

Benchmarking Simulation Software Capabilities Against Distributed Control Requirements: FlexSim Vs AnyLogic

Ali ATTAJER^{1,2,3} **Saber DARMOUL**¹ **Sondes CHAABANE**² **Fouad RIANE**^{1,3} **Yves SALLEZ**²

¹ Ecole Centrale Casablanca, Bouskoura Ville Verte, 27182, Casablanca, Morocco
{ali.atajer, saber.damoul, fouad.riane}@centrale-casablanca.ma

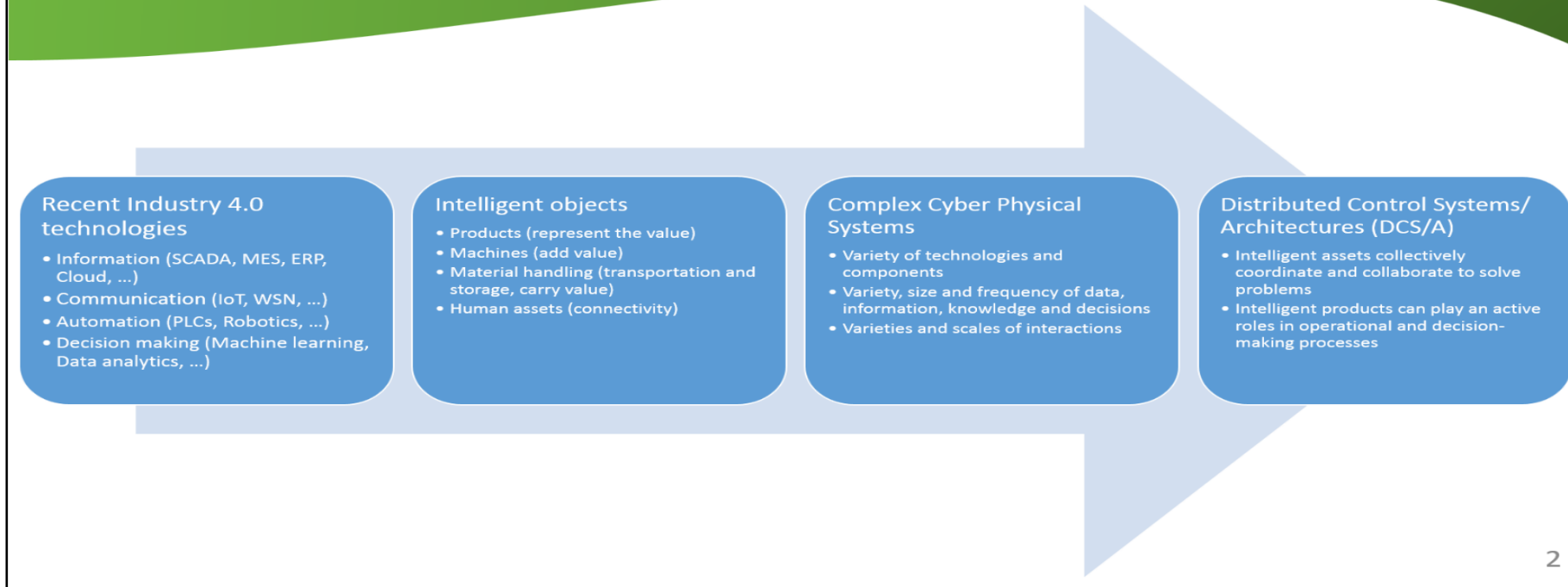
² LAMIH, UMR CNRS 8201, Université Polytechnique des Hauts-de-France, UPHF, Le Mont Houy, 59313 Valenciennes, France

{sondes.chaabane, yves.sallez}@uphf.fr

³ LIMII, Hassan 1^{er} University, Settati, Morocco

The thesis is done at the University Hassan Premier in Morocco and under a joint supervision arrangement with Polytechnic University Hauts de France. The thesis is funded by Ecole Centrale Casablanca and the project of PHC Toubkal. This documentary support aims to present the results of a study we did as part of our research project on managing risk disruptions in cyber physical systems. This is about Benchmarking Simulation Software Capabilities Against Distributed Control Requirements: FlexSim Vs AnyLogic software.

1. Introduction



2

The advent of the industry 4.0 paradigm introduces a set of information and communication technologies that allow the distribution and decentralization of information over several intelligent production entities, including smart manufacturing assets (machines, robots, material handling devices), augmented operators and intelligent products. The convergence of these technologies and intelligent objects has given rise to the paradigm of cyber-physical systems. These systems are characterized by their complexity due to the variety of components, size and frequency of data, information, knowledge and scales of interactions etc.

In this context, the distribution of information encourages the design and development of distributed, product driven control architectures, where intelligent products can play more active roles in operational and decision-making processes.

As manufacturers are often reluctant to experiment these new distributed control architectures directly on their production systems, mainly due to risk such as loss of production capacity, loss of quality and performance, etc.), they prefer first assessing the control architecture using simulation before implementing it on real scale.

2. Simulation in distributed control of manufacturing systems

Authors focus

More on assessing the behavior and performance of DCS/A

Less on the capabilities, ease of use, strengths and weaknesses of the simulation software they used

Simulation Software Survey

summarizes the main characteristics of a variety of simulation software packages available in the market

But characteristics do not specifically address DCS/A needs & requirements

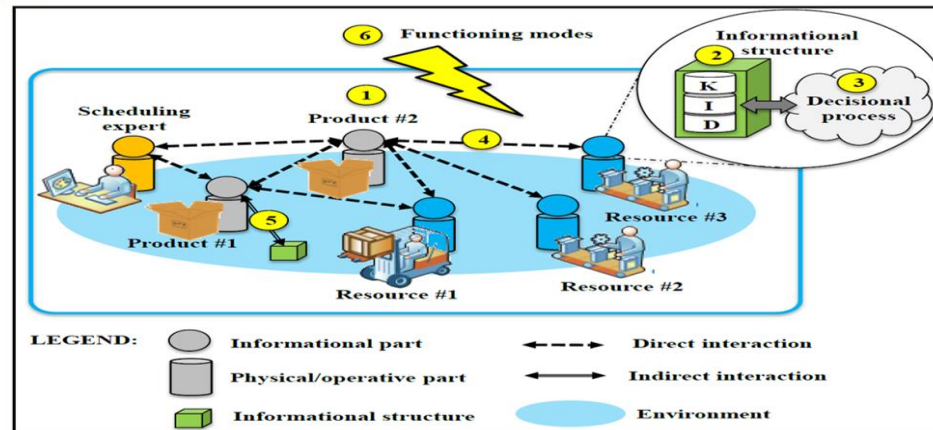
Benchmarking and ranking

Some initiatives for ranking

Some initiatives for benchmarking

Based on the literature survey that we conducted, Authors use often simulation to assess the behavior and performance of distributed control architectures. Most do not argue the selection criteria of the simulation software they use; Authors do discuss the strengths and weaknesses of their control architectures. However, they do not discuss the capabilities, ease of use, strengths and weaknesses of the simulation software they used; Simulation Software Surveys considered as a useful source of information to summarize the main characteristics of a variety of simulation software packages available in the market without specifically addressing DCS/A needs & requirements.

3. Requirements for simulation in distributed control



Production entities

Informational
structures

Interactions

Decision-making

Functioning modes

For a successful simulation and evaluation of product driven control architectures, the simulation software have to satisfy several types of requirements, as shown in the figure.

Production entities: Simulators shall be able to consider several features of entities such as the products, the resources and the decision entities;

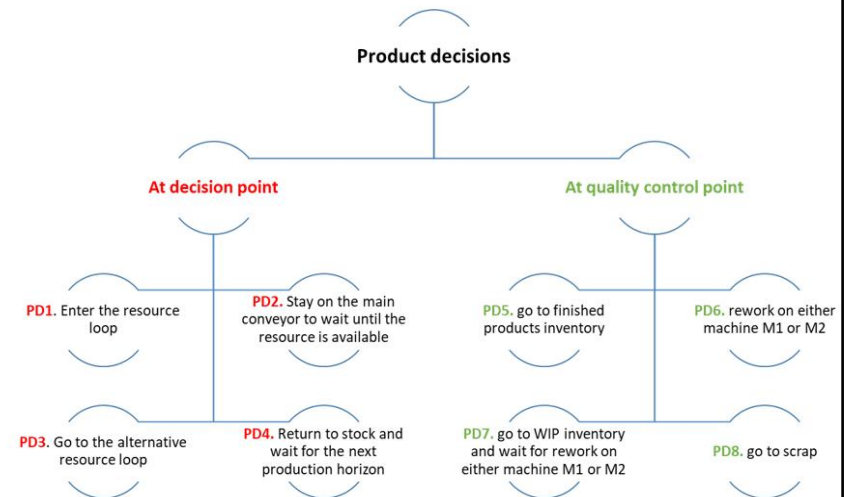
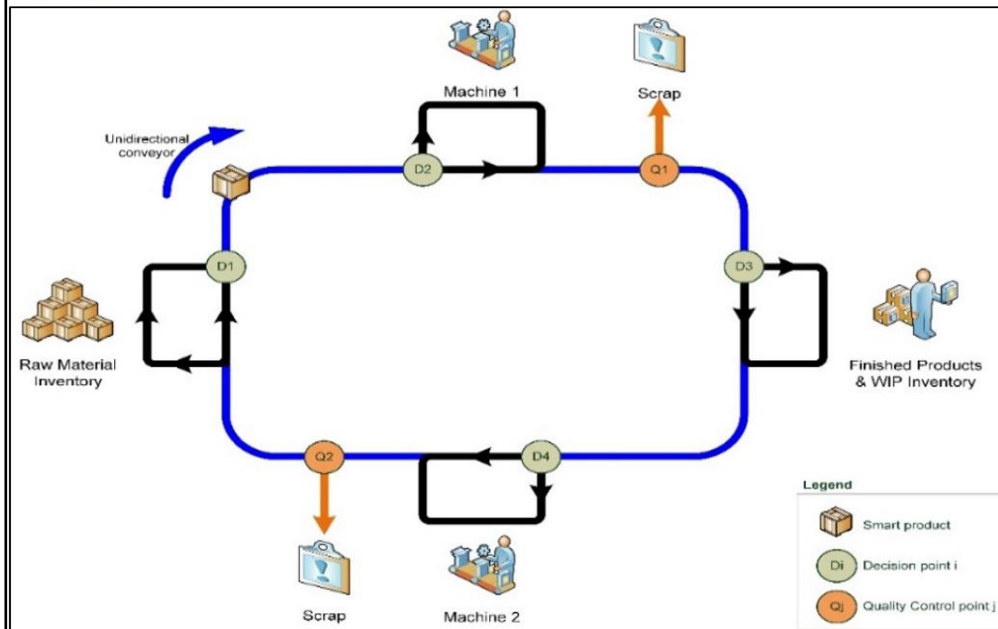
Informational structures: Simulators shall enable considering entity attributes and properties, related to product (bills of materials and routings), as well as indicators of manufacturing processes (key performance indicators, statuses and reports describing normal, and abnormal operating conditions);

Interactions: Simulators shall enable intelligent products to support different types of interactions with production entities for example using direct communication channels or using the environment and indirect communication channels (e.g. blackboards) or even more complex interactions, such as negotiation protocols, etc;

Decision-making: Simulators shall enable intelligent products to integrate different decision-making processes which lead to the satisfaction of production objectives;

Functioning modes: The products shall be aware of all operational settings of the manufacturing system, in terms of both normal and disturbed operational conditions.

4. Case study



The case study is built to evaluate the presented requirements. We consider an automated manufacturing system which is composed of a main unidirectional conveyor loop (shown in blue color) servicing production resources located aside secondary loops (shown in black color), and two scrapping areas (shown in orange color).

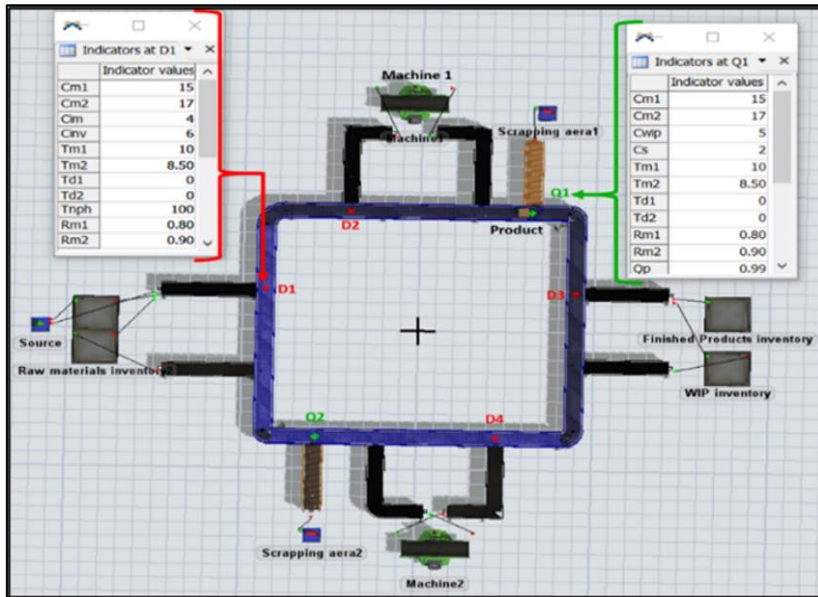
As the main goal of our study is not focused on complex product design and manufacturing, so we consider routing with only one operation to be performed interchangeably either on machine 1 or 2. Machines are subject to failures, and products are subject to quality defects. The scrap areas receive the products if they do not meet quality requirements. Decision and quality control points are located on the main conveyor loop as milestones so that intelligent products check updates about indicators and make decisions.

The product moves on the conveyor and crosses a given decision and quality control points, where it acquires its decision indicators. According to this data and using a decision mechanism which is the AHP mechanism (Analytic Hierarchy Process), the product selects the machine that will perform the next operation in its routing. The product can update its decisions based on indicators each time it crosses a decision or a quality control point.

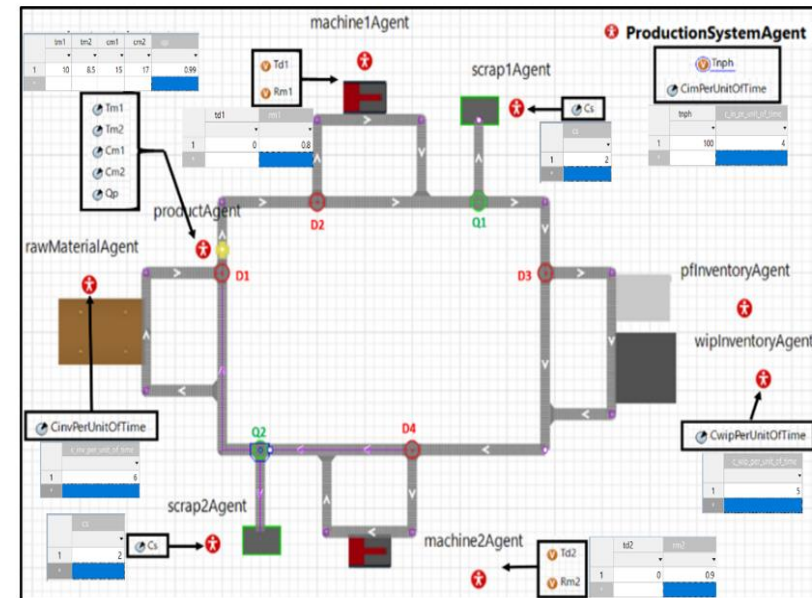
At decision and quality control points a product can make one among four possible decisions.

5. Capabilities benchmarked against requirements : FlexSim & AnyLogic

Simulation using FlexSim :



Simulation using AnyLogic :



The case study is implemented in FlexSim and AnyLogic to evaluate their capabilities to implement a product driven control model.

5. Capabilities benchmarked against requirements : FlexSim & AnyLogic



Production entities:

- A source node; 3 queuing objects; Conveyors; 2 production servers; 2 sink nodes.

Informational structures:

- Data is embedded in a flow item;
- Connection with external data sources MI,
- Exchange data using Open Database Connectivity (ODBC) connection;
- FlexSim defines specific indicators for each model object;
- All indicators are communicated by objects and stored;

Interactions:

- Sending direct messages on state conditions;
- Custom communication protocols can be programmed using either FlexScript or C++;

Decision-making:

- The AHP mechanism is implemented on decision and quality control points;

Functioning modes:

- Machine failures can be generated by the MTBF/MTTR fault profile.

The production entities: FlexSim offers a rich and user-friendly library containing simulation model objects that can be used to design simulation models quickly;

Informational structures: FlexSim allows different ways to store and process data and information;

Interactions: The flow of information between objects can be implemented by sending direct messages on state conditions. Custom communication protocols between objects can be programmed on the FlexSim using either FlexScript or C++;

Decision-making: in FlexSim, products are inert flow items, they cannot directly process information or do calculations, and consequently cannot be directly endowed with decision mechanisms. Therefore, AHP mechanism is implemented directly on decision and quality control points in outside of the product. The decision however cannot be made only if the product reaches the decision or quality control points;

Functioning modes: Machine failures can be generated by the MTBF/MTTR fault profile using probability distributions in the FlexSim “toolbox”.

5. Capabilities benchmarked against requirements : FlexSim & AnyLogic



Production entities:

- A source; 3 queuing agents;; A conveyor agent; 2Delay agents; 2 sink nodes; 4 decision points; and 2 quality control points.

Informational structures:

- AnyLogic assigns the value of the quality indicator using a set of statistical distribution functions;
- The ProductAgents can consult each time the other indicators from all agents.

Interactions:

- Agents living in one environment can directly communicate via sending messages to each other;
- All agents share the same ontology, and the use of an Agent Communication Language is not necessary.

Decision-making:

- AHP mechanism is implemented using the Java programming language;
- The mechanism is embedded directly on productAgents;
- Can work with R programming language to increase the data analytics capability.

Functioning modes:

- Machine failures can be generated by the "ResourcePool" block;
- Recurrent downtime is scheduled by the different types of triggers.

AnyLogic enables modeling all entities as agents using agent-based modeling.

Production entities: The Process Modeling Library of AnyLogic is used to build the agents of production entities;

Informational structures: The Product indicators are defined as productAgent(s)' related parameters since their creation. Each product knows its production cost and processing time and can consult each time the other indicators from all agents;

Interactions: Agents living in one environment can directly communicate via sending messages to each other. We can also use more complex communication such as negotiation protocols;

Decision-making: the AHP mechanism is implemented using the Java programming language. It is embedded directly on productAgents, and is only triggered when product agents reach decision or quality control points;

Functioning modes: Machine failures and downtime are scheduled by the different types of triggers or using the AnyLogic Schedule element.

6. Discussion and results

	AnyLogic	FlexSim
Production entities		
Products	😊	😞
Resources	😊	😊
Decision-making entities (human and/or artificial)	😊	😊
Interactions		
Product-Product	😊	😞
Product-Production resource	😊	😞
Product-Human	😊	😞
Production resource- Production resource	😊	😞
Functioning modes and disturbances		
Machine disturbances	😊	😊
Product disturbances	😊	😞
Intelligence level of the entities (except product for FlexSim)		
Associate informational structures with production entities (except product for FlexSim)	😊	😊
Decision-making (except product for FlexSim)	😊	😊

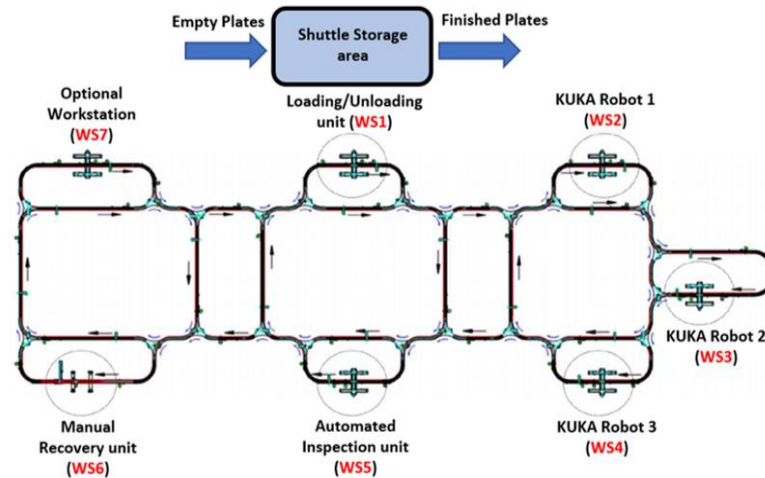
❑ AnyLogic is a consistent simulation software to implement and model a distributed product driven control in industrial context, due to:

- ❑ The conjunction of multi-agents, discrete event, and system dynamics ;
- ❑ Good interoperability with standard programming languages (Java, Python and R) ;
- ❑ Agent based modeling enables achieving product intelligence: data processing, communication, interactions and decision-making ;
- ❑ The ability to simulate the reconfigurable manufacturing systems.

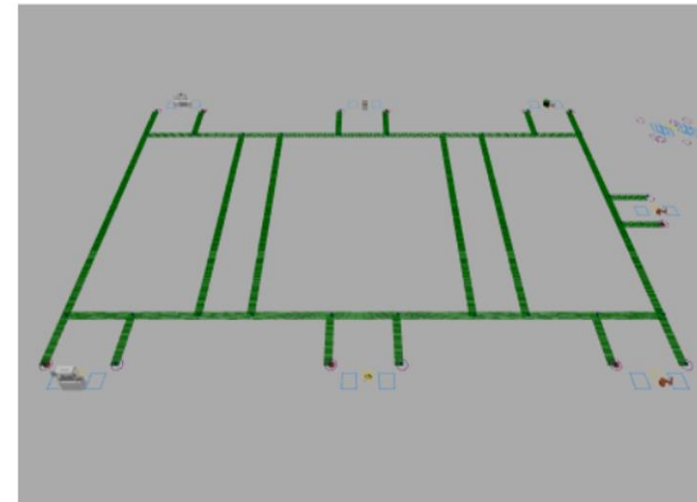
The results of the benchmark of the capabilities of the two software: According to the requirement we set in our study, Anylogic is the most suitable one for product driven control. AnyLogic is a consistent simulation software to implement and model a distributed product driven control in industrial context, due to the conjunction of multi-paradigm simulation such as the agent based modeling that enables achieving product intelligence in terms of data processing, communication, interactions and decision-making.

On the other hand, FlexSim is very strong in 3D animation and is characterized by its user-friendly interface and ease of use. FlexSim offers only discrete event simulation. In this type of simulation, products are represented and handled as flow items. This introduces limitations with respect to implementing product-based decision-making and interactions.

Future works



anylogic



10

We are considering an extension of the work to take into account several types of disruptions in the production environment (e.g., late delivery of raw materials, conveyor breakdown, etc.) in order to progress more on this work and further develop our model. Even several products can be interconnected (a network of products able to communicate with each other), in this case the products can share their experiences when they make a decision, and they can update the set of actions.