J.; Maheut, J. (2016). Assembly plant simulation to support decision-making on layout design considering safety issues. A case study, *Working Papers on Operations Management*, Vol. 7, No. 2, 64-88, doi:10.4995/wpom.v7i2.4721

[21] Wallace, T. (2008). Cycles of production: from assembly lines to cells to assembly lines in the Volvo Cab Plant, *New Technology, Work and Employment*, Vol. 23, No. 1-2, 111-124, doi:10.1111/j.1468-005X.2008.00206.x

[22] Hüttmeir, A.; De Treville, S.; Van Ackere, A.; Monnier, L.; Prenninger, J. (2009). Trading off between heijunka and just-in-sequence, *International Journal of Production Economics*, Vol. 118, No. 2, 501-507, doi:10.1016/J.IJPE.2008.12.014

[23] Garcia-Sabater, J. J.; Garcia-Sabater, J. P.; Marin-Garcia, J. A. (2008). Optimización de los recursos humanos a través de las TIC. Resultados de la reorganización laboral de un proveedor logístico, *Economía Industrial*, Vol. 370, 143-151

[24] Lario Esteban, F. C.; Miralles Insa, C.; García Sabater, J. P. (2001). La programación de la producción en el sector del automóvil. El problema de la segmentación, *IV Congreso de Ingeniería de Organización (CIO 2001)*, 12 pages

[25] Lario Esteban, F. C.; Bautista, J.; Companys, R.; García Sabater, J. P. (2001). Secuenciación en contexto dinámico de unidades homogéneas en el sector del automóvil, *IV Congreso de Ingeniería de Organización (CIO 2001)*, Vol. 1, 311-312