Generation of an adaptive simulation driven by product trajectories

Andres Véjar, Patrick Charpentier

Generation of an Adaptive Simulation Driven by Product Trajectories *

Andrés Véjar
Patrick Charpentier
Centre de Recherche en Automatique de Nancy
UMR 7039 CNRS-UHP, 54000 Nancy, France

February 21, 2011

Abstract

The tracking of products trajectories involves major challenges in simulation generation and adaptation. Positioning techniques and technologies have become available and affordable to incorporate more deeply into workshop operations. We present our 2-year effort into developing a general framework in location and manufacturing applications. We demonstrate the features of the proposed applications using a case study, a synthetic flexible manufacturing environment, with product-driven policy, which enables the generation of a location data stream of product trajectories over the whole plant. These location data are mined and processed to reproduce the manufacturing system dynamics in an adaptive simulation scheme.

This article proposes an original method for the generation of simulation models in discrete event systems. This method uses the product location data in the running system. The data stream of points (product *ID*, location, and time) is the starting point for the algorithm to generate a queuing network simulation model.
keywords Flow Location Product Simulation Modeling Trajectory

1 Introduction and Problem Statement

For several years now, new visions for manufactured goods tend to give them communication and sensory capacities within the framework of the intelligent product paradigm [31]. The informational part of each product is feed by the direct material environment of the physical product or by its own instrumentation (MEMS, GPS, Tags, etc.) The exchanges of information between the product and its environment can be made:

*The original publication (Journal of Intelligent Manufacturing) is available at `www.springerlink.com`

- At certain synchronization points using RFID technologies, bar codes, etc.
- In a quasi-continuous way (wireless networks such as Wifi, Zigbee and Bluetooth)

These communication technologies can also contribute to product localization. They are the subject of much research work aimed at identifying and/or improving architectures, services and communications of location systems. Location models are available for augmented media [26, 56], location-based services [69, 17], real-time mobility models [1], and moving-objects services [66]. There are many applications of these technologies in the field of production and logistics, some examples of which are product traceability [53], stock inventory [38, 28], and the positioning of a transport fleet [65]. Focusing only on Radio Frequency Identification Tags (RFID Tags), we will find RFID-enabled automation in factories [51], RFID-enabled supply chain [35] – delivery chain [34], or to support logistics processes [14]. Until now the spatial location of physical objects has been limited to voluminous objects (lorries, boats and containers) or to people.

The aim of our research consists of showing the new inputs and benefits provided by the use of a product location data stream during the manufacturing process. This article presents an application: the automatic generation of a flow simulation model on the basis of product location data during its passage through the production system. For several years, the simulation of product flow has been the main tool used to evaluate the dynamics of manufacturing systems, [11, 50]. However, it has often been shown that the modeling phase and the maintenance phase are constituted by delicate and time-demanding human operations [20, 43, 21]. These reasons explain the choice of this target problem. One of the ideas is to replace the largest amount of human expert interventions by an automatic generator in both the construction of the simulation model as well as in the maintenance and reconfiguration phases.

In this work, trajectories are the bridge between the manufacturing system, and the discrete-event simulation model. The new technologies make possible the mapping from the manufacturing system to trajectories, because we can obtain the location data of each moving object in the workshop. These data compose an information flow that can be assimilated as product trajectories. To construct the simulation model requirements include the following:

- Identifying a manufacturing system structure in the trajectories data stream.
- Collecting relevant information about flow rules and parameters needed for discrete event modeling.

This study is about the functions that map the trajectories to a simulation model of product flow (see Fig. 1). We proposed a methodology to obtain this function. As we work with manufacturing data, the mapping function will be capable of adapting the model to the exogenous variability. The inputs are the product trajectories and the output is an adaptive simulation model of the manufacturing system, (*i.e.*, an on-line recursive updating model).

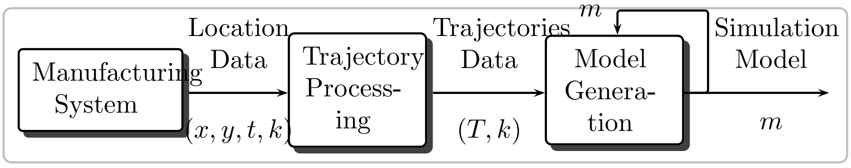


Figure 1: Generation Scheme: The problem is to make a generator that map the location data stream to a simulation model evolving synchronized with the evolution of the data stream (and the manufacturing system). The location data atom is composed by four components (x, y, t, k), the first two are the position in the (x, y) plane, the third component is the time t, and the last component k is a unique identifier of the moving object.

This simulation flow model is constituted by a transportation network with (buffer,machine) nodes. It differs from a queuing network because it is not a pure analytical stochastic model. It is necessary to join the physical infrastructure (transportation network), and reconfiguration capabilities into the model. This type of simulation flow model has a wide range of uses (e.g., controlling, monitoring, and hypothesis testing on off-line scenarios).

First, Section 2 is used to present the existing literature on approaches for automatic design of simulation flow model for manufacturing systems. This part also serves to present the case of modeling, outside of the field of manufacturing systems, using the notion of trajectories. Section 3 allows us then to introduce the modeling of the different elements needed to build the model generator.

2 Bibliographic Study

Until now there are not reported works about the generation of manufacturing simulation models based on trajectories. In this section, we present, for the first time, some references using location data with targets different from ours. Then, we focus on references aimed at assisting the modeler during the construction phases of the simulation models for flow manufacturing systems.

An example that uses trajectory information can be found in the case of animal behavior tracking in which tracks of marine animals in the wild, are determined by electronic tagging of individuals [24]. The goal is to identify Regions of Interest (ROIs): habitats and high-risk areas, and information about the behavior of these animals. The same ideas are used in our application: the ROIs are used to set the plant layout and information about the behavior of products is used to set the flow rules and parameters.

Geographic Information Systems (GIS) exploit the advantages of storing the relations between ROIs and attributes. The widespread use of GIS [15, 52, 58, 5] is the clearest example of the benefits of the use of spatial (and spatio-temporal) information.

Indoor localization using tags [13] is an application of identification and tracking technologies. For example, at construction sites, sensors and tags are incorporated into building materials. Emerging civil engineering applications explore the use of these tags for localization and construction support [10, 61] to track materials and their positions in real time to manage the construction flows.

One of the most advanced domains in the use of localization is mobile robots [6, 45]. Many techniques are used for tracking, navigation and location. For example there are applications using genetic algorithms [4], reactive agents [25] and artificial pheromones [30], wich are some of the most popular artificial intelligence tools. Simultaneous Location and Mapping (SLAM) is the name of the problem posed when a mobile robot is building a map of an unknown environment and needs to simultaneously navigate into the environment using this same map [39]. This problem is important in our case because it manages uncertainty in measurements to construct a formal model of the environment (the map) and to use this model to navigate. For manufacturing systems, the model of the environment is the layout that could be constructed on the basis of the uncertainty of sensor location data of the products.

We have shown some important aspects on location data found in the literature. The ideas in those works are the source of inspiration for our research. Next, we focus on discrete event simulation for manufacturing processes, and then we describe some tools for modeling support.

In manufacturing, as in most of the application areas, a simulation is very often handmade by humans. A discussion about the model formulation time began with [68] and has continued through the present. Simulation is not only modeling, it is studied as a rational process. The proposed methodologies for simulation have not changed in essence over several years. [3, 36, 41, 55, 46, 57]. A scheme that collects the commonly-used steps is showed in Fig. 2.

Some general concepts from the manufacturing simulation process can be located in the literature, from which we focus on: "Developing a Simulation Model", and in particular, on the steps "Collect and Process Real System Data", "Formulate and Develop a Model" and "Verify and Validate the Model". We limit the discussion to a brief description of two tools:

- Generic- or Template-based Simulation (GTS).
- Data-Driven Simulation tools (DDS).

GTS was selected because it is the most commonly known and used tool. DDS was selected because this tool is the closest to our work.

Generic or Template Based Simulation A GTS modeling approach is used for an available set of pre-built modeling objects of common simulation situations. These modeling objects can be simply customized to fit the current application by "switching on" or "switching off" the generic object parameters [27]. An important advantage of GTS is model reuse and is discussed in [62, 22, 40, 7]. In [60], the use of neutral libraries can expand the scope of applications of the GTS. Some authors also speak of *composable simulation* [32, 49] because the modeling can be reduced to the composition of simulation blocks.

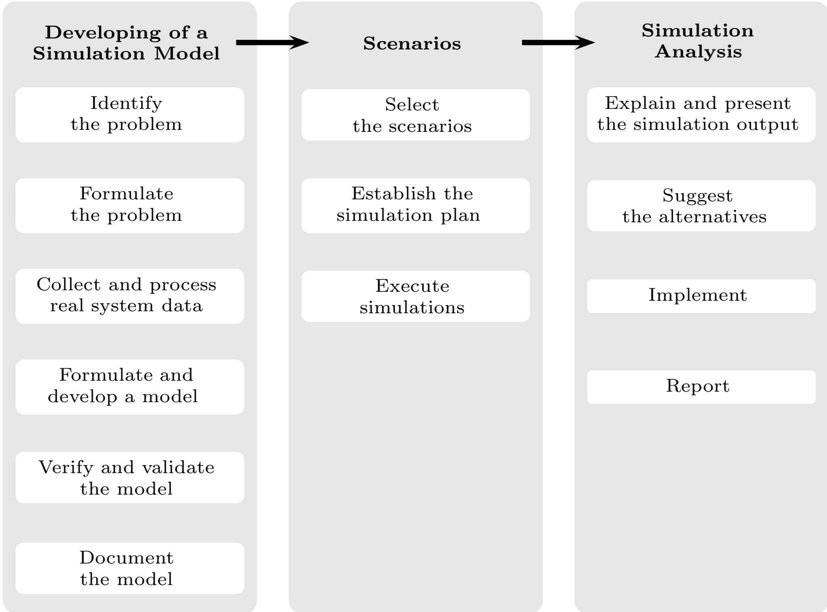


Figure 2: Manufacturing simulation process. Each column represent the more important steps : Modeling, Scenarios and Analysis.

Data-driven simulation tools In DDS, the model is generated automatically on the basis of existing data in the company databases. The model is updated when the data are changed. This approach differs from the traditional simulation approaches, where the human expert analyzes the problem and defines each step of the simulation modeling generally using a GTS tool [9].

Data-driven modeling applications like those for the management and control of water resources [59], or for product level decision support have been reported in the literature. However, the maturity of the concept is due to Darema [18, 19], who introduced "dynamic" DDS systems. The simulation applications can be capable of accepting new data during execution time and extending the measurement process to be controlled dynamically by the application. In [16], the authors present the DDS of a network using manufacturing blocking to control work-in-process. Yang [70] uses DDS to build an inventory model, and Kennedy [33] used DDS for policy decision support.

There are two main axes of this work: the location information of product flows, and the automatic generation and adaptation of simulations in manufacturing systems. We have presented a revision of some important aspects of location information and their impact in different application domains. The principal conclusion from the bibliographic revision, on the first axis, is that one of the more important characteristics of the spatio-temporal data is their capacity to encode structurally discrete relations of the underlying system, for example, by the spatial aggregation of data, the identification of ROIs, and the transportation network between ROIs [71].

For the second axis, we revise the modeling process in discrete event simulations for manufacturing systems. The knowledge acquired about the common problems under study allows the identification of templates and generic modules that have been added to modeling software in manufacturing simulations. The modeling process from the GTS perspective is to join, connect and customize these existing modules. The evolution of the modeling tools from raw code programming to GTS tools is the expression of a handmade reduction on the space of valid configurations. In addition to the support in modeling, the advances in the actual technology allow the following: the addition of automatic injection and processing of new data into applications capabilities; the update of models on-line; and achievement of a desired performance metric in measurements and simulations simultaneously. This type of approach, conceived by DDS, expands the support for modeling and maintenance processes in the simulation. The principal conclusion of the second axis of the literature review, is as follows: the capabilities exist to construct simulation models in manufacturing because there are common situations in modeling that can be approximated for generic structures (for example, a transportation network with client/server nodes) that can be discovered and updated with the available data sources.

In general, the fusion of the antecedents for the two axes (location + simulation) can explain the feasibility of the use of a location data stream for generating a simulation model in an adaptive scheme. We present the details of our construction in the next sections.

3 Simulation Generation and Adaptation

This section describes first a general class of simulation models for manufacturing, followed by the description of our method to transform a location data stream into an adaptive simulation model.

In today's manufacturing, the systems under study are growing, meaning a high quantity of heterogeneous subsystems and a non trivial relationship between them. That technical structure, can be better represented as a *complex network* [29] or as a *complex adaptive system* [44], avoid traditional interactive approaches for one-problem, one-human modeling schemes. From this perspective, the modeling of large production systems with traditional interactive front-end simulation engines has become unpractical, and it is increasingly true that in manufacturing *"simulation technology is not for everyone yet"* [42]. This work addresses these type of manufacturing systems.

A simulation model for manufacturing systems is generally targeted to study materials flows, and their processing relations that maps raw materials to finished products. There are two main processes in this kind of system: transport and product processing. Each product processing step can be modeled as a (buffer, machine) system.

The flow regime is conditioned by the transport of materials between processing steps and the parameters of each (buffer, machine) system. The main parameters are the service time of the machine and the arrival rate of the input flow, generally following probabilistic laws. In practice, a hypothesis of independence between processes is advised. This type of approach for modeling manufacturing systems is the most used and can be extended to fit a large range of situations (for example by means of GTS).

Manufacturing simulation generally deals with modeling the transportation network and the processing units. Our work is focused on identifying this type of model on the basis of location data of raw materials. The location data stream is composed of the position, the time, and a set of attributes.

To generate of the simulation model, we first identify the transportation network by product trajectories.

The product trajectories are identified by the aggregation of location data that follow the same path. The transportation network is generated for composition of the trajectories.

The identification of the machine nodes is made in the transportation network. In the network, it is possible to classify where the process is different from the normal transport by aggregation of spatio-temporal data in ROIs. If the data in the ROI have a "static" behavior in comparison with trajectories "transport" behavior, then the ROI is associated with a node (buffer, machine) in the transportation network.

With the identification of the transportation network and their (buffer, machine) nodes, the layout of the manufacturing system is completed.

A new product, unknown to the system, is identified by:

- its trajectory in the manufacturing plant
- parameters of the machines that are on its trajectory

A product is considered as a new product if it generates a trajectory where the trajectory parameters are significantly different from those of known products.

With the identification of the products, all of the elements for the generation of the simulation model are available.

As the data stream evolves by exogenous variability, the simulation model can be adapted to explain the new data. Each parameter of the model is updated on-line.

The probabilistic laws are estimated recursively for all of the data collected in a slicing window procedure. The data of the slicing windows are updated at each data point.

Each part of the simulation structure is linked to a value that represent the recurrence of the data to fit this structure. This value represents the memory of validation of one modeling object. The validation value is associated with the number of times that the data confirm the structure as a manufacturing "long-term" object. In general, the structures that can not explain the new data will be updated with a decreasing validation value. When the validation value reaches its minimum, the structure is deleted. If the new data fits the structure, the validation value is increased.

Only the trajectories and machines that are validated for the new data are maintained.

After presenting the whole process of automatic generation of simulation model, we make some decisions in the next section about useful objects for the generation process.

4 Modeling

In this section, we present the modeling of products, the location data, the trajectories data and the adaptive scheme in the generation of simulation models.

4.1 Product

The trajectory of one product can be viewed as a way in which to describe its production process. An important problem is how to encode the product processing steps in the space. One product can be viewed as the application of one operation on some raw materials, where the operation is the mapping from materials to outputs:

$$\text{raw materials} \xmapsto{\text{operation}} \text{product} \tag{1}$$

The product is often the result of more than one operation. Because each operation can be composed of sub-operations, then locally we have intermediate products that interact to create others intermediate products:

$$\begin{aligned} &\text{raw materials} \xmapsto{\text{operation}_0} \text{product}_1 \cdots \\ &\text{product}_{i-1} \xmapsto{\text{operation}_{i-1}} \text{product}_i \xmapsto{\text{operation}_i} \text{product}_{i+1} \\ &\cdots \text{product}_{n-1} \xmapsto{\text{operation}_n} \text{product}_n \end{aligned} \tag{2}$$

To allow the representation of processes that are not sequential, such as assembly and disassembly processes, we can model a general product as a directed graph $G = (V, A)$ where nodes in V are of two classes, $V = V_o \cup V_i$. A node in V_o will be on the outside of the process such that the possibilities for this node are as an input of raw materials or an output of a finished product. A node in V_i will be inside of the process, and will represent some operation. The intermediate products are represented by the arcs in A.

In this representation, each type of final product is linked with one graph.

Examples of operations modeled by graphs are as follows: Single Input Single Output, Assembly, and Disassembly.

Single Input Single Output One input, one output, one operation.

$$\begin{aligned} V &= \{\text{in}, \text{op}, \text{out}\} \\ A &= \{(\text{in}, \text{op}), (\text{op}, \text{out})\} \end{aligned} \tag{3}$$

Assembly Two inputs, represented by the operations in_1, in_2, the assembly operation op_A, and the output node out.

$$\begin{aligned} V &= \{\mathrm{in}_1, \mathrm{in}_2, \mathrm{op}_A, \mathrm{out}\} \\ A &= \{(\mathrm{in}_1, \mathrm{op}_A), (\mathrm{in}_2, \mathrm{op}_A), (\mathrm{op}_A, \mathrm{out})\} \end{aligned} \tag{4}$$

Disassembly Two outputs, represented by the operations out_1, out_2, the disassembly operation op_D, and the input node in.

$$\begin{aligned} V &= \{\mathrm{in}, \mathrm{op}_D, \mathrm{out}_1, \mathrm{out}_2\} \\ A &= \{(\mathrm{in}, \mathrm{op}_D), (\mathrm{op}_A, \mathrm{out}_1), (\mathrm{op}_A, \mathrm{out}_2)\} \end{aligned} \tag{5}$$

In general, if each operation has p inputs and q outputs, when $p = q$, we have a single input single output operation; when $p < q$, it is a disassembly operation; and when $p > q$, it is an assembly operation.

Note that if one product is modeled as a graph, disassembly processes can be followed by assembly processes to reach only one ending node of the finished product. The waste or unused materials that begin as a part of product, and are disassembled after use, are modeled in another graph as undesired products and will also be finished in one node.

For each node in V, we make the association of a position in the (x, y) plane, by means of a function $f : V \to \mathbb{R}^2$ that encode the production process to the space.

4.2 Location Data and Trajectory Data

One element of the location data stream is composed of the position r in the (x, y) plane, the time t, and a set of attributes a. Because of other disturbances, the location data could be degraded by noise. In our work, we use only one attribute, a unique object identifier, which is assigned before the input of the object in the system until its release. The accessible location information is defined by the 3-tuple (R, T, I) where I is the set of *Ids* for each object. The set R represents the positions (x, y), and $T > 0$ represents the time. Each $d = (r, t, i) \in R \times T \times I$ is an element in the data stream. The data acquisition period can be fixed discrete or by events.

The trajectory data are the paths followed by a moving object, for a moving object with *Id* i, its trajectory will be $T(i)$ and represents the sequence of observations $\{(r_k, t_k)\}$ of the object i:

$$T(i) = \{(r_1, t_1), (r_2, t_2), \ldots\} \tag{6}$$

4.3 Generation of Adaptive Simulation

The location data stream $D = \{d_1, d_2, d_3, \ldots\}$ is processed to split the trajectories of each moving object, and then the trajectories are processed to generate

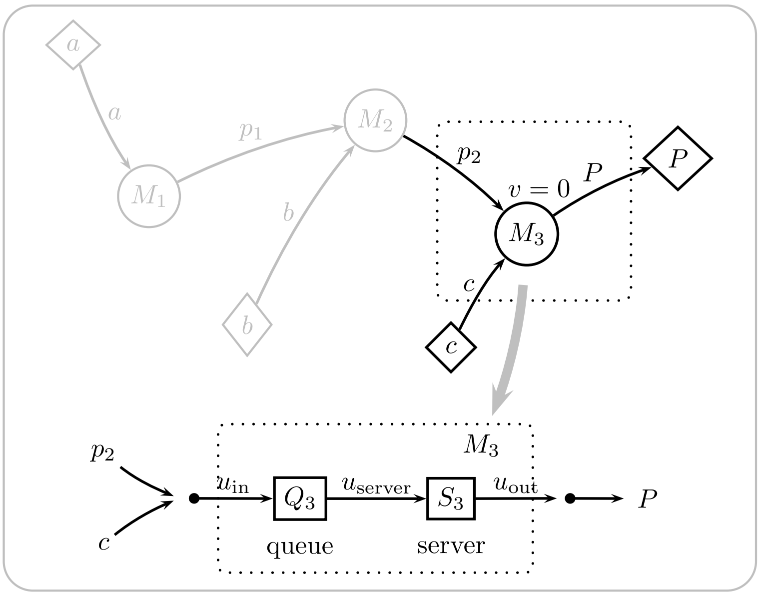


Figure 3: Service point $v = 0$ in the server node M_3 and its association with a waiting queue behavior

the simulation model, as shown in Fig. 1. When new data arrive at the generator system, the model is updated to modify the simulation model parameters. If new trajectories are identified, the model structure is adapted to reflect the new scenario.

Our problem consists of conceiving a generator of simulation models on-line F, capable of developing a model m from the location data stream D. The generator F allows adaptation of the model to reflect the modifications of the real system: the generator can react to reflect the changes (7).

$$m_k = F(d_k, m_{k-1}), \quad m_0 = \emptyset \tag{7}$$

Figure 3 shows a simple product process modeled by the graph $G = (V, A)$, with

$$\begin{aligned} V &= \{a, b, c, M_1, M_2, M_3, P\} \\ A &= \{(a, M_1), (M_1, M_2), (b, M_2), (M_2, M_3), (c, M_3), (M_3, P)\} \end{aligned} \tag{8}$$

This example is used to explain the process of generating a simulation model described in Section 3.

The monitoring of the trajectory of an object in the plane (x, y), in time, allows us to determine the speed v: either it is zero, or it is positive (4). The product is moving or is stopped: the latter property is a sign of a waiting time, hence the choice to build a simulation model based on a queuing network. The

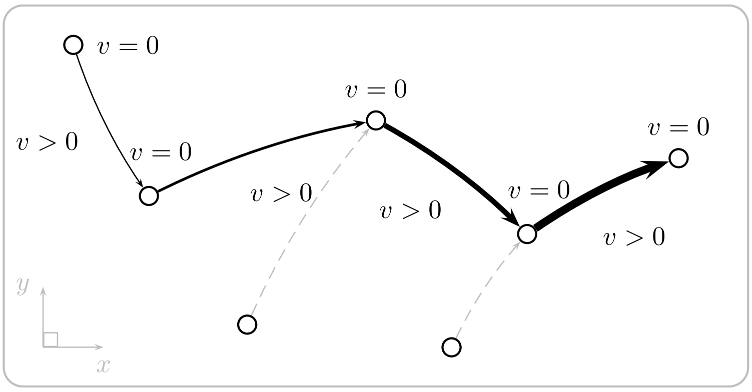


Figure 4: Speed component a, in the network of the joint movement of another components of the product

information $v = 0$ locates a particular point at the base of the construction of our model. Because of each of these points, it is possible to obtain a histogram that represent the service time for each product type. However, before the disappearance of a product, no information on its composition is available. The objects (initials and intermediaries) provide no information about their final destination. The composition of the final product is discovered when it disappears from the system. Then is possible to trace all of the pathways of its constituents to the moments of their birth (by the joint movement of the components, as in Figure 4) and to update the histograms of the service time of a part for a final product at a point $v = 0$.

We associate a waiting queue behavior at each point where $v = 0$ (See Figure 3).

At these points, two types of events may occur: the arrival or the departure of object(s). As the composition of a product can be known only at the output of this waiting queue, the object's constituents of the product then follow the same trajectory. Then, we define (See Figure 5):

- e_1 the instant of arrival of the first object that constitutes the product.
- e_2 the instant of arrival of the last object that constitutes the product.
- s is the common instant of departure all objects that are constituents of the product.

On this basis, we can obtain:

- O, the time between the arrival of the first object and the arrival of the last object needed for the product constitution.
- A, the waiting time for service: defined as the period between the time when all of the required pieces have arrived, and the instant when the service starts.

- T, the service time at point $v = 0$.
- T_{Tot}, the total waiting time at point $v = 0$.

We can link these different variables in the equation:

$$T_{Tot} = O + A + T \tag{9}$$

To compute A, it is necessary to know the departure instant of the previous product s^{k-1} where $k-1$ is the order of departure of the products. The rule for calculating A is: if $e_2^k < s^{k-1}$, then the waiting time is $s^{k-1} - e_2^k$, else the waiting time is 0.

In a more formal manner and using the Heaviside function:

$$\theta(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0, \end{cases} \tag{10}$$

Now we can define at each product departure k, the variables introduced so far:

$$\begin{array}{rcl} y^k & = & e_2^k - s^{k-1} \\ \alpha^k & = & \theta(y^k) \\ \beta^k & = & 1 - \alpha^k \\ T^k & = & \alpha^k(s^k - e_2^k) + \beta^k(s^k - s^{k-1}) \\ A^k & = & \beta^k(s^{k-1} - e_2^k) \\ O^k & = & e_2^k - e_1^k \end{array} \tag{11}$$

with $s^0 = 0$.

The 3-tuple (T, A, O) has all the necessary information to characterize the point $v = 0$ as a service point.

We must note that the analysis presented here is valid only under the theoretical condition that the queue is reduced to one logical point (all objects waiting at the same region of interest).

5 Algorithm

The algorithm proposed here is naturally the generator of the simulation code. It comprises three main parts. The first part generates the machine positions and the paths followed by products on the basis of the real-time data stream. The second part deals with the issue of product departures in the system. The product departure is the instant when it is possible to know whether its type existed before or not. If the type existed, the algorithm updates the product data, otherwise, a new product type is created. The third part is used to model the laws of behavior for the simulation model. Statistical laws are proposed to represent the inter-arrival time and service time for each product type at each machine.

The algorithm associated with the first part is triggered with every change in the flow. At each new data point (r, t, i) a new product is created if the

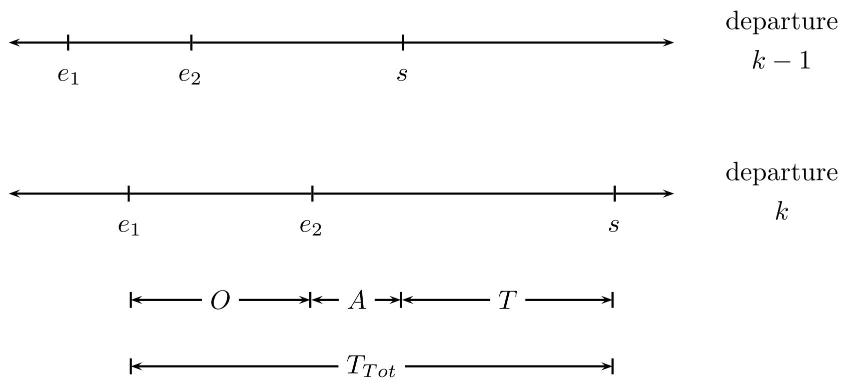


Figure 5: Waiting periods at congestion point $v = 0$ (server) for a product with departure k. O is the time waiting for raw materials, A is the time waiting for service and T is the time in service

Table 1: Data structure m

var	description	defaut
$m.r$	position (x, y)	none
$m.t$	current time	none
$m.s$	binary for stop condition	0
$m.T$	service time	0.0
$m.l$	departure instant	0.0
$m.a$	stop time	0.0

Table 2: Data structure p

var	description	defaut
$p.i$	*id*	none
$p.m$	structure of type m	none
$p.M$	ordered set of m structures	$\emptyset$

data *id* is not known. At each stop of this product, a *stop point* (location point of a queue and/or service point) is created, and a link between product and the *stop point* is created. This *stop point* is validated if the position of the product i is in the same position between the instants t and ts, $t < ts$, then the point corresponds to a real waiting point. In this case, the time of arrival and departure of the product i, at the waiting point located in r, are retained. They are used for modeling the dynamic behavior of this type of product at this point.

The algorithm associated with the second part is triggered at each product departure of the system. Whether a product comes out of the system may be a difficult problem to solve. We have chosen to define a product as out of the system if the identifier i no longer appears in the data flow during a considerable period (chosen arbitrarily but large enough). As we have already explained in the Equation (8), and in Figure (3) it must be possible to know only the type of product at the exit. In this level, we use the adjacency matrix of the graph linked to each elementary product. The sum of these matrices to each elementary product that departs out of the system at the same space-time instant can return the composition of the final product. The algorithm compares this sum of matrices with those obtained previously. If this matrix does not exist, it means that a new product came out of the system. Then it connects the information on its service (or stop) points obtained during the movements of its various components in the workshop.

The third part is made to characterize the stochastic behavior of different stop points. This third and final phase is the phase of post-treatment of all the information collected in phases 1 and 2. An estimation of the probability density function for the inter arrival time and the service time is driven using kernel density estimators. It is then validated by a Wilcoxon test [67].

We present (Algorithm 1) a fragment of the algorithm developed in its simplified version. This fragment can collect the positions of stop points for each product, and their inter-arrival times and service times at each stop point.

In the presented algorithm, a point of data (i, r, t) is defined by the structure d, where $d.i$ is the *id*, $d.r$ is the (x, y) position, and $d.t$ is the time.

Other data structures used are m and p. The structure m is defined in the Table (1), and the function to create a new structure is $\breve{m}(r, t)$.

The structure p is defined in the Table (2), and the function to create a new structure is $\breve{p}(i, m)$.

In addition to the structures d, m, p , there are two global sets M and P, empties by default.

6 Application & Validation

This section is intended to validate the principle of feasibility of the automatic simulation code generation on the basis of location information.

Flexible Manufacturing Systems (FMSs) are one of the most used models for testing new hypothesis and paradigms in production research, principally for their combinatorial capabilities that practically forbid the use of traditional centralized and optimal approaches for scheduling and control (see [37, 64, 12]). Then, to test our simulation application, we generated a FMS simulation module using SimPy (Simulation in Python) the object-oriented, process-based discrete-event simulation language [47, 48, 2]. In its FMS module, we generated the product location data stream. An example of the simulation output is shown in Figure 6. It acts as an artifact of the real system. We limited our application case here voluntarily to a simple case to limit the volume of its presentation

Algorithm 1 Simplified algorithm

h!

for d **do**
  if $\exists\, p \in P\colon p.i = d.i$ **then**
    if $p.m.s = 1$ **then**
      if $p.m.r = d.r$ **then**
        $p.m.a \leftarrow p.m.a + d.t - p.m.t$
        $p.m.t \leftarrow d.t$
      else
        $m \leftarrow m' \in M\colon m'.r = p.m.r$
        if $p.m.t - p.m.a < m.l$ **then**
          $p.m.T \leftarrow p.m.t - m.l$
        else
          $p.m.T \leftarrow p.m.a$
        end if
        $m.l \leftarrow p.m.t$
        $p.M \leftarrow p.M \cup \{p.m\}$
        $p.m \leftarrow \breve{m}(d.r, d.t)$
      end if
    else
      if $p.m.r = d.r$ **then**
        $p.m.s \leftarrow 1$
        if $\nexists\, m \in M\colon m.r = d.r$ **then**
          $M \leftarrow M \cup \{p.m\}$
        end if
      else
        $p.m.r \leftarrow d.r$
      end if
      $p.m.t \leftarrow d.t$
    end if
  else
    $m \leftarrow \breve{m}(d.r, d.t)$
    $p \leftarrow \breve{p}(d.i, m)$
    $P \leftarrow P \cup \{p\}$
  end if
end for

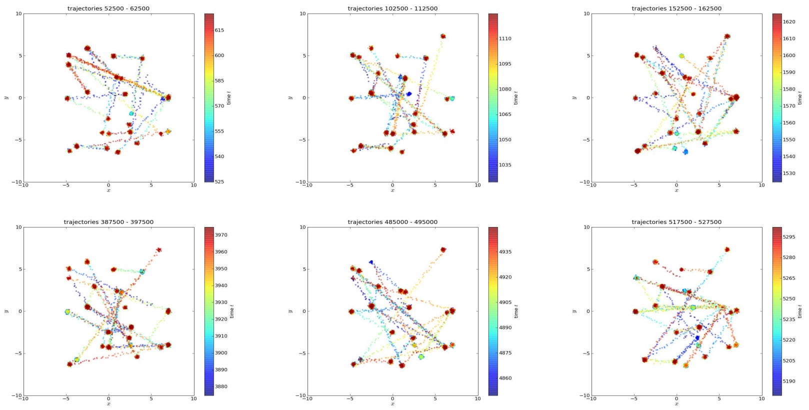


Figure 6: Location data stream snapshots in a FMS Simulation. Each dot is a location data point, the high density circle nodes are identified as servers

Table 3: Experimental conditions

parameter	value
product types	3
machines	3
operation for each product	3
products	100
inter arrival time	exponential, $\mu = 1/3$
service time	exponential, $\mu = 13$
AGVs	10
AGVs speed	0.44
workshop limits	$(-10, -10), (10, 10)$
location sampling period	0.5

and analysis of results. The location data stream is then retrieved by the generator module of the simulation model, which generates the simulation model by implementing phases 1 and 2, as described before. Finally, a module for data analysis (phase 3) allows verification of the results. All of the modules are programmed in Python [54, 23, 8].

6.1 Presentation of the chosen application

The workshop modeled here is composed of different machines among which the products are evolving, transported by Automated Guided Vehicles (AGVs).

The different types of products are generated for the formula:

$$mach(s, l) = (s - l) \mod M \tag{12}$$

Table 4: Wilcoxon test, inter arrival time

Product	Operation		
0	0	1	2
t-statistic	168.5	138.0	131.0
two-tailed p-value	0.7	0.0	0.0
1	0	1	2
t-statistic	96.0	5.0	124.0
two-tailed p-value	0.49	0.0	0.0
2	0	1	2
t-statistic	240.5	195.5	192.5
two-tailed p-value	0.73	0.0	0.0

- l: product type, $l = 0, \dots, L-1$,
- s: operations, $s = 0, \dots, S-1$,
- M :quantity of machines,

based on the original formulation proposed by [63]. The proposed formula is used to generate sequences of processing for each product type.

- The inter-arrival times of products in the workshop are provided by an exponential distribution. Each product type is generated in the same proportion.
- The service time of each machine is also provided by an exponential distribution function.
- The machine positions are carried out randomly between the dimensions of the system at the beginning of the simulation.
- The tracking system allows to obtain the determination of location information of Product (i, r, t) with a fixed frequency.

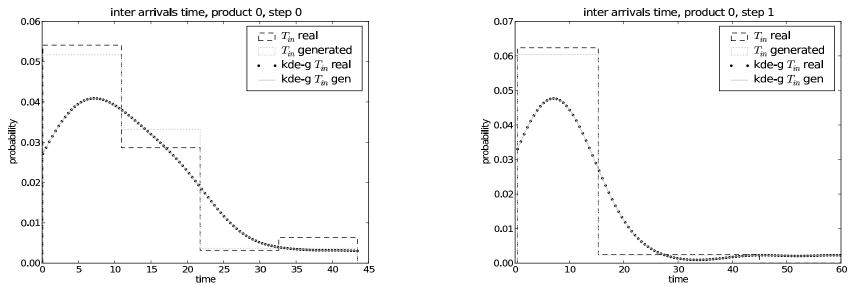
The parameters of the simulation used in the implementation with SimPy are expressed in the Table 3.

6.2 Results

The results provided by the model from the code generator are quite consistent with those obtained from the artifacts of the real system. In fact, the Wilcoxon test, shows that the data of inter-arrival time and service time generated by the two models are distributed under the same statistical law (Tables 4 and 5). The products and their types are fully identified by the code generator. It also sets out in detail the different resources implemented for the development of products. These early results show the feasibility of the proposed method. They were confirmed by tests on larger systems (up to 15 machines, 15 operations, and 15 product types).

Table 5: Wilcoxon Test, service time

Product	Operation		
0	0	1	2
t-statistic	220.0	187.0	185.5
two-tailed p-value	0.8	0.35	0.3
1	0	1	2
t-statistic	154.0	175.0	166.0
two-tailed p-value	0.07	0.15	0.17
2	0	1	2
t-statistic	261.0	270.0	309.0
two-tailed p-value	0.07	0.09	0.37



h

Figure 7: Inter-arrival time. T_{in}, distribution for two steps of a product for the experimental conditions presented in Table 3. The histogram of the original data (T_{in} real) is represented by the dashed line. The histogram regenerated by the application, for the same product and process step, is represented by the dotted line (T_{in} generated). Continuous estimation of the probability density function for the real case (kde-g T_{in} real, dots) and the generated case (kde-g T_{in} gen, solid line) are made by the non-parametric method of kernel density estimation.

h

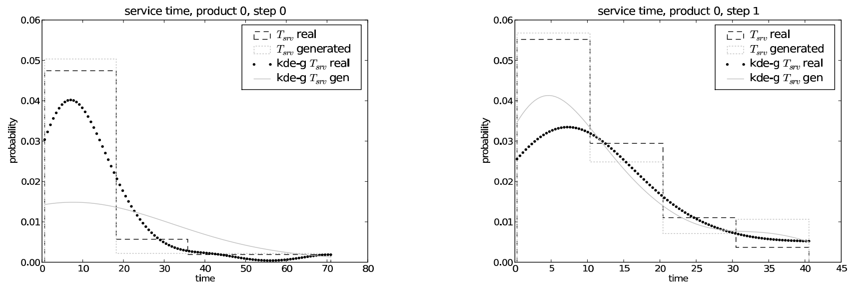


Figure 8: Service time. T_{srv}, distribution for two steps of a product for the experimental conditions presented in Table 3. The histogram of the original data (T_{srv} real) is represented by the dashed line. The histogram regenerated by the application, for the same product and process step, is represented by the dotted line (T_{srv} generated). Continuous estimation of the probability density function for the real (kde-g T_{srv} real, dots) and the generated case (kde-g T_{srv} gen, solid line) are made by the non-parametric method of kernel density estimation. Note that the estimation with kde is less accurate for service time than for the inter-arrival time, but the histogram estimation is robust.

7 Conclusions

In this paper, we have presented the use of product trajectories for generation of adaptive manufacturing simulation models. The time of model development is eliminated because the simulation model obtained is updated automatically to be adapted to new exogenous influences.

The simulation model has a generic simulation structure that is constituted by a transportation network with (buffer, machine) nodes. This resulting model is neutral, and it will be used for different purposes (controlling, monitoring, hypothesis testing on off-line scenarios, etc.). As the model is constructed by the spatio-temporal data stream, the reliability of the model and its output data can be tracked and controlled.

The feasibility of this approach is verified with a practical implementation *in silico* using a model of reconfigurable Flexible Manufacturing System. The data acquisition is constrained to technological advances for autonomous self positioning of each moving object. In real life applications, the simulation model generation scheme is best suited to manufacturing processes with strong spatial behavior. From this perspective, more high-level behavior, such as scheduling rules, can be identified and incorporated into the model.

The modeling formalism of a transportation network with (buffer,machine) nodes is easily mapped to the (process based) simulation engine SimPy. Further work includes mapping to a discrete-event formalism, Discrete Event Specification (DEVS), allowing mapping to different simulation engines to parallel or grid computing environments.

References

[1] Mohammed ALRahmawy and Andy Wellings. A model for real time mobility based on the RTSJ. In *JTRES '07: Proceedings of the 5th international workshop on Java technologies for real-time and embedded systems*, pages 155–164, New York, NY, USA, 2007. ACM Press.

[2] Alex Bahouth, Steven Crites, Norman Matloff, and Todd Williamson. Revisiting the Issue of Performance Enhancement of Discrete Event Simulation Software. In *40th Annual Simulation Symposium (ANSS{'}07)*, volume 0, pages 114–122, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[3] Jerry Banks and John S Carson II. Applying the simulation process. In *WSC {'}88: Proceedings of the 20th conference on Winter simulation*, pages 52–55, New York, NY, USA, 1988. ACM.

[4] Momotaz Begum, George K.I. Mann, and Raymond G. Gosin. Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots. *Applied Soft Computing*, 2008.

[5] Philippe Bonnifait, Maged Jabbour, and Gérald Dherbomez. Real-Time Implementation of a GIS-Based Localization System for Intelligent Vehicles. *EURASIP Journal on Embedded Systems*, 2007, 2007.

[6] G A Borges and M J Aidon. Design of a Robust Real-Time Dynamic Localization System for Mobile Robots. In Michel Devy and Lerasle Frédéric, editors, *SIRS 2001 : proceedings of the 9th international symposium on intelligent robotic systems*, pages 425–434. LAAS-CNRS, 2001.

[7] N Brown and S Powers. Simulation in a Box: (A Generic Reusable Maintenance Model). In *Proceeedings of the 2000 Winter Simulation Conference*, pages 1050–1056, 2000.

[8] X. Cai. On the performance of the Python programming language for serial and parallel scientific computations. *Scientific Programming*, 13(1):31–56, 2005.

[9] Bing Cao, Richard Farr, Mike Byrne, and James Tannock. Data-driven Simulation of the Extended Enterprise. *18th International Conference on Production Research*, 2004.

[10] Francois Caron, Saiedeh Navabzadeh Razavi, Jongchul Song, Philippe Vanheeghe, Emmanuel Duflos, Carlos Caldas, and Carl Haas. Locating sensor nodes on construction projects. *Auton Robot*, 2007.

[11] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Events Systems.* Kluwer Academic Publishers, Dordrecht, 1999.

[12] Felix T. S. Chan and H. K. Chan. A comprehensive survey and future trend of simulation study on fms scheduling. *Journal of Intelligent Manufacturing*, 15:87–102, 2004. 10.1023/B:JIMS.0000010077.27141.be.

[13] Lun-Chi Chen, Ruey-Kai Sheu, Hui-Chieh Lu, Win-Tsung Lo, and Yen-Ping Chu. Object Finding System Based on RFID Technology. In Heng Tao Shen, Jinbao Li, Minglu Li, Jun Ni, and Wei Wang, editors, *APWeb Workshops: XRA, IWSN, MEGA, and ICSE, Harbin, China, January 16-18, 2006, Proceedings*, volume 3842, pages 383–396. Springer, 2006.

[14] Harry K. H. Chow, K. L. Choy, and W. B. Lee. A dynamic logistics process knowledge-based system - An RFID multi-agent approach. *Know.-Based Syst.*, 20(4):357–372, 2007.

[15] R. L. Church. Geographical information systems and location science. *Computers and Operations Research*, 29(6):541–562, 2002.

[16] G M Clark and C R Cash. Data-driven simulation of networks with manufacturing blocking. In *Proceedings of the 1993 Winter Simulation Conference*, 1993.

[17] Antonio Coronato, Giuseppe De Pietro, and Massimo Esposito. A Semantic Location Service for Pervasive Grids. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, volume 4278, pages 1274–1284. Springer, 2006.

[18] F Darema. Dynamic Data Driven Application Systems (Symbiotic Measurement & Simulation Systems), "A new paradigm for application simulations and a new paradigm for measurement systems". NSF sponsored Workshop, 2000.

[19] F Darema. Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements. In *Proceeding of the 2004 International Conference on Computational Science*, pages 662–669, 2004.

[20] L. J. De Vin, A. H. C. Ng, and J. Oscarsson. Simulation-Based Decision Support for Maunufacturing System Life Cycle Management. *Journal of Advanced Manufacturing Systems*, 3:115–128, 2004.

[21] L J De Vin, A H C Ng, J Oscarsson, and S F Andler. Information Fusion for Simulation Based Decision Support in Manufacturing. *FAIM 2005 Special Issue of Robotics and Computer Integrated Manufacture*, 22:429–436, 2006.

[22] A Diaz-Calderon, C Paredis, and Khosla P. Architectures and Languages for Model Building and Reuse: Organization and Selection of Reconfigurable Models. In *Proceeedings of the 2000 Winter Simulation Conference*, pages 386–393, 2000.

[23] A. Downey, J. Elkner, and C. Meyers. *How to Think Like a Computer Scientist: Learning with Python*. Green Tea Press, 2002.

[24] P. Gaspar, J.-Y. Georges, A. Lenoble, S. Ferraroli, S. Fossette, and Y. Le Maho. Marine animal behaviour: neglecting ocean currents can lead us up the wrong track. *Proc. R. Soc. B.*, 273(1602):2697–2702, 11 2006.

[25] Franck Gechter, Vincent Chevrier, and François Charpillet. A reactive agent-based problem-solving model: Application to localization and tracking. *ACM Trans. Auton. Adapt. Syst.*, 1(2):189–222, 2006.

[26] Joachim Goßmann and Marcus Specht. Location Models for Augmented Environments. *Personal Ubiquitous Comput.*, 6(5-6):334–340, 2002.

[27] Ashu Guru and Paul Savory. A Template-Based Conceptual Modeling Infrastructure for Simulation of Physical Security Systems. In R G Ingalls, M D Rossetti, J S Smith, and B A Peters, editors, *WSC'04: Proceedings of the 36th conference on Winter simulation*, pages 866–873, 2004.

[28] Jiawei Han, Hector Gonzalez, Xiaolei Li, and Diego Klabjan. Warehousing and mining massive rfid data sets. In Xue Li, Osmar Zaïane, and Zhanhuai Li, editors, *Advanced Data Mining and Applications*, volume 4093 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin / Heidelberg, 2006.

[29] Dirk Helbing, Dieter Armbruster, Alexander S. Mikhailov, and Erjen Lefeber. Information and material flows in complex networks. *Physica A: Statistical Mechanics and its Applications*, 363(1):xi – xvi, 2006. Information and Material Flows in Complex Networks, Information and Material Flows in Complex Networks.

[30] Herianto, Toshiki Sakakibara, and Daisuke Kurabayashi. Artificial Pheromone System Using RFID for Navigation of Autonomous Robots. *Journal of Bionic Engineering*, 4(4):245–253, December 2007.

[31] M. Karkkainen, J. Holmstrom, K Framling, and K. Artto. Intelligent products - a step towards a more effective project delivery chain. *Computers in Industry*, 50:141–151, 2003.

[32] S Kasputis and H C Ng. Model Composability: Formulating a Research Thrust: Composable Simulations. In *Proceedings of the 2000 Winter Simulation Conference*, pages 1577–1584, 2000.

[33] C Kennedy and G Theodoropoulos. Towards Intelligent Data-Driven Simulation for Policy Decision Support in the Social Sciences. Technical report, University of Birmingham, School of Computer Science, 2005.

[34] Jindae Kim, Kaizhi Tang, Soundar Kumara, Shang-Tae Yee, and Jeffrey Tew. Value analysis of location-enabled radio-frequency identification information on delivery chain performance. *International Journal of Production Economics*, 112(1):403–415, Mar 2008.

[35] Moon-Chan Kim, Chang Ouk Kim, Seong Rok Hong, and Ick-Hyun Kwon. Forward-backward analysis of rfid-enabled supply chain using fuzzy cognitive map and genetic algorithm. *Expert Systems with Applications*, 35(3):1166 – 1176, 2008.

[36] Robert Kittell and Ann Dunkin. Applying the simulation process: simulation study of HDA parts degreaser. In *WSC { '}91: Proceedings of the 23rd conference on Winter simulation*, pages 816–821, Washington, DC, USA, 1991. IEEE Computer Society.

[37] Khalid Kouiss, Henri Pierreval, and Nasser Mebarki. Using multi-agent architecture in fms for dynamic scheduling. *Journal of Intelligent Manufacturing*, 8:41–47, 1997. 10.1023/A:1018540317470.

[38] L.A. Lefebvre, É. Lefebvre, Y. Bendavid, S.F. Wamba, and H. Boeck. The potential of RFID in warehousing activities in a retail industry supply chain. *Journal on Chain and Network Science*, 5(2):101–110, 2005.

[39] J. J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.

[40] G T Mackulak, F P Lawrence, and T Colvin. Effective Simulation Model Reuse: A case study for AMHS modeling. In *Proceedings of the 1998 Winter Simulation Conference*, pages 979–984, 1998.

[41] Anu Maria. Introduction to modeling and simulation. In *Proceedings of the 1997 Winter Simulation Conference*, pages 7–13, Washington, DC, USA, 1997. IEEE Computer Society.

[42] Charles Mclean and Swee Leong. The role of simulation in strategic manufacturing. In *In Proceedings of the 2001 International working Conference on Strategic Manufacturing*, 2001.

[43] Saurabh Mittal, Eddie Mak, and James J. Nutaro. DEVS-Based Dynamic Model Reconfiguration and Simulation Control in the Enhanced DoDAF Design Process. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 3(4):239–267, 2006.

[44] László Monostori and Balázs Csanád Csáji. Complex adaptive systems (cas) approach to production systems and organisations. In Mamoru Mitsuishi, Kanji Ueda, and Fumihiko Kimura, editors, *Manufacturing Systems and Technologies for the New Frontier*, pages 19–24. Springer London, 2008.

[45] Antonio Paulo Moreira, Armando Sousa, and Paulo Costa. Vision Based Real-Time Localization of Multiple Mobile Robots. In *3rd Int. Conf. on Field and Service Robotics*, pages 103–106, jun 2001.

[46] Michael F Morris. Simulation as a Process. *SIGSIM Simul. Dig.*, 4(1), 1972.

[47] K. Muller. Advanced systems simulation capabilities in SimPy. *Europython 2004*, 2004.

[48] K. Muller and T. Vignaux. SimPy: Simulating Systems in Python. *ONLamp.com Python Devcenter*, 2003.

[49] E H Page and J M Opper. Observation on the Complexity of the Composable Simulation. In *Proceedings of the 1999 Winter Simulation Conference*, pages 553–560, 1999.

[50] K.-J. Park and Y.-H. Lee. A On-line Simulation Approach to Search Efficient Values of Decision Variables in Stochastic Systems. *Int J Adv Manuf Technol*, 2005.

[51] Robin G. Qiu. RFID-enabled automation in support of factory integration. *Robot. Comput.-Integr. Manuf.*, 23(6):677–683, 2007.

[52] Cesar A. Quiroga. Performance measures and data requirements for congestion management systems. *Transportation Research Part C*, 8:287–306, 2000.

[53] A. Regattieri, M. Gamberi, and R. Manzini. Traceability of food products: General framework and experimental evidence. *Journal of Food Engineering*, 81(2):347 – 356, 2007.

[54] G. Rossum. Python reference manual. CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, 1995.

[55] John Ryan and Cathal Heavey. Process modeling for simulation. *Computers in Industry*, 2006.

[56] Ichiro Satoh. A location model for smart environments. *Pervasive Mob. Comput.*, 3(2):158–179, 2007.

[57] Robert E Shannon. Simulation modeling and methodology. In *WSC {'}76: Proceedings of the 76 Bicentennial conference on Winter simulation*, pages 9–15. Winter Simulation Conference, 1976.

[58] Marcelo Gurgel Simas de Oliveira and Paulo Cezar Martins Ribeiro. Production and analysis of coordination plans using a geographic information system. *Transportation Research Part C*, 9:53–68, 2001.

[59] D P Solomatine. Data-driven modelling: paradigm, methods, experiences. In *Proc. 5th International Conference on Hydroinformatics*, 2002.

[60] Young Jun Son, Albert T Jones, and Richard A Wysk. Automatic Generation of Simulation Models From Neutral Libraries: An Example. In J A Joines, R R Barton, K Kang, and P A Fishwick, editors, *WSC'00: Proceedings of the 32th conference on Winter simulation*, pages 1558–1567, 2000.

[61] Jongchul Song, Carl T. Haas, and Carlos H. Caldas. A proximity-based method for locating RFID tagged objects. *Advanced Engineering Informatics*, 21(4):367–376, 2007.

[62] M J Steele, M M G Rabadi, and G Cates. Generic Simulation Models of Reusable Launch Vehicles. In *Proceedings of the 2002 Winter Simulation Conference*, pages 747–753, 2002.

[63] Frédéric Thiesse and Elgar Fleisch. On the value of location information to lot scheduling in complex manufacturing processes. *International Journal of Production Economics*, 112(2):532–547, 2008.

[64] D. Trentesaux, P. Pesin, and C. Tahon. Distributed artificial intelligence for fms scheduling, control and design support. *Journal of Intelligent Manufacturing*, 11:573–589, 2000. 10.1023/A:1026556507109.

[65] M. Tsakiri, M. Stewart, T. Forward, D. Sandison, and J. Walker. Urban fleet monitoring with GPS and GLONASS. *The Journal of Navigation*, 51(03):382–393, 1998.

[66] Tao Wan, Karine Zeitouni, and Xiaofeng Meng. An OLAP system for network-constrained moving objects. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 13–18, New York, NY, USA, 2007. ACM Press.

[67] F. Wilcoxon and A. C. Co. Individual Comparisons by Ranking Methods. *Breakthroughs in Statistics*, 1997.

[68] T R Willemain. Model Formulation: What Experts Think About and When. *Operations Research*, 43(6):916–932, 1995.

[69] Zhengdao Xu and Arno Jacobsen. Adaptive location constraint processing. In *SIGMOD {'}07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 581–592, New York, NY, USA, 2007. ACM.

[70] Minghui Yang. Using data driven simulation to build inventory model. *Proceedings of the 2008 Winter Simulation Conference*, pages 2595–2599, 2008.

[71] M. Zanin and A. Véjar. The emergence of transportation networks. In *European Conference on Complex Systems, ECCS09. Satellite Meeting EmergeNET3: Emergence and Networks European Conference on Complex Systems, ECCS09. Satellite Meeting EmergeNET3: Emergence and Networks*, Warwick UK, 09 2009.