Automatic generation of simulation code using product location information

Andrés VÉJAR Patrick CHARPENTIER

{andres.vejar,patrick.charpentier}@cran.uhp-nancy.fr

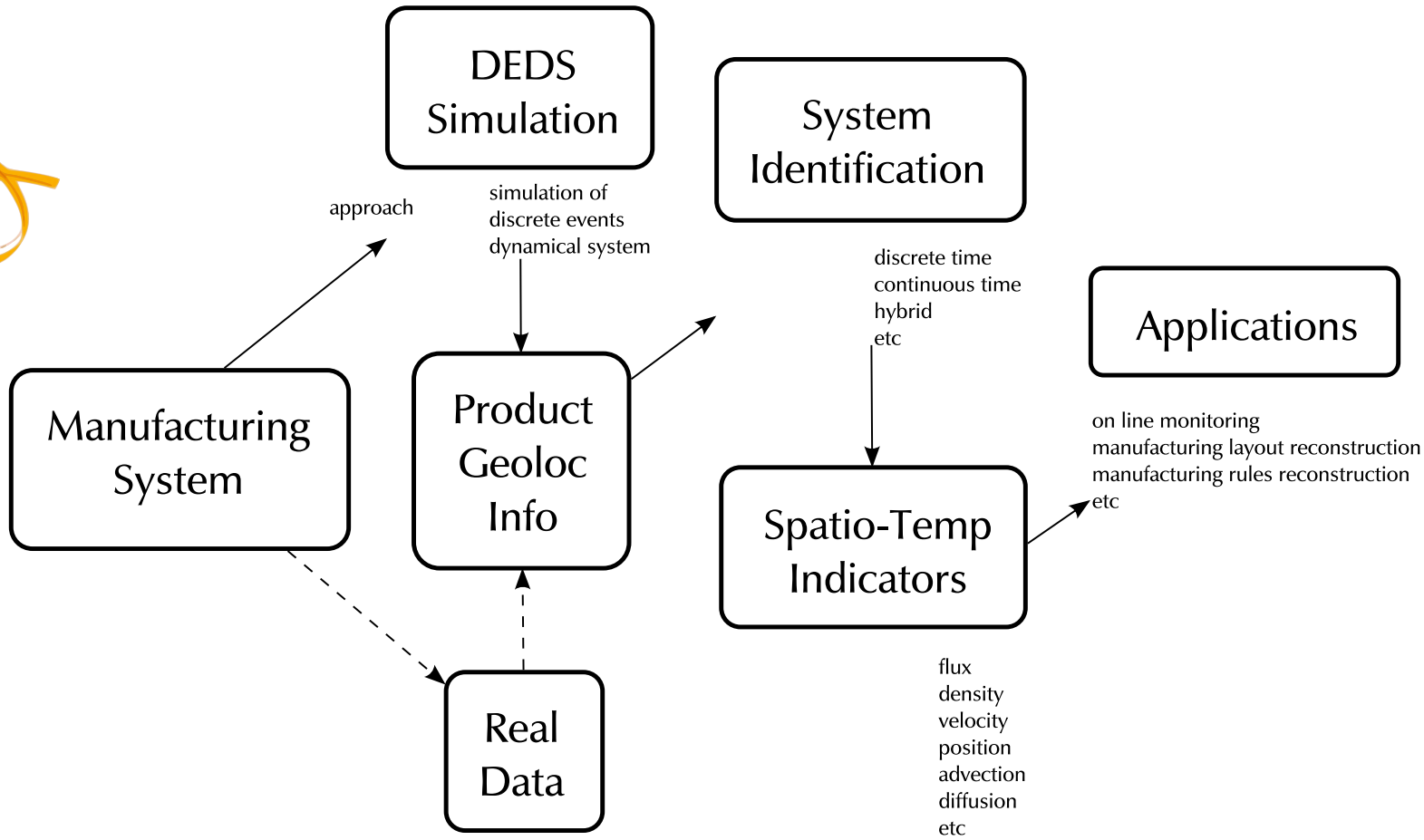
Research Centre for Automatic Control (CRAN)

CNRS UMR 7039, University of Nancy

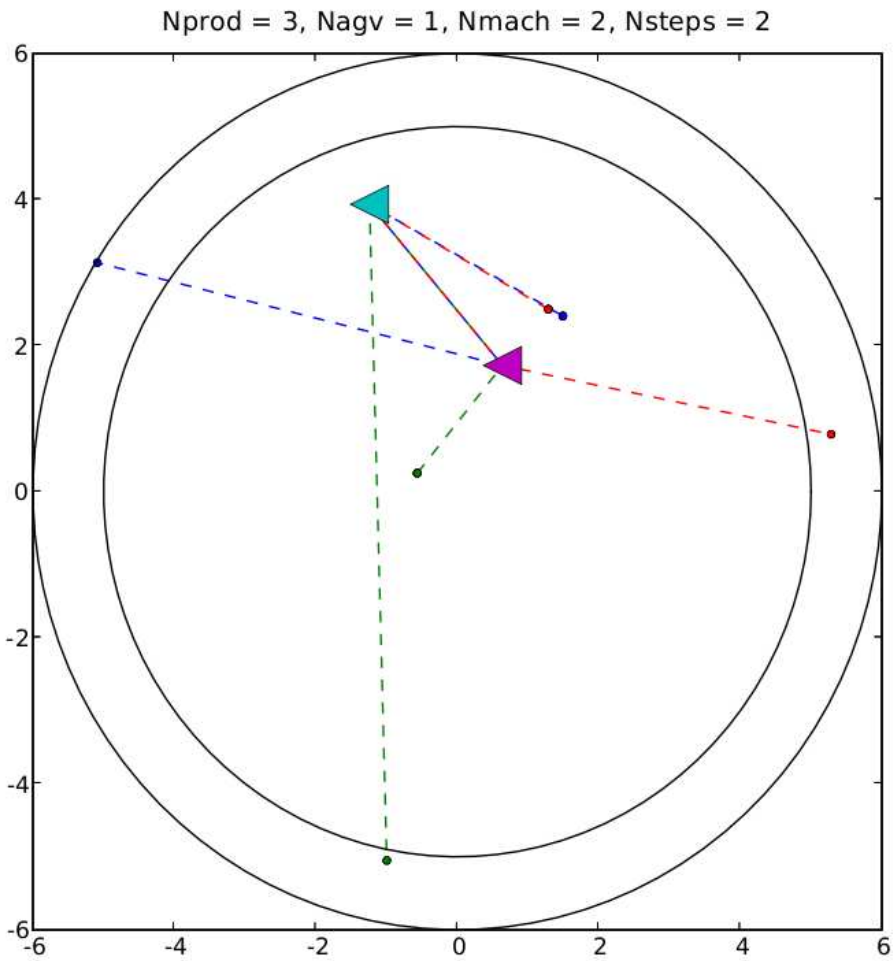
Nancy-Université

13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09). June 3-5, 2009. Moscow, Russia.

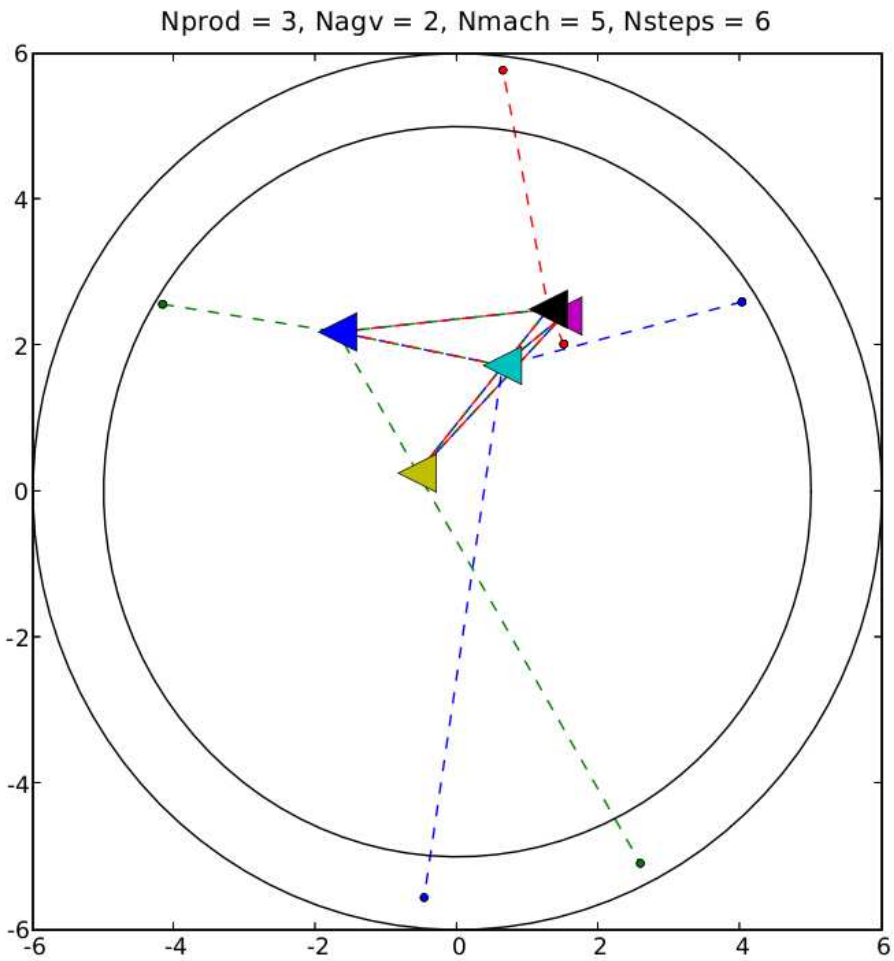
Work Diagram



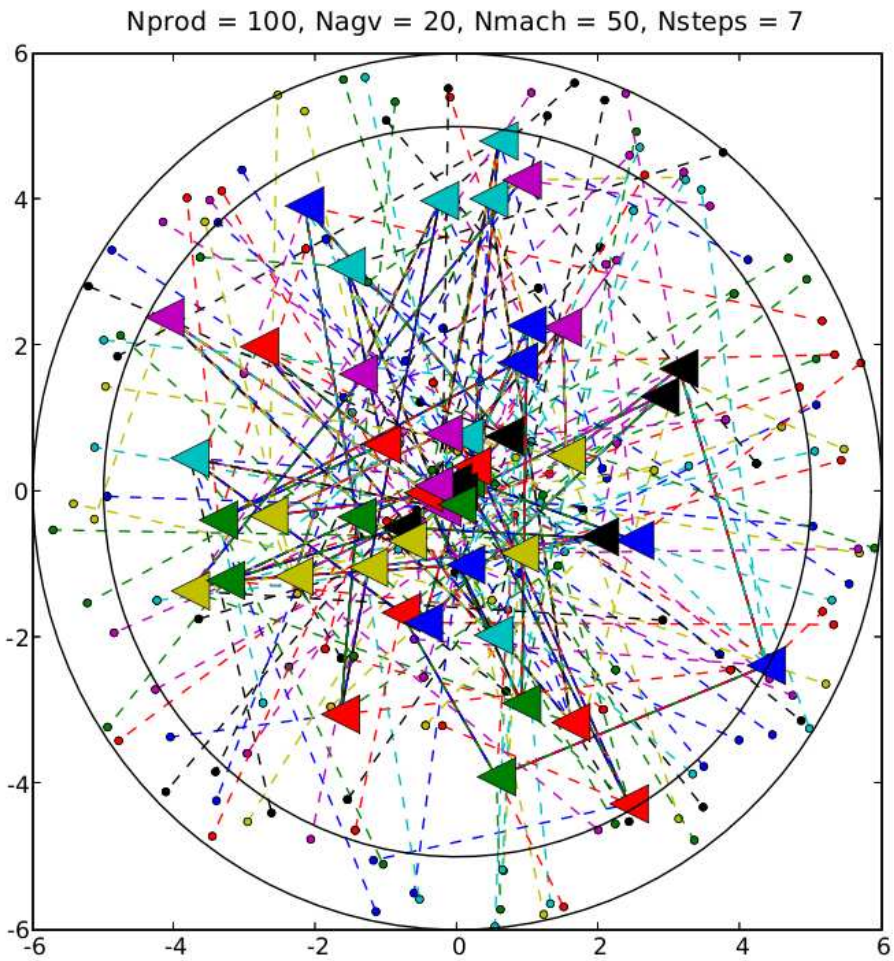
Simulation Example 1



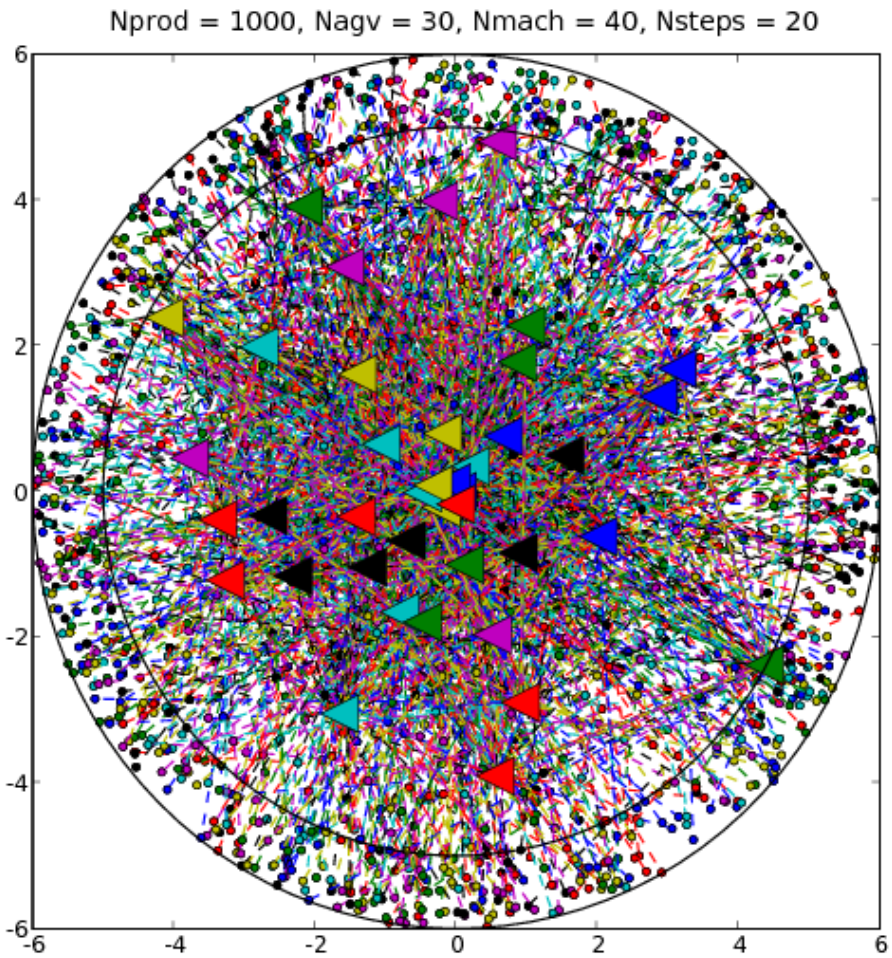
Simulation Example 2



Simulation Example 3



Simulation Example 4



Modeling Approaches
Transport

- Population Based :

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \vec{J} = 0$$

- Individual Based :

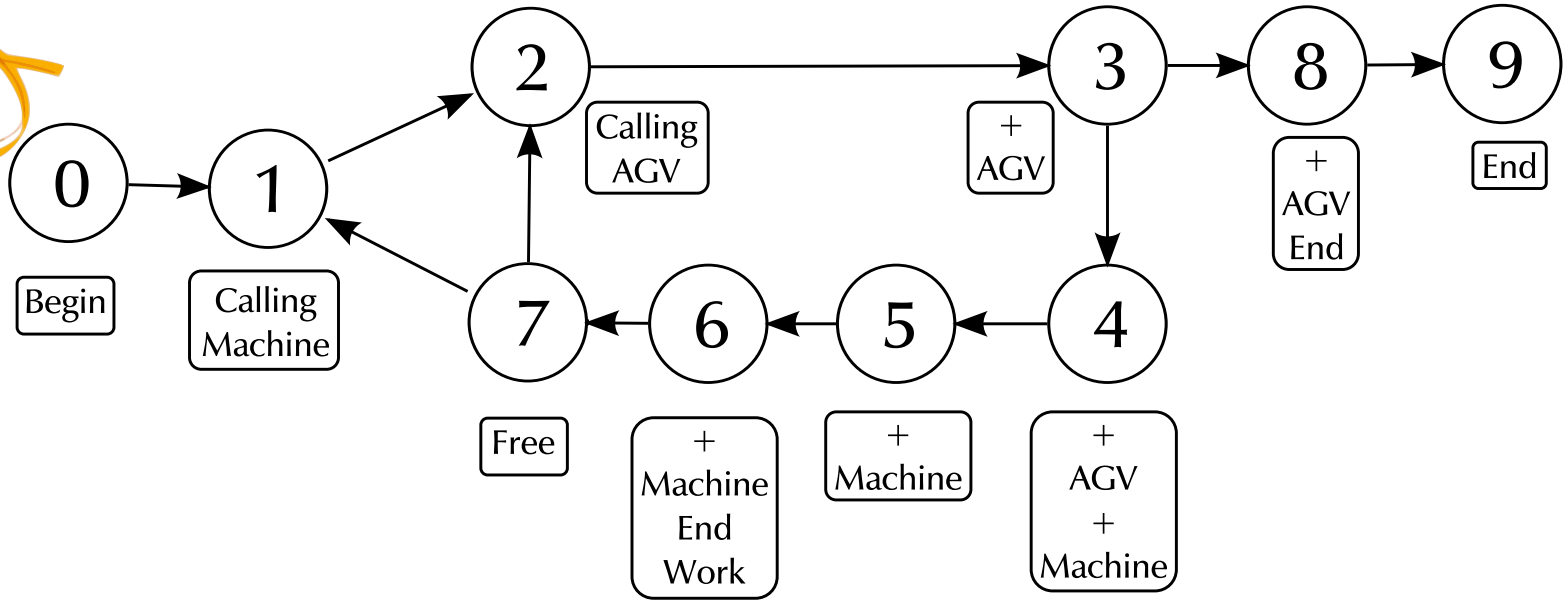
$$\frac{d}{dt}\frac{\partial L}{\frac{d}{dt}\mathbf{r}_i} - \frac{\partial L}{\mathbf{r}_i} = 0$$

- Hybrid :

Particles in a potential field

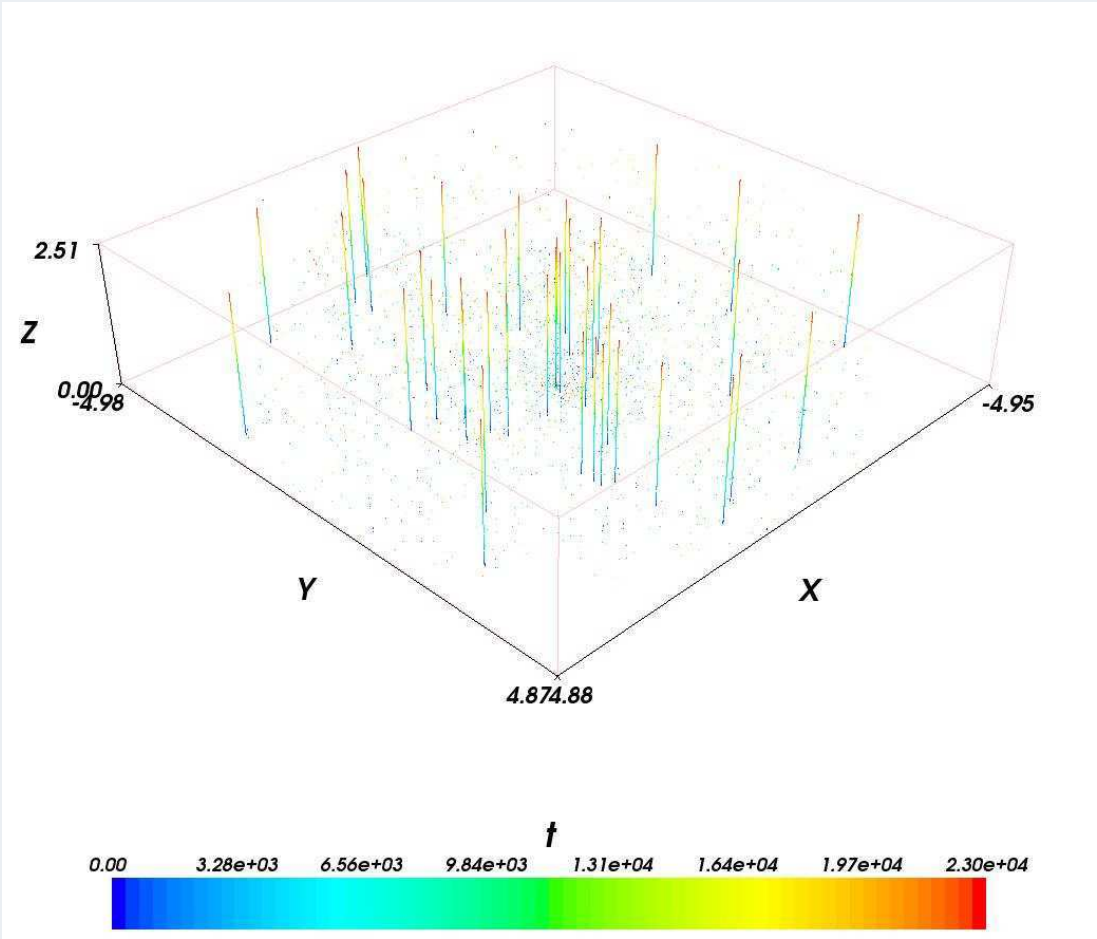
Product Life in the Manufacturing System

FSM



Production plant

The 40 machines, z – *axis* is the time



Spatio-temporal Indicator

Density ρ

$$\rho_{\epsilon,\tau}(\mathbf{x}_s, t_s) = \frac{1}{\pi\tau\epsilon^2} \sum_{\{(\mathbf{x},t)\in\mathcal{S}|t\in\mathcal{T}\}} \theta(\epsilon - \|\mathbf{x} - \mathbf{x}_s\|)\theta(\tau - |t - t_s|)$$

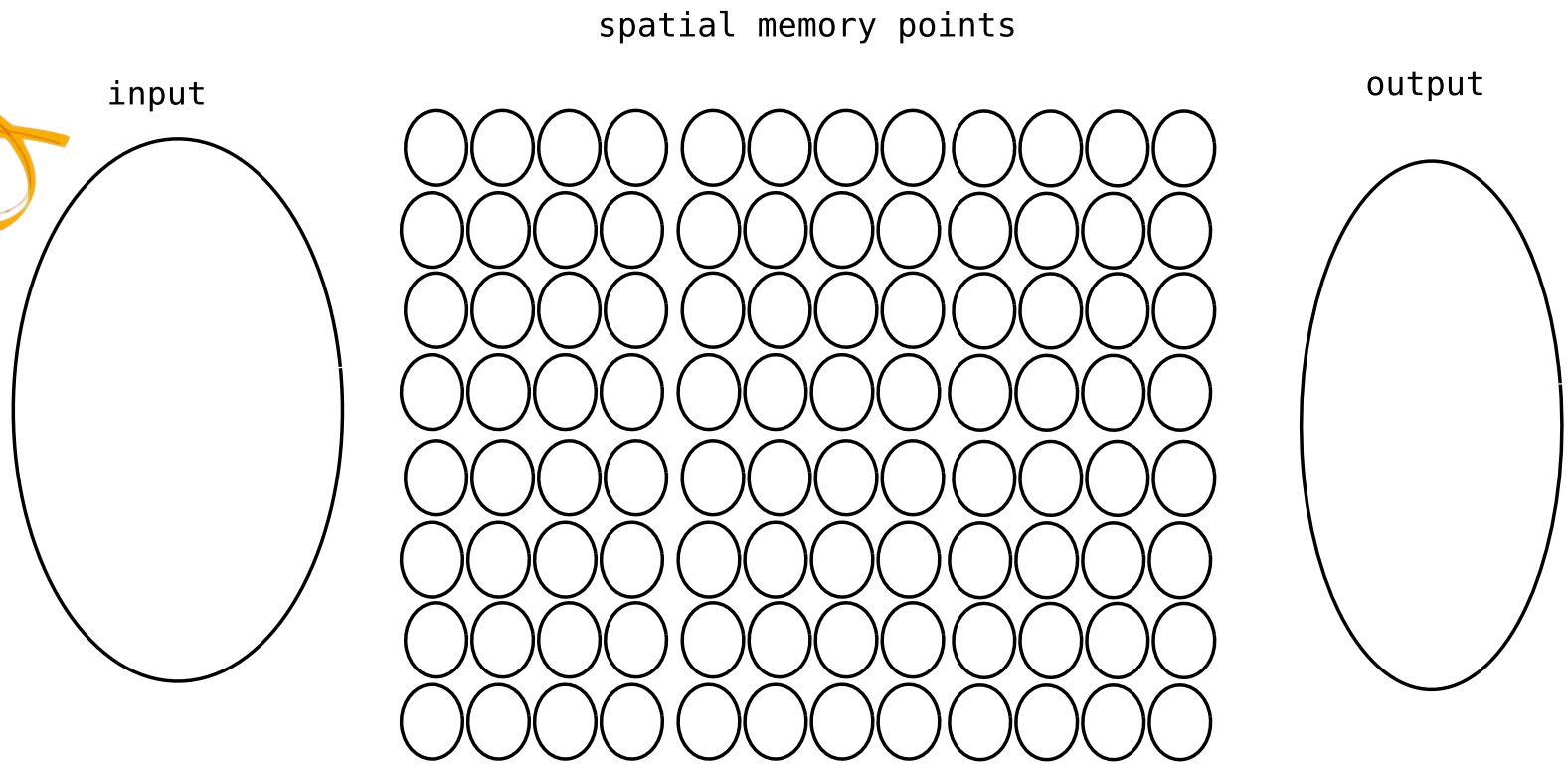
$$\mathbf{x}_s \in k\mathbb{Z}^2 \quad t_s \in \tau_d\mathbb{Z} \quad \mathcal{T} \in t_s + \tau/2[-1,1]$$

$$\mathcal{S} = \{(\mathbf{x}_k, t_k)\}_{k=1}^{N}$$

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

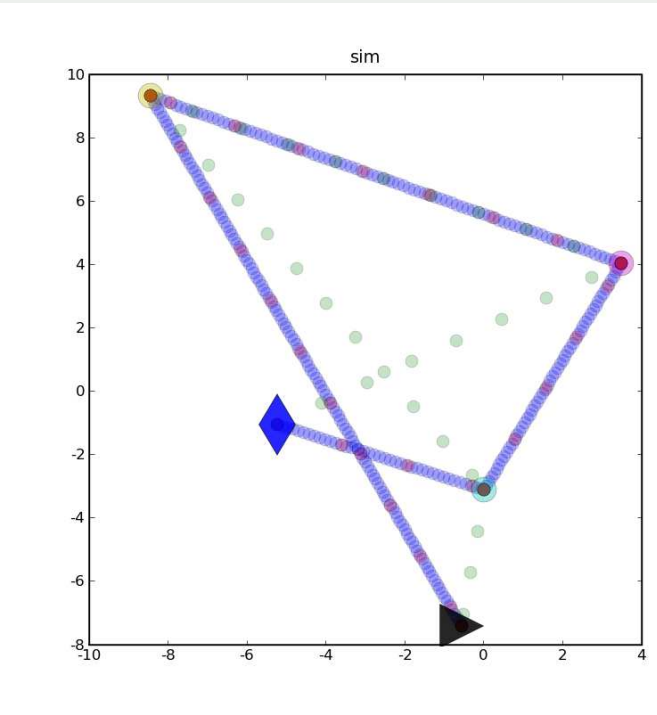
. . . Hybrid : Particles in a potential field

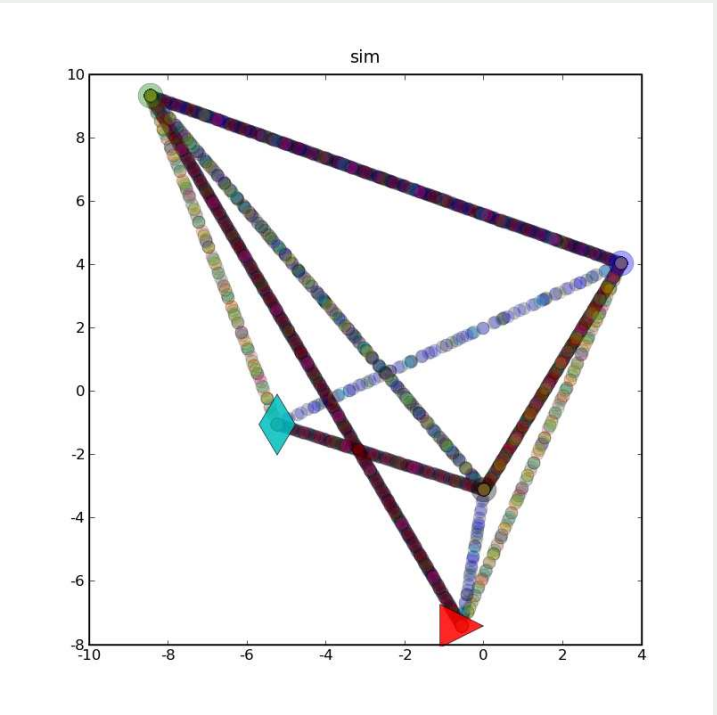
Transport : Modeling approaches



Simulation
3 types, 3 machines

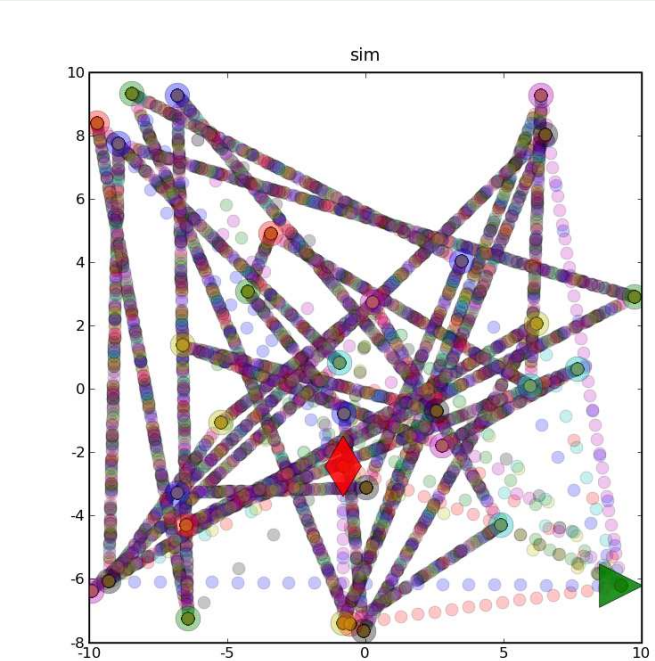
3 pructs - 20 products

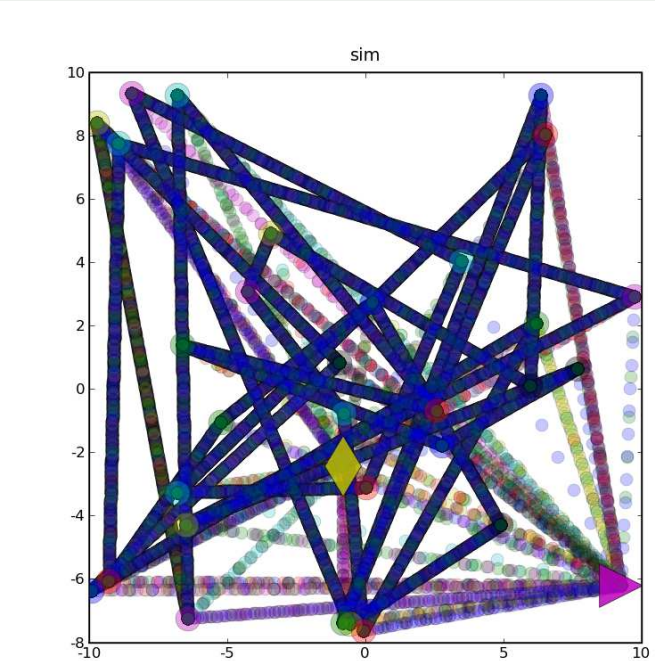




Simulation
30 types, 30 machines

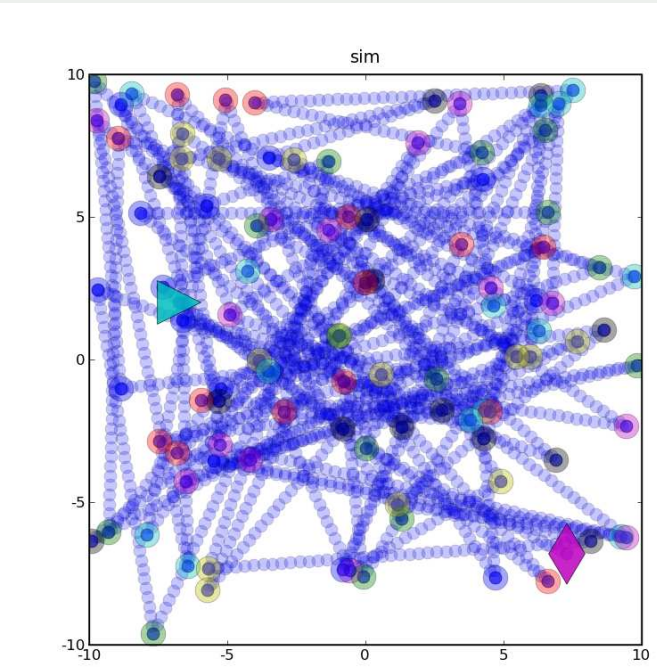
20 products - 100 products

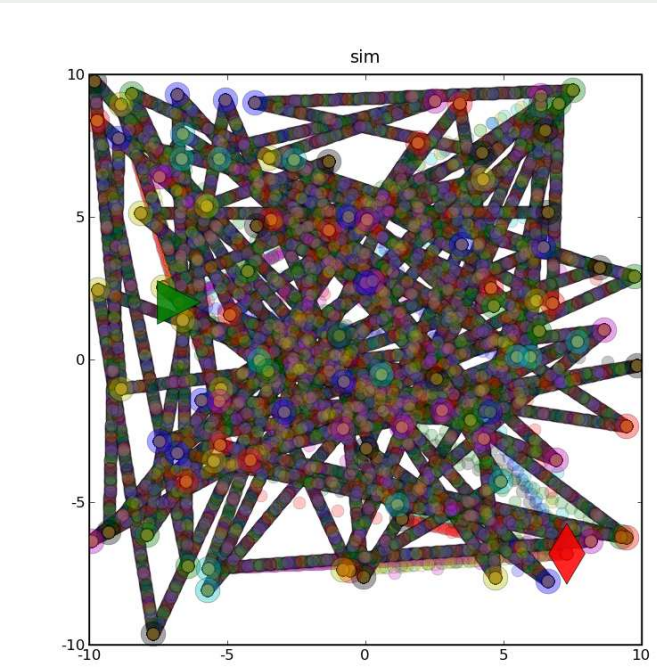




Simulation
100 types, 100 machines

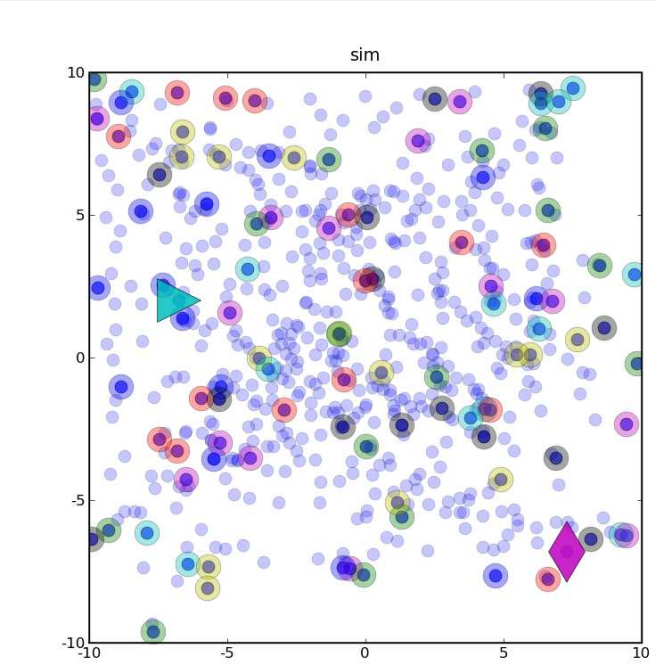
1 product - 100 products

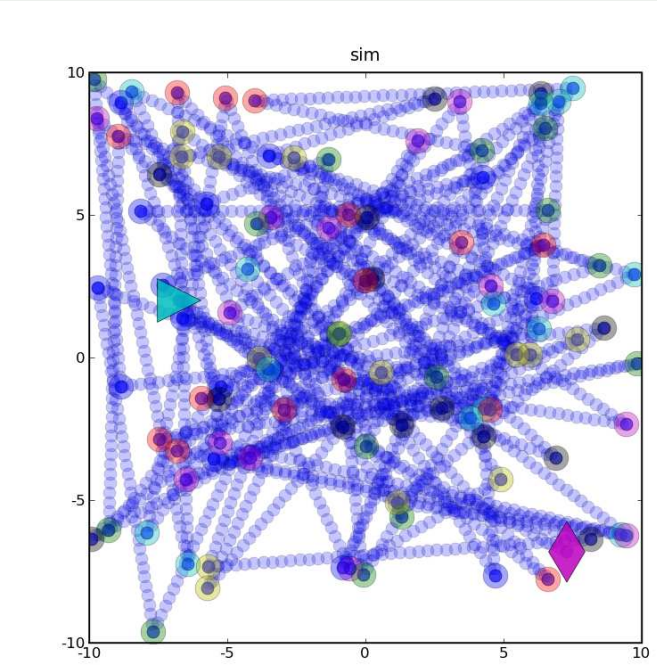




Simulation
100 types, 100 machines

1 product, low loc. frequency - high loc. frequency



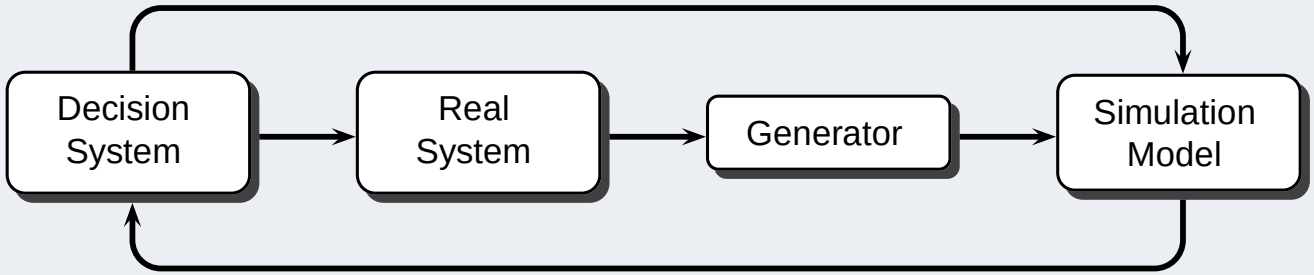


The Problem

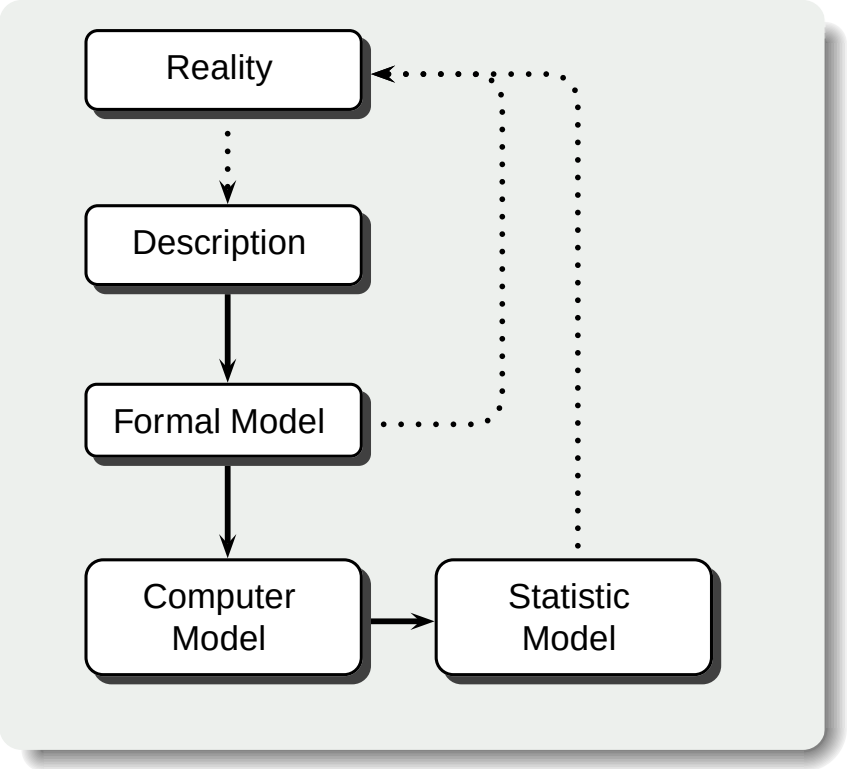
Simulation of manufacturing flux

- the main tool to evaluate the dynamic of manufacturing systems, used on :
  - *Design* (to develop).
  - *Re-engineering* (to improve), and
  - *Control*, (in operation).
- the modeling phase and the maintenance phase are constituted by delicate and time-demanding human operations
- results depend on the human expert skills

Make automatically (in *re-engineering* or in *operation*)



Modeling Process
Development of simulation models



1. System of Interest, Descriptive Model
2. Formal Model (Analytic Solution ?)
3. Computer Model
4. Statistic Model
5. Verification and Validation

(Sánchez 2006)

Some others works :

(Law & Kelton '82, Banks & Carson '85, Balci '90, DoD '96, Maria '97)

Steps of a Simulation Study

Developing of a Simulation Model

- Identify the problem
- Formulate the problem
- Collect and process real system data
- Formulate and develop a model
- Verify and validate the model
- Document the model

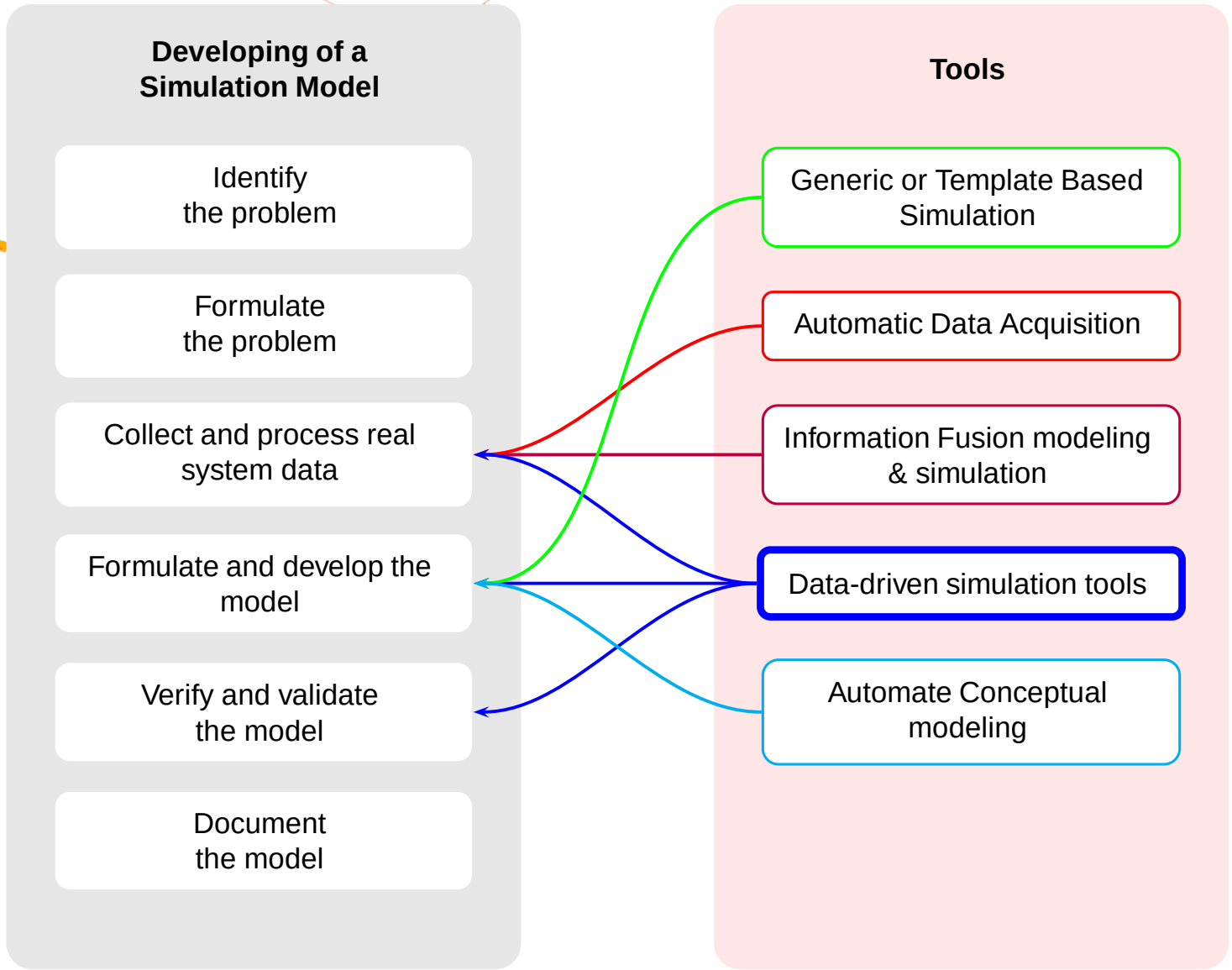
Scenarios

- Select the scenarios
- Establish the simulation plan
- Execute simulations

Simulation Analysis

- Explain and present the simulation output
- Suggest the alternatives
- Implement
- Report

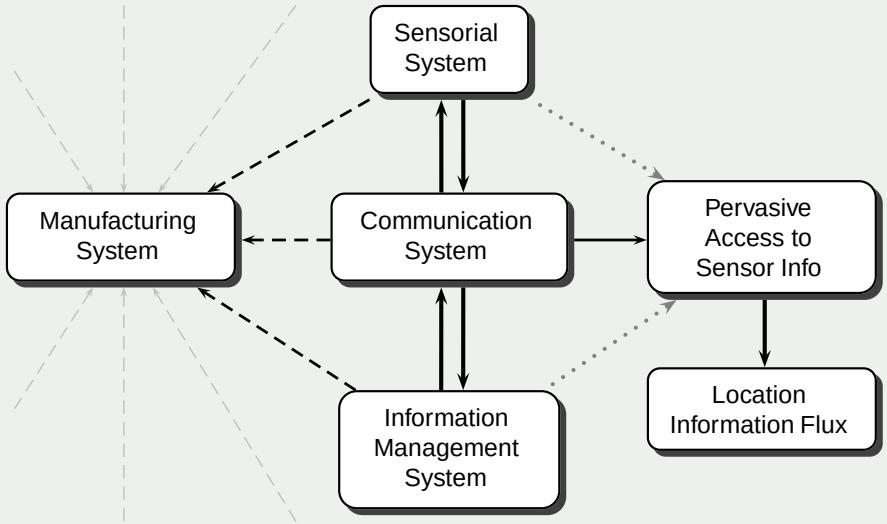
Modeling Process . . .
Support tools



Proposal
and current research

Automatic generation of simulation code using product location information

Show the new inputs and benefits, in the use of a product location information flux, during the manufacturing operation processes.



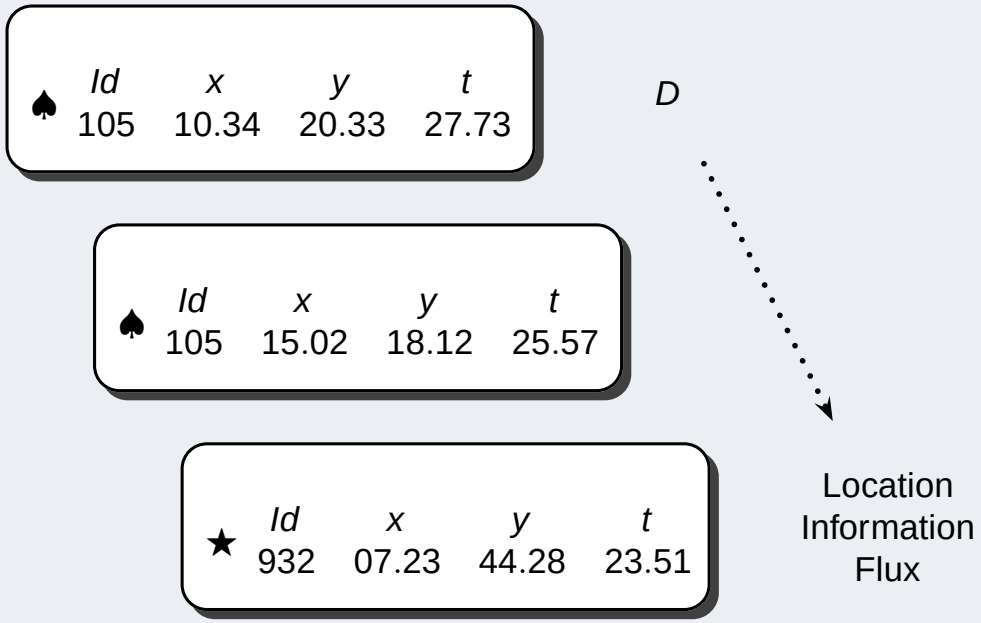
Location Information Flux

- Object Location (positioning) technologies are becoming *available* and *affordable*
- The Spatiotemporal data linked to products obey to useful and necessary rules for the model construction
  (EXAMPLE : one object cannot be in more than one place at the same time, etc.)

- Products ♠, ★, ...
- Object *Id*
- Position (x, y)
- Time t

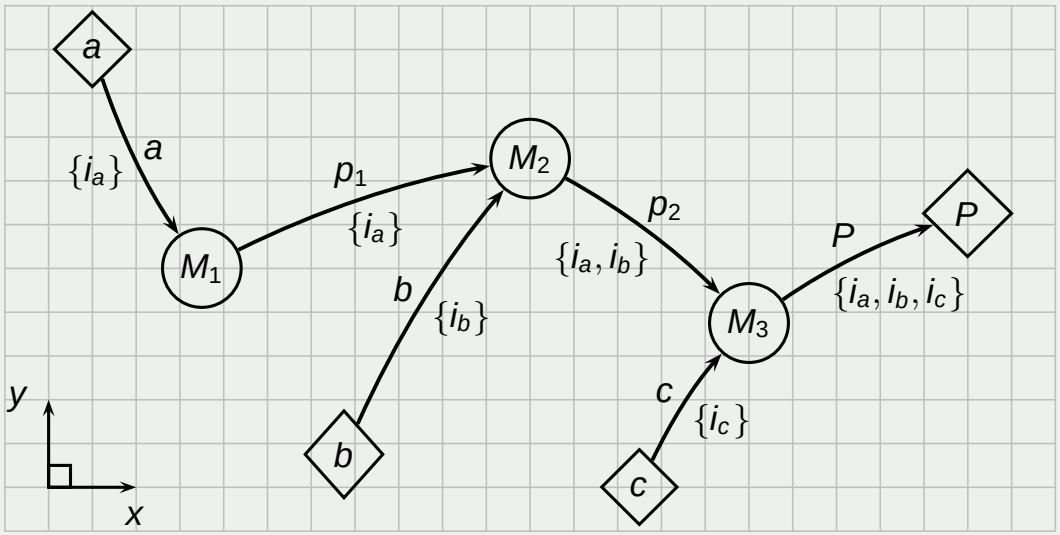
$$d_k = (Id, x, y, t)_k$$

$$D = \{d_0, d_1, d_2, \ldots\}$$



Product Nomenclature

Workshop, three machines and one product

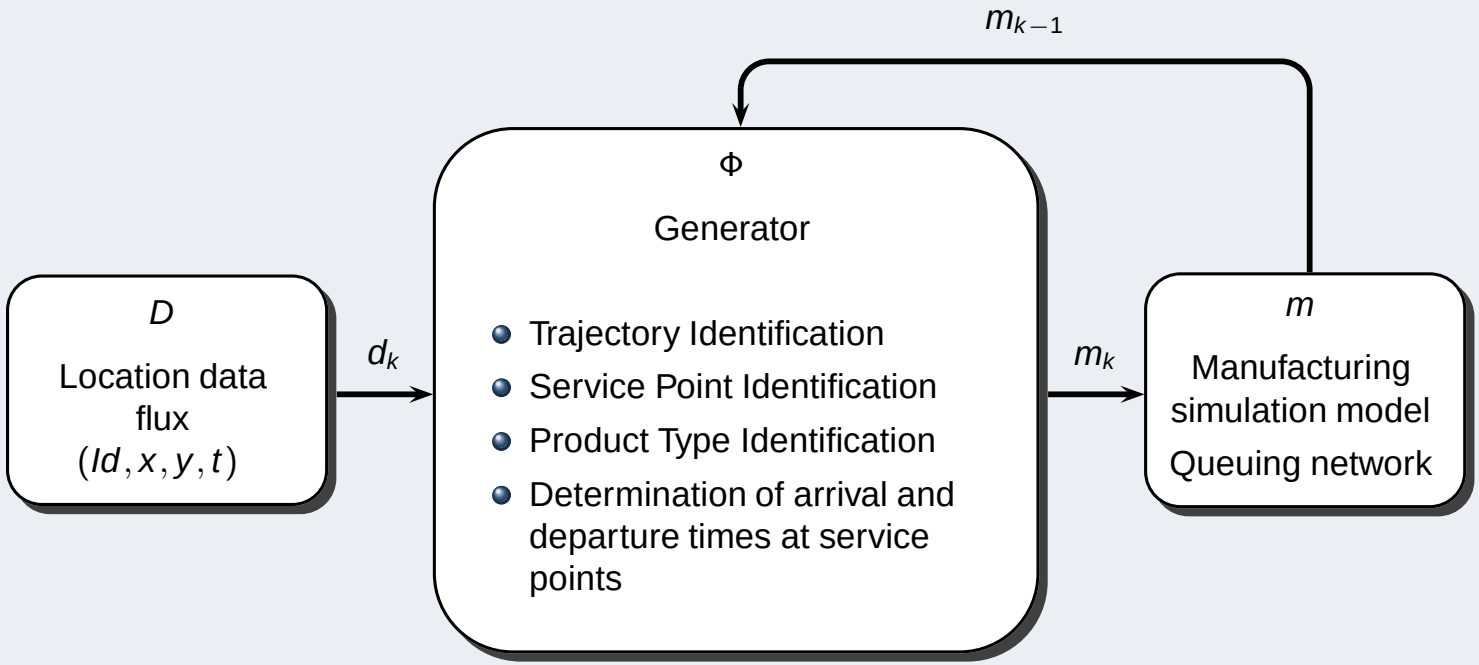


$$
\begin{array}{ccccc}
a & \mapsto & M_1(a) & = & p_1 \\
(p_1, b) & \mapsto & M_2(p_1, b) & = & p_2 \\
(p_2, c) & \mapsto & M_3(p_2, c) & = & P
\end{array}
$$

$$P = M_3\left(M_2\left(M_1(a), b\right), c\right)$$

Generator

Only the product location data (during its passage through the production system) are used.



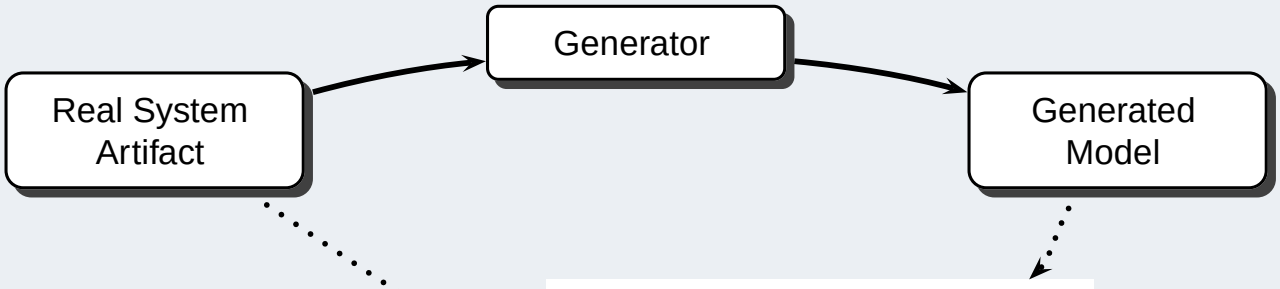
$$m_0 = \emptyset$$
$$m_k = \Phi(d_k, m_{k-1})$$

Algorithm

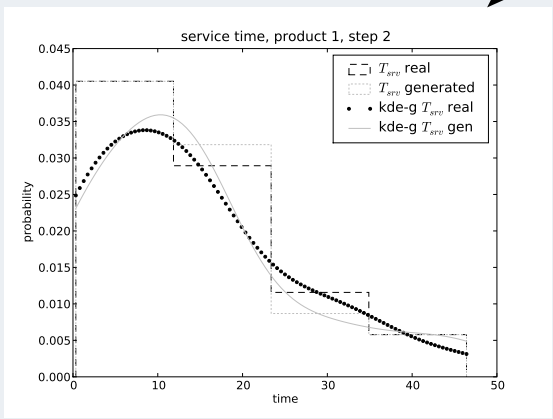
- **Server Identification** : null speed points (with recurrence).
  `S=GetServers(D)`
- **Trajectory Identification** : Arcs between points.
  `A=GetTrajectories(D,S)`
- **Product Identification** : Graph, $p_j = (A_j, S_j)$.
  `P=GetProducts(D,S,A)`
- **Layout Generation** : Graph, union of all product graphs, $L = \cup_j p_j$
  `L=GetLayout(P)`
- **Arrivals Distribution** : for each product in each server.
  `E=GetArrivalSite(L,D)`
- **Inter-arrivals time Distribution** : for each product in each server.
  `I=GetTimeIA(L,D)`
- **Service time Distribution** : for each product in each server.
  `T=GetTimeS(L,D)`
- **Model Generation** :
  `M=Gen(L,E,I,T)`

for d **do**
  if $\exists\ p \in P : p.i = d.i$ **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
        $m \leftarrow m' \in M : m'.r = p.m.r$
        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 $\nexists\ m \in M : m.r = d.r$ **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

Implemented Principle

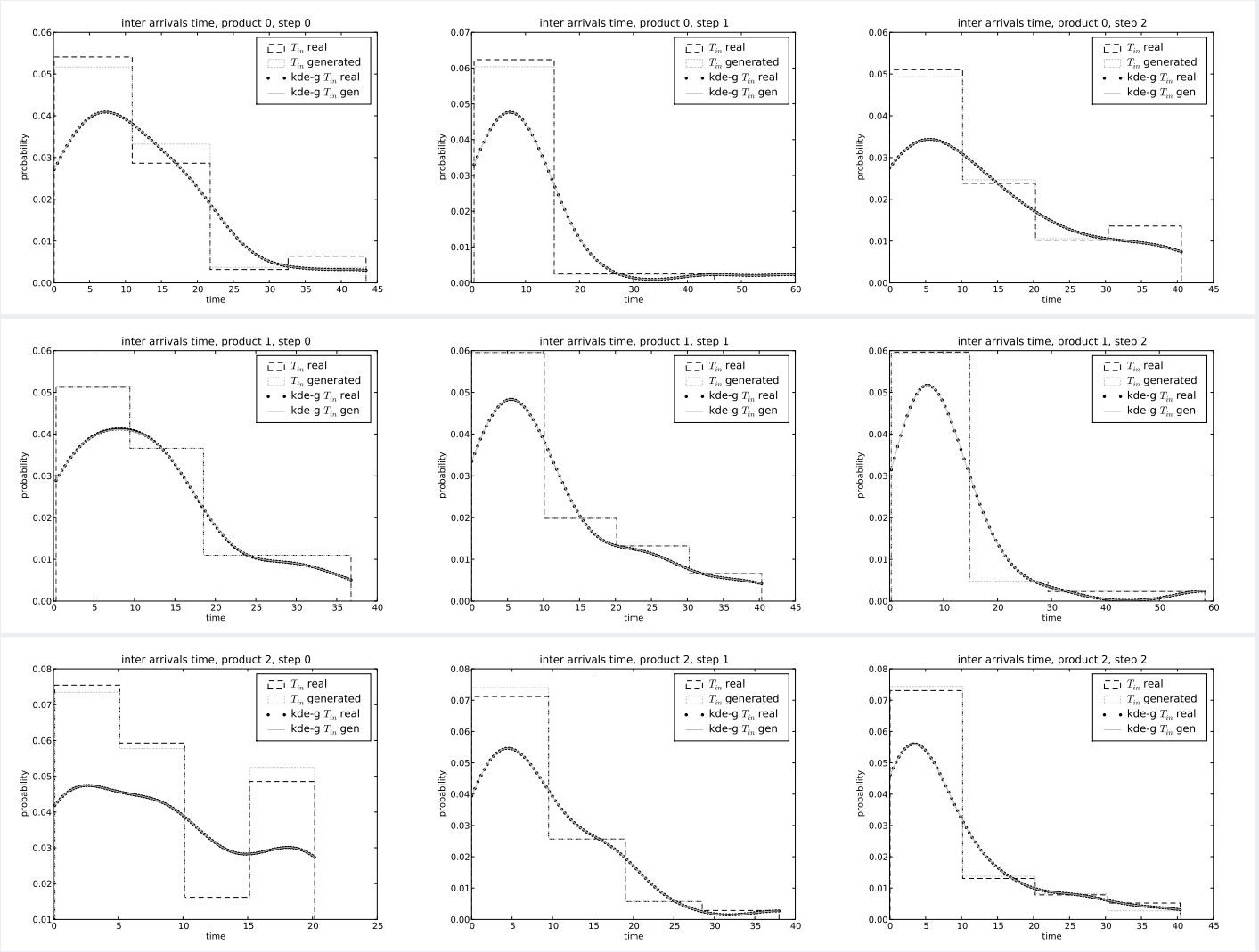


parameter	value
product types	3
machines	3
operation for each product	3
products	100
inter arrival time	exponential, $\mu = 1/3$
service time	exponential, $\mu = 13$
AGV's	10
AGV's velocity	0.44
workshop limits	$(-10, -10), (10, 10)$
location sampling period	0.5

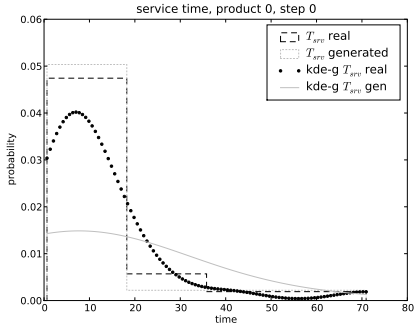


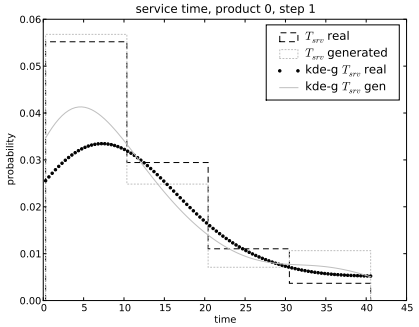
- Simulation module made with SimPy (Simulation in Python) (Muller et Vignaux, 2003)
- Real System Artifact is used to create the product location flux *D*
- *D* is gathered by the generation module
- The results are verified by the data analysis module

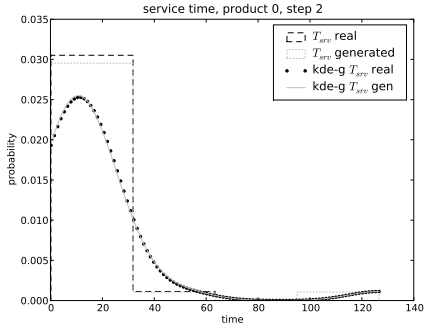
Results
Inter-arrival time distributions

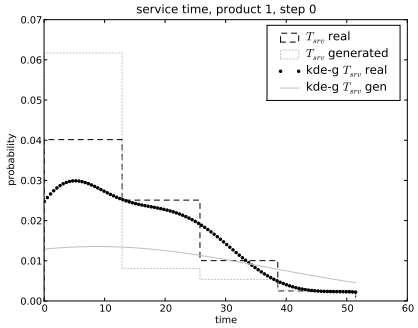


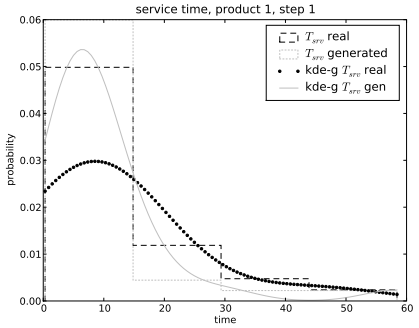
Results
Service time distributions

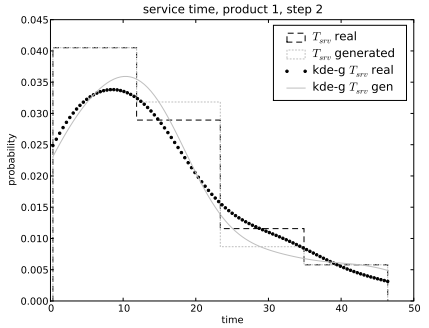


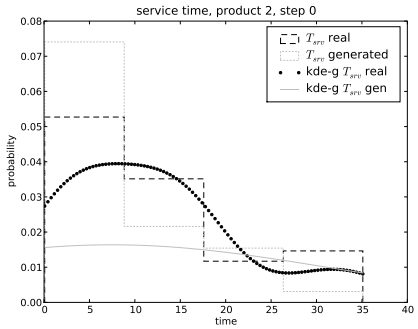


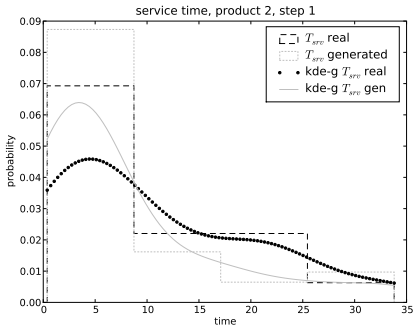


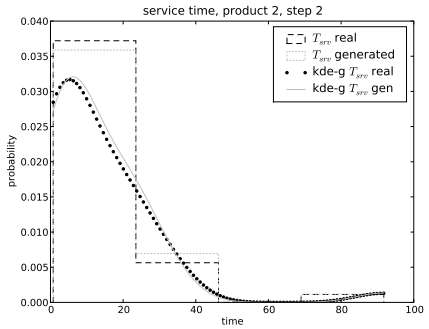






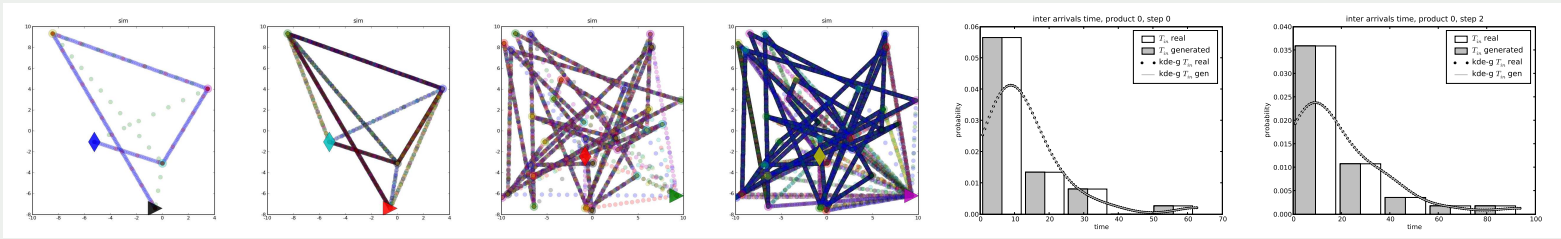






Conclusion

- The output provided by the model from the code generator are quite consistent with those obtained from the *artifact* of the real system
- The results validate the implementation approach as well as the tool that we have developed
- The feasibility of the proposed method was confirmed by tests on larger systems (up to 15 machines, 15 operations, 15 product types)



Further Research

- Take into account imprecise location measures
- Test and technological feasibility (GPS, accelerometers, . . .)
- Interaction between *Technologies* and *Applications*
- Validation with real-size manufacturing systems
- Extend Application Ranges (other than manufacturing)