Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

SCALABLE, RECONFIGURABLE SIMULATION MODELS IN INDUSTRY4.0-ORIENTED ENTERPRISE MODELING

Mahdi El Alaoui El Abdellaoui
Emmanuel Bricard

elm.leblanc – Innovation Center of Operations
Bosch Group, Thermotechnology Division
Drancy, 93700, FRANCE

Frédéric Grimaud
Paolo Gianessi
Xavier Delorme

Mines Saint-Étienne, Univ. Clermont Auvergne
CNRS, UMR 6158 LIMOS, Institut Henri Fayol
F-42023, Saint-Étienne, FRANCE

ABSTRACT

The increasingly unpredictable demand is among the major challenges of Industry 4.0, asking manufacturing systems (MS) for a capacity of adaptation that accelerates the decision processes. Decision Support Systems (DSS), and among them Simulation-based DSS, must therefore result from new Enterprise Modeling (EM) frameworks, capable of taking advantage of advances in data integration and processing possibilities. After briefly introducing such a new EM framework, MEMO I4.0, this article shows how to use it to derive scalable, flexible simulation models, so as to improve and monitor the MS processes and finally address this challenge. An example to generate a simulation model for a real case study is presented.

1 INTRODUCTION

In the new era of Industry 4.0, market complexity is increasing more and more, and due to new customer requirements for product customization and the globalization concept, customer demand becomes more difficult to predict leading to the development of new business models (Pereira and Romero 2017). Therefore, companies should have flexible and reconfigurable Manufacturing Systems (MS), as the product life cycle is shorter, competitiveness has increased and involves new investments and business strategies. Thus, the production methods are completely impacted, and the same goes for the Enterprise Modeling (EM) methodologies. In the Last three decades, EM frameworks have been used to describe the MS from the conceptual analysis of an industrial activity to the implementation of a Decision Support Systems (DSS). The decision support at different levels according to the time horizon allows to monitor key performance and react to unpredictable events (Roboam et al. 1989). More and more often, EMs integrate simulation-based DSS (Barjis 2011). The evolution of industrial paradigms imposes the evolution of DSS, which should not only include the uncertain aspect, but also be redesigned and implemented in parallel to the MS, if not in advance (Felsberger et al. 2016). However, the more recent EMs, inspite of integrating at the conceptual level the specific concepts of Industry 4.0, are not yet able to carry out the modeling up to the implementation of an agile, quickly reconfigurable simulation-based DSS.

In this article, we propose a new EM framework, MEMO I4.0, which has this conceptual-to-implementation feature, and focus on its ability to generate a MS simulation model and to quickly scale it up, or reconfigure it, whenever a new configuration of the MS is at study. After reviewing the literature of the most known EM frameworks and their limits in Section 2, Section 3 briefly introduces MEMO I4.0 that is used in Section 4 to implement a reconfigurable MS simulation model. In Section 5, a case study inspired from a real case of elm.leblanc company is presented. Finally, Section 6 concludes this work.

2 ENTERPRISE MODELING AND SIMULATION: A SHORT REVIEW

This section briefly reviews both classical EM frameworks and the more recent ones, designed to capture the emerging needs of Industry 4.0. After an analysis of the limits of existing EM frameworks, the urge of new EM paradigms is shown, especially for what concerns the generation of simulation models.

2.1 Features and Drawbacks of Classical EM Frameworks

We present three well known frameworks used in EM (GIM, CIMOSA and GERAM), which could be considered as basis of the EM standards namely ISO 19439 and 19440 norms (Millet et al. 2013).

The GIM framework (GRAI Integrated Methodology) is based on the GRAI (Graph with Results and Actions Inter-related) model (Chen and Doumeingts 1996). It divides the MS into three subsystems: physical (or operational), decisional and informational and defines three dimensions, namely system view, life cycle and abstraction level. The first considers functional, physical, decisional and informational views. The second deals with system phases from definition of study domain to implementation. The last refer to three levels (conceptual, structural, realizational) of system abstraction.

The CIMOSA architecture considers three dimensions to examine different aspects of enterprise. The instantiation dimension concerns the genericity degree of modeling building blocks, allowing to model generic to specific enterprises. In the following, for the sake of clarity, we will talk about genericity degree dimension to refer to this dimension. The life cycle dimension considers the different steps of the MS life cycle from the analysis of user requirements to implementation. The view dimension partially differs from that of GIM and captures system functionality and behaviour by considering Function, Information, Resources and Organisation views: by applying a life cycle from definition to implementation, we obtain the corresponding sub-models (Vernadat 2014).

GERAM aims at covering some concepts not considered by GIM and CIMOSA. GERAM proposes the same dimensions of CIMOSA but the life cycle dimension is enlarged with identification and concept phases before the requirement analysis to the operation and decommission phases after the implementation phase. One of the major contributions of GERAM is that the view dimension becomes extensible and can be adapted to the MS object of study (Bernus et al. 2014). GERAM is the more advanced among classical EM frameworks, as the identification phase allows to assess the features, objective, values and strategies of the system under study, while the concept phase injects the knowledge of how to react to new needs. However, if such new needs are detected, the modeling of the MS must restart from scratch. Another drawback is that although the Information view is present, it lacks the modeling data treatment capabilities that nowadays are required to consider the automatization of data. That is why, generation of new models using classical methods would take much time and is not relevant within Industry 4.0.

2.2 Features and Drawbacks of New EM Frameworks

Perhaps the most known Industry 4.0-oriented EM framework is RAMI 4.0 (Reference Architecture Modeling Industry 4.0). RAMI 4.0 introduces new features as identifiability, protocol of communication, safety to adapt to advanced components (e.g. IoT) and communication modes. In order to represent new, complex industrial scenarios, RAMI 4.0 incorporates (Pisching et al. 2018) three types (vertical, horizontal, end-to-end) of integration to address three new dimensions that decompose the MS. The vertical integration refers to the layer dimension, a new take on the view dimension of GIM, CIMOSA and GERAM that integrates new system layers of a smart MS (e.g. Integration, Communication, Business). The horizontal integration refers to the hierarchical level, or scale dimension, from product level to smart factories. Finally, the end-to-end integration refers to the life cycle value stream dimension, which includes the value chain from development of product to its production and maintenance.

IIRA (Industrial Internet Reference Architecture) is another recent EM framework that focuses on how smart devices and the related data are handled and specifies how to address interoperability issues. Its main advantage remain on its open architecture. IIRA introduces four viewpoints namely business, usage, functional and implementation (Bader et al. 2019).

In spite of being EM frameworks inspired by new Industry 4.0 trends, both RAMI4.0 and IIRA seem to lack the genericity dimension, which is one of the key element to achieve the reusability of different models. Moreover, as far we are aware of, no work exists to date that used them as reference from definition to the implementation in real-cases (Langmann and Rojas-Pena 2016).

2.3 EM and Simulation. The Need for a New Paradigm

Discrete event simulation (DES) is one among the most common simulation methods and also one of the most decision support techniques as it enables evaluation of different system configurations and policies. Among recent reviews about simulation, we refer the reader to (Negahban and Smith 2014). The new technologies allow the emergence of real-time simulation, and therefore of digital twins, i.e. of simulation models that are fed by real-time data and run in parallel with the real system (Alcácer and Cruz-Machado 2019). In EM frameworks, simulation models are most of the times obtained following the implementation phase and using the process view as a core model that control and ensure the consistency between view models. However, neither classical nor more recent EM seem suitable to generate simulation models that live up to the expectations of Industry 4.0. The former do not consider reconfigurability in the MS model, forcing the rebuild from scratch of simulation models in case of changes, and ultimately preventing the build of scalable models. The latter do contemplate frequent changes in the MS, but seem to lack the genericity degree aspect and also a structured approach to implement simulation models, and thus the possibility to test the developed enterprise models in real cases.

Table 1 summarizes the dimensions of the reviewed existing EM frameworks, along with their suitability for Industry 4.0 and the aforementioned "conceptual-to-implementation" feature (*Workflow* column). In our opinion, new EM architectures should be defined that rest upon Industry 4.0 concepts but offer a conceptual-to-implementation backbone with the integration of different views, in order to allow the generation of scalable, flexible and quickly reconfigurable simulation models.

Table 1: Most relevant features of existing EM frameworks.

framework	**Dimensions**				**I4.0-oriented**	**Workflow**
	Scale	Genericity	Views	Life cycle		
GIM	×	×	not smart	✓	×	not complete
CIMOSA	×	✓	not smart	✓	×	not complete
GERAM	×	✓	not smart	✓	×	✓
RAMI 4.0	✓	×	smart	✓	✓	×
IIRA	×	×	smart	not generic	✓	×

3 A NEW METHODOLOGY OF ENTERPRISE MODELING FOR INDUSTRY 4.0 (MEMO I4.0)

This section presents MEMO I4.0, a new, fully Data-Driven EM approach we developed to help the design of a MS in the Industry 4.0 context. MEMO I4.0 has all the dimensions and Industry 4.0-oriented features of Table 1 and hence stands as new, smart EM tool, meant for companies with a long-term vision about future orientations which operate in a structured, standardized framework based on their DNA and guidelines.

3.1 Key Principles of MEMO I4.0

The key modeling principles of MEMO I4.0 can be summarized into four features:

- **Agility:** the implementation of new system configurations, or the introduction of new KPIs, should be made rapidly. To achieve this, we recommend the development of Data-Driven models.

- **Modularity:** all models should be made up of modules that are assembled and communicate to build the targeted system. To this end, it is essential to use incremental development (module by module) and ensure the decoupling of flows (material, information, and control).
- **Interoperability:** simulation models should be platform-independent and more in general not be restricted to DES. The use of mapping layers as gateways is proposed.
- **Robustness:** modules are validated separately, then integrated tests are led to validate assembly of modules.

3.2 Structured Approach of MEMO I4.0

The main motivation of MEMO I4.0 is to reduce the time spent in the development cycle of simulation projects (modeling, coding, gathering data/information and verification) in favor of agility, increased reactivity and process analysis capabilities. MEMO I4.0 aims at covering the lacks of existing tools (see Section 2) and offers a framework to support the modeling process and generate an Optimization/Simulation-Based DSS. The structured approach of MEMO I4.0 to achieve this follows the Modularity principle and consists of two main stages, organised according to the life cycle phases to ensure robustness and shown in Figure 1:

- **Strategical/tactical stage:** it concerns the *Development Cycle* to create the modules used at operational level to build integrated simulation models, based on DNA/guidelines and Key principles. Newly conceived modules are added to a *Library of Modules* to allow for knowledge capitalization;
- **Operational stage:** it concerns the *Execution Cycle* to generate integrated simulation models and assess the system performances. Missing or obsolete modules trigger a new Development Cycle. Required modules are instantiated with case-specific parameters.

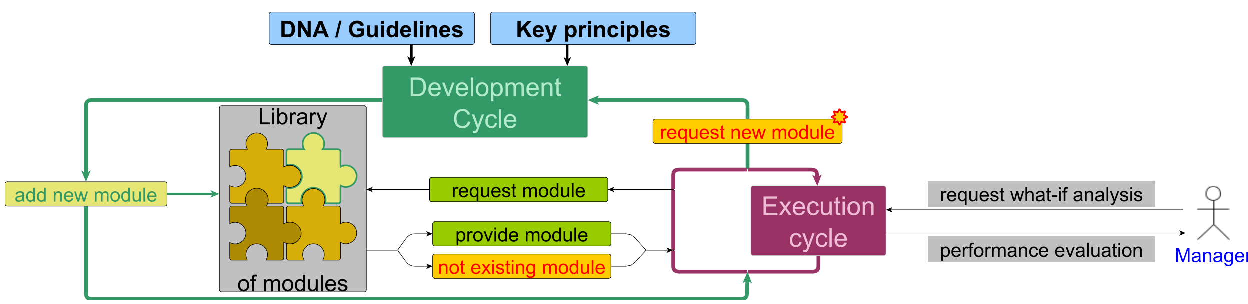

Figure 1: The Development Cycle is the key element in building up the Library of Modules.

3.3 Dimensions of MEMO I4.0

MEMO I4.0 is a framework based on four dimensions, namely genericity degree, scale, life-cycle and view.

As with CIMOSA, genericity degree is about reusability. Three types of modeling modules are defined: generic (that can be used in all contexts), partial and specific (to a specific context).

The scale dimension can be defined (as in RAMI 4.0) as the modeling granularity level of modules. The same production process can be e.g. described at the plant, line, workstation or even product level. A finer granularity allows for a greater flexibility. Modules that differ in scale can be created for the same activity and genericity degree. The scale dimension of modules allow to tune the overall model granularity.

As with most EM frameworks, the life cycle dimension of MEMO I4.0 acts as a guideline to structure the modeling process in four phases: definition, design, implementation and maintenance. These phases are followed at strategical/tactical stage (see Section 3.1) to generate the modules, and help structuring the generation of a simulation model at operational stage. They are dealt with in Section 4.

The view dimension of MEMO I4.0 is inspired from that of existing EM frameworks and has the same purpose of simplifying the modeling task. This dimension represents a support to generate a compatible model regarding needs, objectives, system features, etc. The proposed set of views (function, process, decision, performance, information, data, resource & organization) is meant to cover different enterprise aspects. Views interact to ensure the global consistency of the modeling during the life cycle phases.

4 SIMULATION

We focus in this section on the application of MEMO I4.0 to generate the simulation model for the MS at study. To achieve this, we follow the structured approach presented in Section 3.2.

4.1 Strategical/Tactical Stage : Development Cycle (Development of Simulation Modules)

As said, each stage is organised according to the life cycle phases.

- **Definition phase:** a strategy to build a digital twin is defined that takes into account both the company DNA and guidelines and the vision towards Industry 4.0 in order to consider the change of decision-making process, the evolution of the types of MS, the emergence of new KPIs (El Alaoui El Abdellaoui et al. 2019). According to this vision, the genericity degree is decided so that the module to develop integrates both current needs and possible future evolutions. In this phase, SADT (Structured Analysis and Design Technics), a method used to yield a systemic view of a complex system based on a hierarchical structure, is used to explain the details of each activity in a process workflow diagram. This task can be achieved either by means of interviews, or, if the MS already exists, by means of Gemba walk. Then, functional specifications are identified using reference documents and interviews with the managers of the different activities, so to understand needs, objectives, most frequent problems, decision workflow. Further, global process KPIs and specific KPIs are defined in detail for each activity. To ease maintenance, for each activity informations are grouped and the resource set is identified, as well as the related roles and data, according to the department in charge of it. All these elements require to choose the adapted scale level and will impact the modeling granularity in the design phase.
- **Design phase:** based on SADT diagram, a flow diagram is created to detail all process flows: physical, information and control. UML (Unified Modeling Language) is used both to create a meta-model diagram compatible with the SADT diagram and to include all information related to decision process. The computation of the already identified KPIs, as well as the required information, are detailed. Data model and structure are defined so as to make the model fully Data-Driven. Tree structures are preferred in case that a many-to-many relationship exists between the entities, otherwise, a 2D or 3D matrix is sufficient. Then, the modules that are compatible with UML diagram are developed using an incremental approach and the scale level decided in the definition phase allows to adjust the modeling granularity and ultimately the flexibility of the module. According to the chosen genericity degree, the modeling choices are made. These are influenced by the guidelines, but could take into account potential evolutions that it may be interesting to investigate. Modules are developed in parallel: compatibility is made possible by replacing connected modules of a specific module by simplified versions.
- **Implementation phase:** an object-oriented approach is used that consists in identifying objects that are compatible with the process to simulate. In order to ease the model verification, traceability files are created for each activity that include all its attributes and variables. Further, corner tests are run to ensure the stability of the model. The parallel, compatibility-driven development and integration of modules is carried out and the expected integrated model behavior is verified. In this phase, tables related to module are created in a database (DB) based on the data model to ensure data integrity and compatibility with meta-model diagram. Experimentations are launched first to

check whether the genericity degree of module fixed is respected, then to check if the module flexibility is achieved w.r.t. the chosen scale level.

- **Maintenance phase:** the documentation related to each module is created to keep track of modeling choices and managed by a versioning platform. All the tests are run in the test branch; it is recommended to save all versions of the tests regardless of whether their exit status is positive or not. Based on the traceability files and KPIs the results are discussed with the team in review meetings to validate or not the module version. In case of validation, and only after compatibility among views is checked, a versioning commit is performed, otherwise an analysis is carried out about how the model can be adjusted in the design phase, e.g. by changing the control logic or the scale level. If the degree of genericity decided at definition phase is no longer correct to model the process, the module is updated or a new one is created from scratch.

The strategical/tactical stage is responsible to develop and update the modules versions. These modules are used directly at operational stage when needed to derive the simulation model as explained below.

4.2 Operational Stage : Execution Cycle (Creation and Instantiation of the Integrated Model)

- **Definition phase:** starting from the needs expressed by the management, the modeler decides whether they can be covered with modules that have been already derived by the DNA/guidelines of the group; if not, modules are designed from scratch in a classic scheme. In the latter case, after choosing the suitable preexisting modules, the modeler decides the new ones that are required to generate the integrated model. The creation of new modules must follow the same development described in Section 4.1 so as to put it in a strategic/tactical perspective. However, to avoid as much as possible the creation of new modules, it is suggested to decide the genericity degree and the scale level so as to consider expected evolutions.
- **Design phase:** once all the required modules are available, the modeler checks whether they need to be updated. If so, this change is made in the maintenance phase of the strategical/tactical stage, along with the related documentation. Then, the modeler assembles these modules to obtain an integrated simulation model, whose behavior is already checked at the implementation phase of the strategical/tactical stage. A version of the integrated model can then be instantiated to simulate a specific configuration of the MS and perform a what-if analysis. Finally, to comply with the Interoperability principle of MEMO I4.0 (see Section 3.1), a mapping layer of the architecture of the integrated model (Figure 2) is defined that depends on the specific simulation platform and transforms the business model into a runnable simulation object.
- **Implementation phase:** a flexible development architecture (Figure 2) is used which is inspired by a classical 3-tier software architecture. It consists of a presentation layer (user-interface), a business layer (logic) and a data access layer. A mapping layer is added to be able to instantiate the integrated model, represented by the business layer, to a runnable object. The tuning of the integrated model settings, which is needed in order for the mapping layer to derive the simulation object, can in some cases (e.g. with digital twins) be partially automated and fed by the MES (Manufacturing Execution System). The validation step is based on historical data, Gemba walk and the advice of experts. Two situations (As-Is or To-Be) are possibles in the validation process. In the former, the simulation model represents the existing system: the validation can then be based on Gemba walk and meeting review to discuss the KPIs values with experts, and historical data and statistical tests (e.g. chi-square goodness-of-fit) can be used to study the difference between the observed and the expected values. As to the To-Be situation, the model simulation represents a future system and the validation cannot be based on historical data: the correctness of model behavior is then ensured based not only on expertise but also on sensitivity tests and other statistical tests (variance analysis, regression analysis, etc).

- **Maintenance phase:** the same actions of the development of modules are performed, except that versioning is made for the integrated model. The documentation related to the process and simulation model, which also includes solutions and recommandations, is created or updated. This task has the highest priority to allow knowledge capitalization and version improvement.

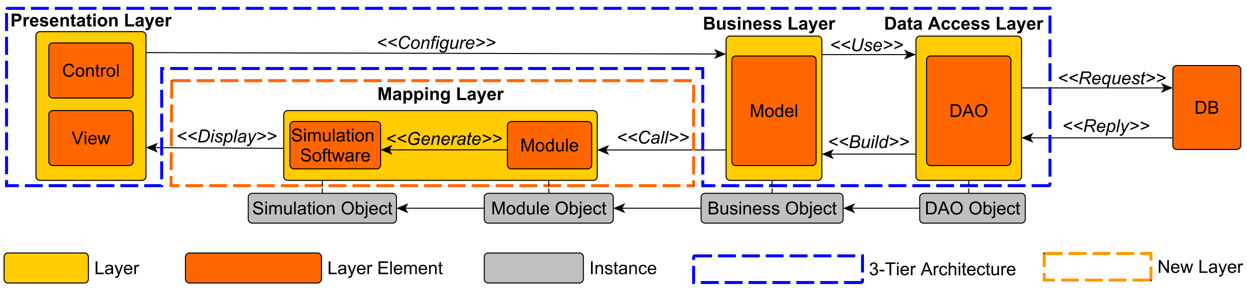

Figure 2: Development architecture for the integrated simulation model.

4.3 Views: Support for Modeling Consistency

During the development of each module, views contribute at different phases of strategical/tactical stage to ensure the global consistency and help the modeler making the most fitting modeling choice w.r.t. the fixed genericity degree and scale level. In this section, we explain how.

- **Function view:** it allows to specify DNA and guidelines of the group at the definition phase, decompose the global process into activities and detect in/out flows of each activity, so as to generate the SADT diagram.
- **Process view:** it focuses on the definition of different flows (physical, informational, control) at definition phase, before separing such flows w.r.t the decomposition of activities at design phase. In addition to rules imposed by guidelines, others rules derived from system reality are captured by Gemba walk and meetings with team project. The pseudo-code format used to formalize these rules also simplifies later coding steps to produce the simulation model (implementation phase).
- **Decision view:** it associates each activity with a set of decisions at definition phase. To do so, the cartography tool of (El Alaoui El Abdellaoui et al. 2019) is used to outline the structure of the decision-making process according to time horizon (strategical, tactical, operational) and activities. This view ensures the coupling between optimization and simulation models. Depending on which decision problem to address, the modeling choices are made at design phase, while the choice of the solving approach is made during implementation phase. Due to the mapping between activities and decisions, the results of optimization are directly matched with the simulation model.
- **Performance view:** it defines the KPIs related to each activity at the definition phase. A careful decision of the scale level, and thus of the modeling granularity, is recommended to properly choose the KPIs to be added. The problem of how to assess the solution provided by the decision view for each activity can also help defining the proper KPIs. Once KPIs are defined, their structure is built at the design phase and the data required to compute them are specified, as well as the representation of data (dashboard, tables, etc) that better allows to check the module behavior at the implementation phase.
- **Information view:** it specifies the information associated with each activity (and related decisions) at the definition phase (i.e. the information needed to represent the input/output flows of each activity), build the model of business layer and compute the corresponding KPIs. The information model uses UML diagram at the design phase, while at the implementation phase the data access layer is created to transform the raw data to information required to set up the module.

- **Data view:** it determines the raw data needed to extract the information used in the information view at the definition phase. This view chooses the data structure and uses UML diagram to build the data model at the design phase. For conformity with the other views and the Modularity principle, the data model for each activity is built first, before the global data model of the system at study. At the implementation phase, the DB tables corresponding to the data model are created.
- **Resource-Organisation view:** it allows to define departments linked to each activity and resources associated with departments at the definition phase. To simplify the data management, an organogram is suggested at design phase to define the roles of resources and related data. Since resources are data sources, they could be either decision-makers, machines or connected objects. At implementation phase, the technical solution to capture the raw data is chosen.

When developing a module, the maintenance phase is essential to ensure compatibility among views. Based on the documents generated by each view (SADT/flow/UML diagrams, data model, etc), the feasibility of the mapping must be checked. In case compatibility is not achieved, the module cannot be validated, and review of definition and design phases is necessary.

5 CASE STUDY

This section presents an application of MEMO I4.0 on a real-world case study to derive a simulation model for a MS, more precisely an assembly line of a plant of elm.leblanc company, a part of the Bosch group.

5.1 Development Of Modules

- **Definition Phase:** the context of study, activities and KPIs are defined. The plant has two storage areas and several assembly lines. Each line can produce one or more products according to a push policy between workstations. Each storage area has its own picking station area to store full carts when prepared. Milkruns load full carts and follow specific routes to feed the assembly line with components. They stop at each supermarket, unload the corresponding boxes and carts, retrieve empty ones, go back to the picking stations, drop empty boxes and carts. The consumption of components in supermarkets by workstations depends on the tasks and the bill of materials of each product. A preparation order is sent to the associated storage area for replenishment either when a trigger quantity is reached or the lot size of product is assembled. Preparation orders are fulfilled according to some priority rules. Some of the constraints to take into account are: locations of storage areas, picking stations, supermarkets, workstations; picking station space; picking capacity of carts; storage capacity of supermarkets. In each shift, a break time should be allocated. Each day, the logistic manager decides the resources (preparators and milkruns) to allocate to each line, while the production manager defines a production planning and decides the resources (operators) assigned to each workstation. In addition, based on technological constraints and for optimization reasons, the production sequence of the two products is fixed. A SADT diagram (Figure 3, see Section 4) is used to describe the MS activities, here Preparation, Loading, Transport and Production.
  For each activity, the main features (e.g. type of line, replenishment policy, lot size) are determined, based on the Bosch Production System (BPS) guidelines and the plant-specific constraints, and are validated after consultation with the experts. We define the KPI of each activity (see Table 2) with the team project and we use them as one of the tools to analyze the results of our model.
  Once activities have been identified, the genericity degree of their corresponding modules can be decided: in this case, it is set to *specific* for the preparation and loading modules and to *partial* for the transport and production modules. To have maximum flexibility and meet the chosen KPIs, the scale dimension of all the modules is fixed at the *product* level.
- **Design Phase:** the UML diagram is built and the different flows are separated. The process described by SADT is exploited to design the UML diagram of Figure 4, which in turn is used to implement the business layer (see Section 4.2) of the development architecture.

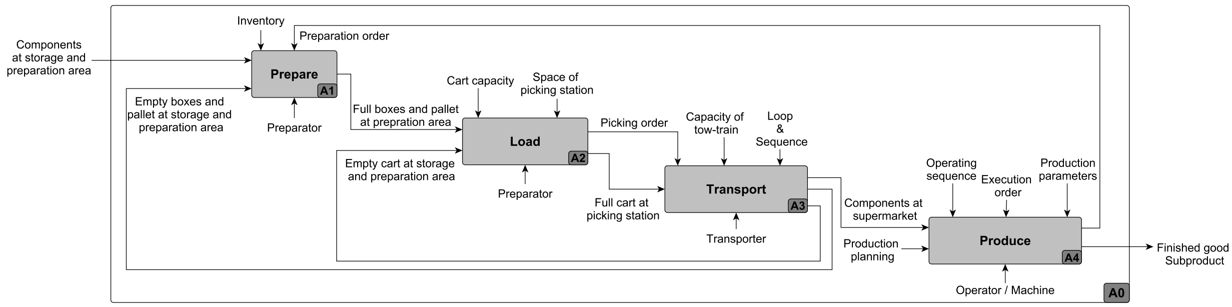

Figure 3: SADT of MS process.

Table 2: KPI of each activity.

Preparation	Loading	Transport	Production
Mean time to prepare one preparation order per storage area	Mean # of carts loaded per loop per milkrun driver	Mean # of loop per shift	Productivity
Mean time to prepare a list of preparation orders per storage area	% of carts filling	Mean # of empty boxes and carts (un)loaded per supermarket	Mean # of breakdown due to lack of components
per-shift usage % of preparators	per-shift usage % picking stations	per-shift usage % of transporters	per-shift usage % of operators/machines

For each activity, flows (physical, informational, control) are separated to comply with the Modularity key principle of MEMO I4.0 (see Section 3.1). Table 3 provides the process flows.
For instance, to model the production policy in the production module and comply with the chosen genericity degree, the push policy is modeled as autonomous pull-policy i.e after synchronization of information and physical flows, the fabrication order to start production is sent to the workstation. In accordance with the fixed scale level, this decision is made at *product* level.

- **Implementation Phase:** after definition and design, modules corresponding to each activity are developed and their behavior is checked. The UML diagram represents the information model, and each class in it includes what is required by the module to evaluate the performance of the system in case a specific decision related to activity is changed. Moreover, tables of each module are created in databases, so as to ensure data integrity and compatibility between modules when they will be assembled into the integrated model. To check the behavior of modules, we create traceability files that report the execution details of simulation and we use the KPIs already defined

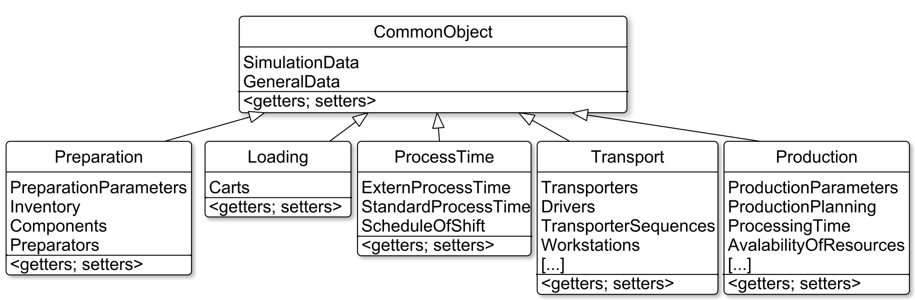

Figure 4: UML diagram of MS process.

Table 3: Flows of each activity of MS process.

<table>
<tr><th>Flow type</th><th>Preparation</th><th>Loading</th><th>Transport</th><th>Production</th></tr>
<tr><td>Physical</td><td>Component,
Box, Pallet</td><td>Box,
Pallet, Cart</td><td>Milkrun driver,
Cart</td><td>WIP</td></tr>
<tr><td>Informational</td><td>Preparation order</td><td>-</td><td>Picking order</td><td>Execution/
Fabrication/Launch/
Transfer orders</td></tr>
<tr><td rowspan="2">Control</td><td>Control preparation,
Inventory, box check</td><td>-</td><td>Control picking order,
Check empty boxes/carts</td><td>Control production,
Control demand</td></tr>
<tr><td colspan="4">Control shift and resources</td></tr>
</table>

at strategical/tactical stage. Moreover, to conduct sensitivity analysis, corner tests are run and possible corrections are made based on the results. To check the respect of the genericity degree and the scale level, different instances of the same module are created and tested.

- **Maintenance Phase:** the documentation (attributes, modeling choices, data structures, etc.) of modules is created. To achieve incremental development, each time a new module version is validated and compatibility of views is approved, version commit occurs. For example, to develop the production module with *partial* genericity degree, a base version with specific degree is developed, then modeling choices are extended to achieve a partial degree. The same applies for flexibility.

5.2 Development and Instantiation Of Integrated Model

- **Definition Phase:** the management request is analyzed. In this case study, elm.leblanc needs to verify whether the current configuration of the assembly line at study is capable to offer an appropriate response in case the customer demand increases. The simulation model must include the entire MS process: preparation, loading, transport and production activities. The considered assembly line produces two main products according to a push policy between workstations. This is compatible with the assembly module developed in the strategical/tactical stage and no further changes are required. Following the procedure outlined in Section 4, an integrated simulation model of the assembly line is derived together with elm.leblanc (experts, operators) so as to comply with elm.leblanc guidelines. After checking whether the modules versions are up to date or not, the instantiation of the integrated model is carried out and tuned. Finally, a what-if analysis is performed to assess how the plant reacts when the demand increases w.r.t. the production planning.
- **Design Phase:** the integrated model is assembled and instantiated to derive the case-specific simulation model. Table 4 gives an example of some parameters of elements of plant configuration.

Table 4: Example of elements of plant configuration.

Element	**Parameters**	**Element**	**Parameters**
Line	Lot size, Product type...	Storage Area	Position, Preparation time...
Workstation	Position, Operations List...	Resource	Planning, Assignments...
Supermarket	Position, Storage capacity...	Shift	Availability & Break time...
Operation	Workstations list, Resources list...	Transporter	Sequence, Velocity...
Component	Replenishment policy, Storage holder...	Picking Station	Position, Cart Type...
Product	Production policy, Operating seq...	...	...

To generate a possible plant configuration, it is sufficient to fix the number of each element (number of lines, of workstations, etc.) and to define the parameters of each element. However, the flexibility and modularity of the integrated model can be fully exploited, and more configurations can be tested, with a different tuning of the parameters. This fully Data-Driven model allows to achieve the Agility key principle (Section 3.1) of MEMO I4.0.

- **Implementation Phase:** the goal is to run the tests, validate the model and discuss the results. Settings are validated with elm.leblanc managers and experts. The tuning of the instance model is done trying to reproduce the daily operations of the MS process. The correctness of the KPI is assessed in meeting reviews. A Gemba walk is performed to check the model behavior w.r.t that of the real assembly line. The production of one week (5 days) is simulated; 10 replications are run. Throughout the simulated week, achieved production varies between 89.5% and 96.8% of the targeted demand. The KPIs defined for the production activity (Capacity utilization and per-day mean blocking time of each workstation) are inspected to find the root causes of this production gap (Figure 5). Three workstations are found to have a high blocking time due to a lack of components, causing the capacity utilization of all workstations from WS7 on to decrease considerably. Traceability files then allow to track the inventory level of components and detect the critical ones (Table 5).

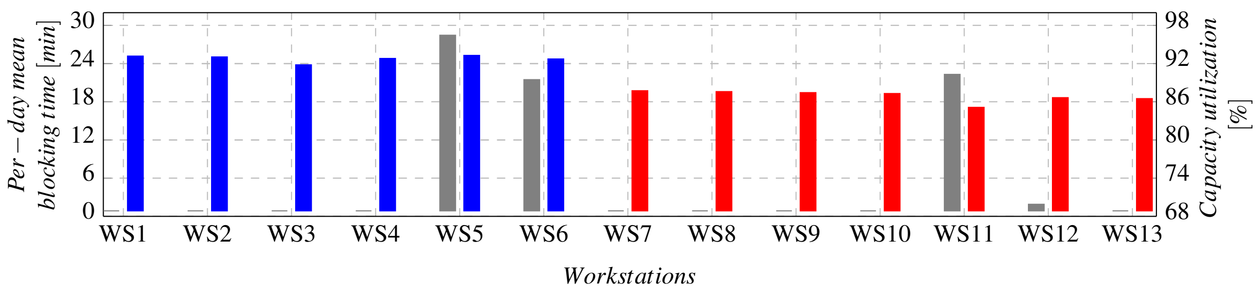

Figure 5: Per-day mean blocking time (grey) and capacity utilization (blue, red if critical) of workstations.

Table 5: Critical components of workstations with non-zero mean blocking time.

		Characteristics			
Workstations	**Components**	**Box capacity**	**Storage capacity**	**Triggering qty**	**Delivered qty**
WS5	W	12	3	18	12
WS6	X	8	4	24	8
WS11	Y	12	2	24	12
WS12	Z	12	4	24	12

- **Maintenance Phase:** some solutions are recommended based on the results of the implementation phase. As for capacity, we could increase either the component box capacity or storage capacity; as for milkrun deliveries, we could increase either the triggering quantity or the delivered quantity. Since both would heavily impact the process, they can be simulated by a new instantiation of the integrated model before being deployed in the plant. The same model is able to represent other plant configurations. Finally, the model documentation and the versioning are updated.

6 CONCLUSION

This study proposes a new EM methodology (MEMO I4.0) with a structured approach to generate a simulation model. A real case study is presented in which elm.leblanc company seeks to anticipate the impacts on its current configuration of a possibly increased demand. Using the fully Data-Driven model, elm.leblanc is able to test different plant configurations and estimate the impact of different parameters. The proposed development architecture and key principles act as a reference for developing/upgrading a simulation model, and could lead to obtain a digital twin. Using the same methodology to develop an optimization model, and the integration of it into the simulation model, is another future perspective, whose goal would be to obtain a decision support tool to provide an optimal plant configuration.

REFERENCES

Alcácer, V., and V. Cruz-Machado. 2019. "Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems". *Engineering Science and Technology, an International Journal* 22(3):899–919.

Bader, S., M. Maleshkova, and S. Lohmann. 2019. "Structuring Reference Architectures for the Industrial Internet of Things". *Future Internet* 11(7):151.

Barjis, J. 2011. "Enterprise Modeling and Simulation Within Enterprise Engineering". *Journal of Enterprise Transformation* 1(3):185–207.

Bernus, P., O. Noran, and A. Molina. 2014. "Enterprise Architecture: Twenty Years of the GERAM Framework". *IFAC Proceedings Volumes* 47(3):3300–3308.

Chen, D., and G. Doumeingts. 1996. "The GRAI-GIM Reference Model, Architecture and Methodology". In *Architectures for Enterprise Integration*, edited by P. Bernus, L. Nemes, and T. J. Williams, 102–126. Boston, MA: Springer US.

El Alaoui El Abdellaoui, M., F. Grimaud, P. Gianessi, and X. Delorme. 2019. "Integrated Decision Process to Design Manufacturing Systems towards Industry 4.0". *IFAC-PapersOnLine* 52(13):1373–1378.

Felsberger, A., B. Oberegger, and G. Reiner. 2016. "A Review of Decision Support Systems for Manufacturing Systems". In *Proceedings of the 1st Intl. Workshop on Science, Application and Methods in Industry 4.0*. October 18th-19th, Graz, Austria.

Langmann, R., and L. F. Rojas-Pena. 2016. "A PLC as an Industry 4.0 Component". In *13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 10–15. Madrid: Institute of Electrical and Electronics Engineers.

Millet, P.-A., L. Trilling, T. Moyaux, and O. Sakka. 2013. "Ontology of SCOR for the Strategic Alignment of Organizations and Information Systems". In *Supply Chain Performance: Collaboration, Alignment and Coordination*, edited by V. Botta-Genoulaz, J.-P. Campagne, D. Llerena, and C. Pellegrin, 171–210. London: John Wiley & Sons, Inc.

Negahban, A., and J. S. Smith. 2014. "Simulation for Manufacturing System Design and Operation: Literature Review and Analysis". *Journal of Manufacturing Systems* 33(2):241–261.

Pereira, A. C., and F. Romero. 2017. "A Review of the Meanings and the Implications of the Industry 4.0 Concept". *Procedia Manufacturing* 13:1206–1214.

Pisching, M. A., M. A. Pessoa, F. Junqueira, D. J. d. S. Filho, and P. Miyagi. 2018. "An Architecture based on RAMI 4.0 to Discover Equipment to Process Operations Required by Products". *Computers & Industrial Engineering* 125:574–591.

Roboam, M., M. Zanettin, and L. Pun. 1989. "GRAI-IDEF0-Merise (GIM): Integrated Methodology to Analyse and Design Manufacturing Systems". *Computer Integrated Manufacturing Systems* 2(2):82–98.

Vernadat, F. 2014. "Enterprise Modeling in the Context of Enterprise Engineering: State of the Art and Outlook". *International Journal of Production Management and Engineering* 2(2):57–73.

AUTHOR BIOGRAPHIES

MAHDI EL ALAOUI EL ABDELLAOUI is a PhD Student in Industrial Engineering at the Ecole des Mines de Saint-Étienne (France). He is a project manager at elm.leblanc, in charge of developing a decision support tool. His research interest is in modeling, simulation, and optimization. His email address is mahdi.elalaouielabdellaoui@fr.bosch.com / mahdi.el-alaoui@emse.fr.

FRÉDÉRIC GRIMAUD, M.Sc. and Ph.D. in Computer Science, is Associate Professor at the Ecole des Mines de Saint-Étienne (France) and member of the Laboratory of Computer Science, Systems Modelling and Optimisation (LIMOS, UMR CNRS 6158). His research focuses on supply chain simulation and optimizations. His email address is frederic.grimaud@mines-stetienne.fr.

PAOLO GIANESSI is Associate Professor at the Ecole des Mines de Saint-Étienne (France) and member of the Laboratory of Computer Science, Systems Modelling and Optimisation (LIMOS, UMR CNRS 6158). His research focuses on Operations Research models for production systems and vehicle routing problems. His email address is paolo.gianessi@emse.fr.

XAVIER DELORME is Full Professor at the Ecole des Mines de Saint-Étienne (France) and Deputy Director of the Laboratory of Computer Science, Systems Modelling and Optimisation (LIMOS, UMR CNRS 6158). His research focuses on Operations Research models for production line design and supply chain optimization. His email address is delorme@emse.fr.

EMMANUEL BRICARD is IT director of elm.leblanc, French leader in thermal comfort and part of Bosch group. Engineer and holder of an Executive MBA, he heads the R&D department "Innovation Center for Operations" that focuses on Industry 4.0 technologies, from IoT to augmented reality. His email address is Emmanuel.Bricard@fr.bosch.com.