

# Générateur de code de simulation à partir de l'information de localisation du produit

Andrés VÉJAR Patrick CHARPENTIER

prénom.nom@cran.uhp-nancy.fr

Centre de Recherche en Automatique de Nancy (CRAN), CNRS UMR 7039  
Groupe Thématique SYMPA, Équipe-Projet SP



Nancy-Université

3<sup>èmes</sup> Journées Doctorales / Journées Nationales MACS – JD-JN-MACS'09  
17-18 mars 2009, Angers, France

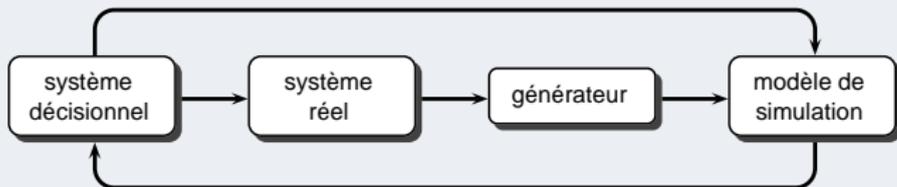
JD-JN-MACS 2009

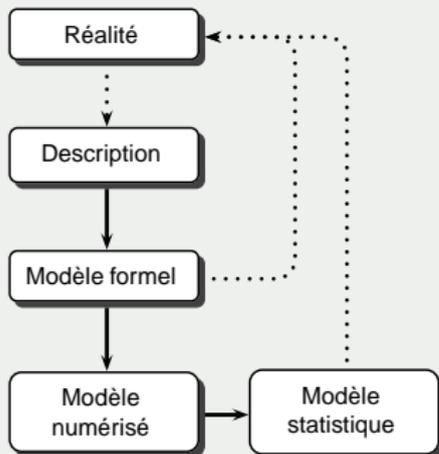


## La simulation de flux

- incontournable pour l'évaluation de la dynamique des systèmes manufacturiers, utilisée dans les phases de :
  - *Design* (pour Concevoir),
  - *Re-engineering* (pour Améliorer), et
  - *Control*, (en Exploitation).
- les phases de modélisation, puis de maintenance restent des opérations délicates et chronophages
- les résultats sont dépendants de l'habileté des experts humains

## Automatiser ces phases (en *re-engineering* ou en *exploitation*)





- 1 Système d'intérêt, modèle descriptif
- 2 Modèle formel (solution analytique ?)
- 3 Modèle numérisé
- 4 Modèle statistique
- 5 Vérification et validation

(Sánchez 2006)

## D'autres travaux :

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

# Etapes d'une étude de simulation

## Développer un modèle de simulation

Identifier le problème

Formuler le problème

Collecter et traiter les données du système réel

Formuler et développer le modèle

Vérifier et valider le modèle

Documenter le modèle

## Scenarii

Choisir les scenarii

Établir le plan pour les exécutions

Réaliser les exécutions de la simulation

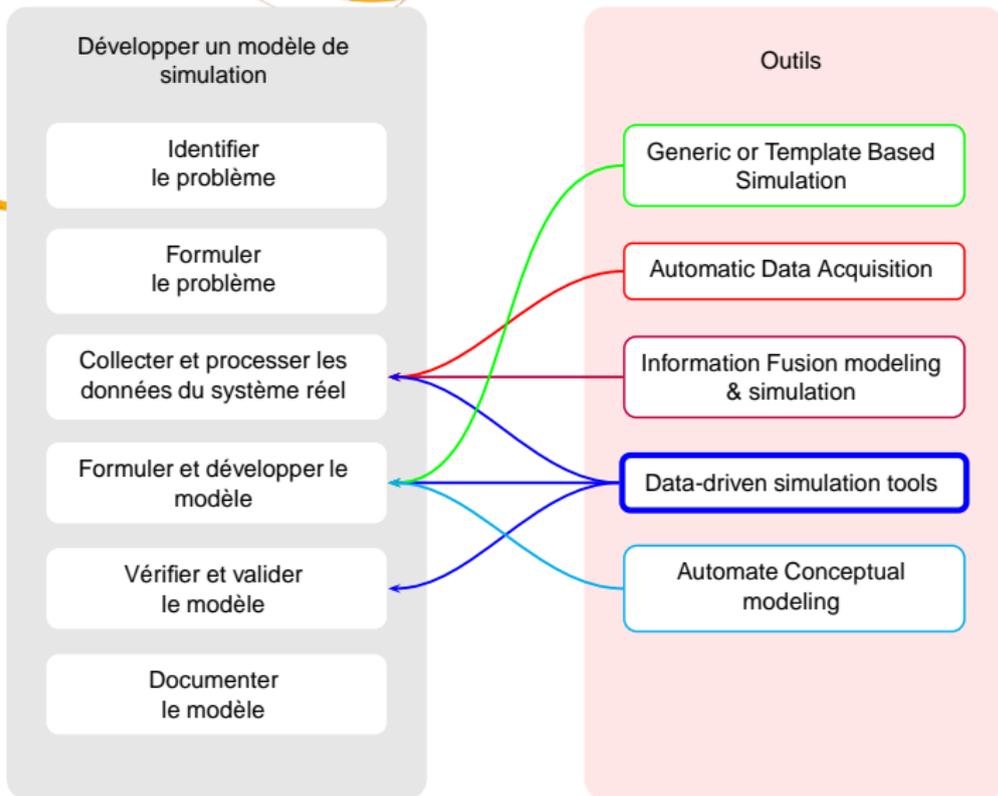
## Réaliser une analyse de la simulation

Interpréter et présenter les résultats

Suggérer les alternatives possibles d'action

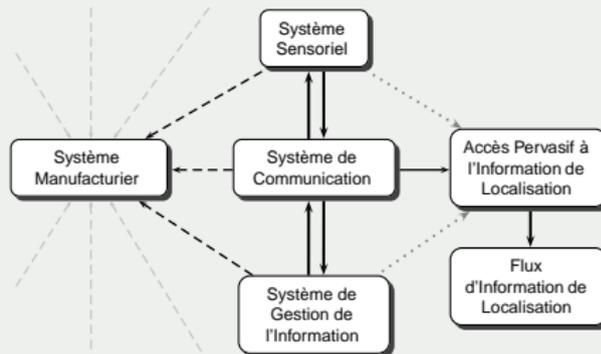
Implémentation

Rapport



## Générateur de code de simulation à partir de l'information de localisation du produit

Montrer les apports, sur le contrôle et les performances d'un système manufacturier, d'un flux d'information de localisation de produits durant leur élaboration.



- La technologie pour localiser des objets est devenue disponible et abordable
- Les données spatio-temporelles liées aux produits obéissent à des règles utiles et nécessaires lors de la construction de modèles  
(EXEMPLE : un objet ne peut être qu'à une et une seule position à un même instant, etc.)

- Id Objet
- Position ( $x, y$ )
- Temps

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

$$F = \{f_0, f_1, f_2, \dots\}$$

♠	<i>Id</i>	<i>x</i>	<i>y</i>	<i>t</i>
	105	10.34	20.33	27.73

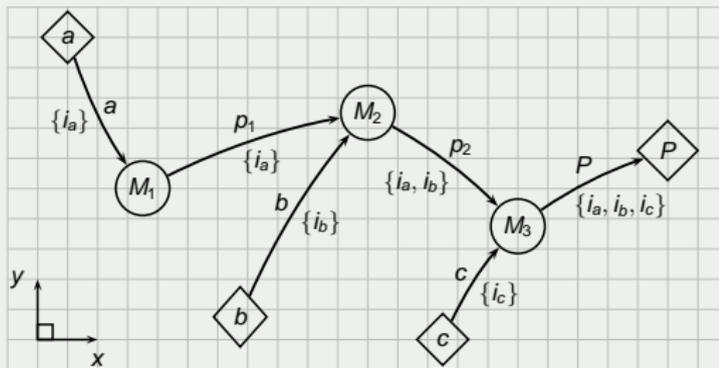
♠	<i>Id</i>	<i>x</i>	<i>y</i>	<i>t</i>
	105	15.02	18.12	25.57

★	<i>Id</i>	<i>x</i>	<i>y</i>	<i>t</i>
	932	07.23	44.28	23.51

F

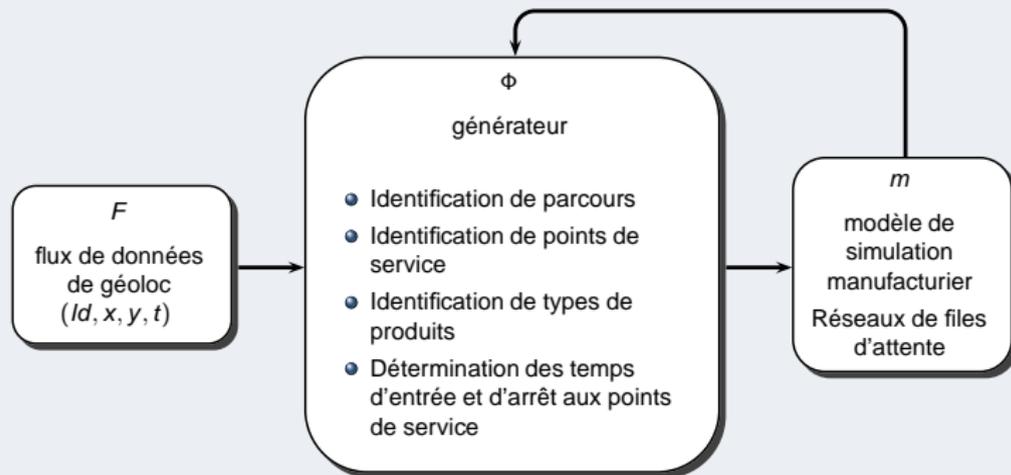
Flux d'information  
de localisation

## Atelier, trois machines et un produit



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

Les seules données utilisées sont les données de localisation des produits durant leur passage en production



$$m_0 = \emptyset$$

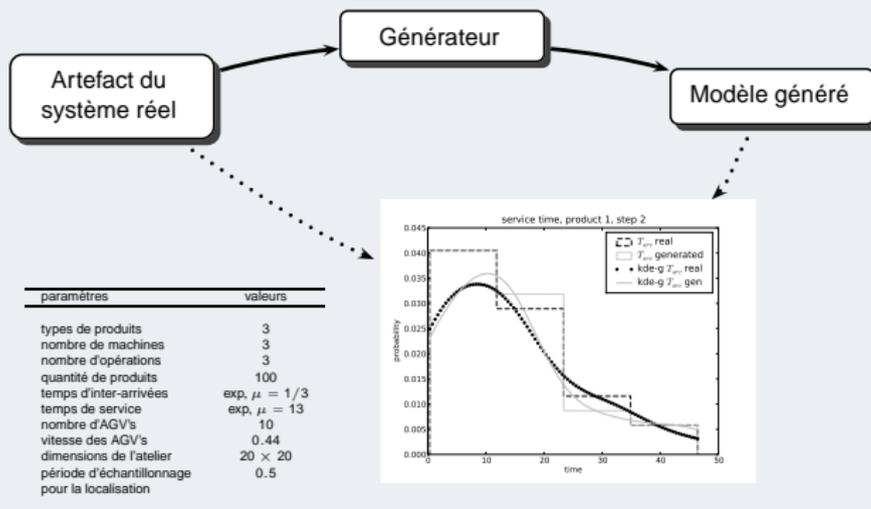
$$m_k = \Phi(f_k, m_{k-1})$$

- **Identifier les serveurs** : Points de vitesse nulle (avec récurrence).  
S=Serveurs(F)
- **Identifier les trajectoires** : Arcs entre points.  
A=Trajectoires(F,S)
- **Identifier produits** : Graphe,  $p_j = (A_j, S_j)$ .  
P=Produits(F,S,A)
- **Générer layout** : Graphe, union de l'ensemble des graphes produits,  $L = \cup_j p_j$   
L=Layout(P)
- **Distribution des entrées** : pour chaque produit sur chaque serveur.  
E=Entrees(L,F)
- **Distribution des temps d'inter-arrivées** : pour chaque produit sur chaque serveur.  
I=TempsIA(L,F)
- **Distribution des temps de service** : pour chaque produit sur chaque serveur.  
T=TempsS(L,F)
- **Générer modèle**.  
M=Gen(L,E,I,T)

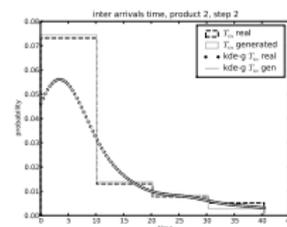
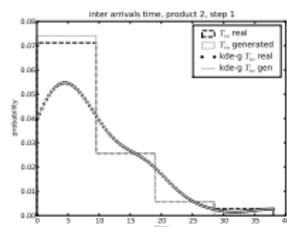
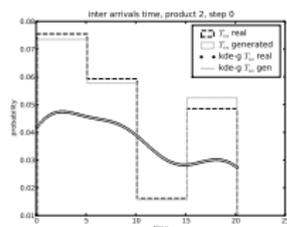
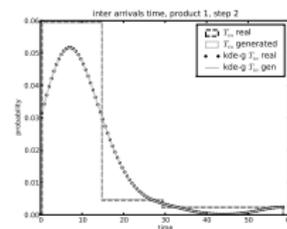
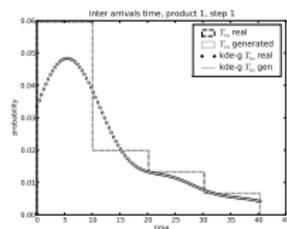
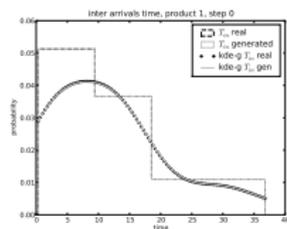
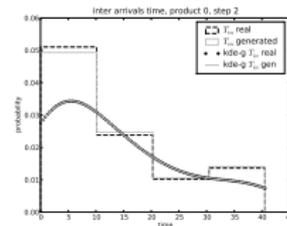
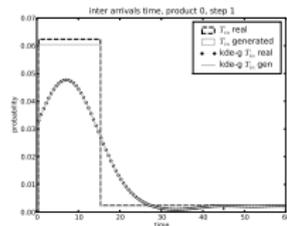
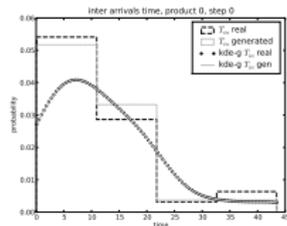
```

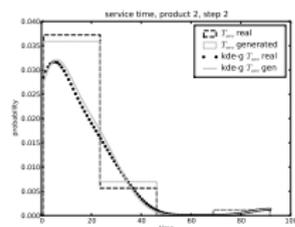
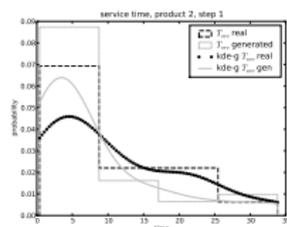
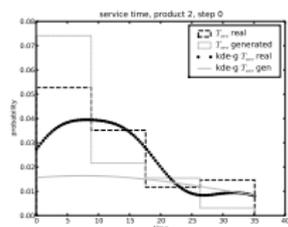
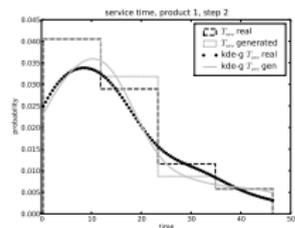
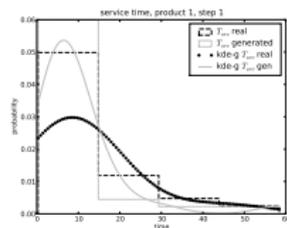
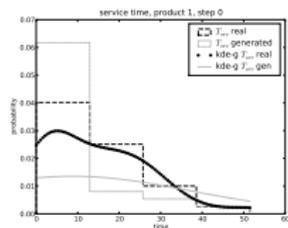
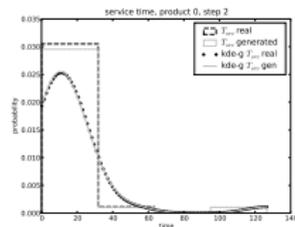
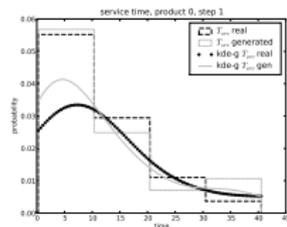
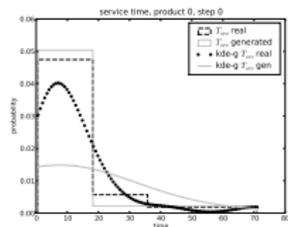
for d do
  if  $\exists p \in P: p.i = d.i$  then
    if  $p.m.s = 1$  then
      if  $p.m.r = d.r$  then
         $p.m.a \leftarrow p.m.a + d.t - p.m.t$ 
         $p.m.t \leftarrow d.t$ 
      else
         $m \leftarrow m' \in M: m'.r = p.m.r$ 
        if  $p.m.t - p.m.a < m.l$  then
           $p.m.T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

```

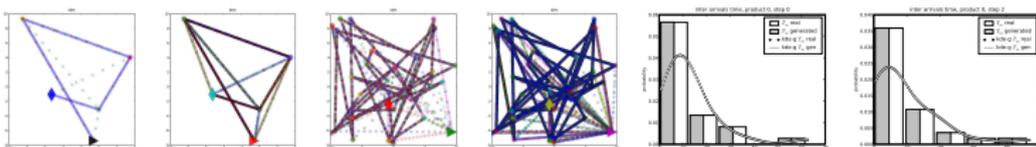


- Module de simulation avec SimPy (Simulation in Python) (Muller et Vignaux, 2003)
- Artefact du système réel pour générer le flux de localisation des produits  $F$
- $F$  est récupéré par le module générateur de modèle de simulation
- Un module d'analyse des données permet de vérifier les résultats obtenus





- Les résultats obtenus valident la démarche mise en œuvre, ainsi que l'outil développé
- Confirmés par des tests effectués sur des systèmes de taille plus importante (jusqu'à 15 machines, 15 opérations ou phases, 15 types de produits)



- Prendre en compte les imprécisions de mesure de localisation
- Tests et faisabilité technologiques (GPS, accéléromètres, ...)
- Mise en relation Technologies-Applications
- Validation sur cas réels en vraie grandeur
- Élargir les champs d'application (hors manufacturing)

