

UNIVERSITE ABDELHAFID BOUSSOUF MILA

DEPARTEMENT MATHEMATIQUE

Simulation avancée

Dr. BOUNAIRA Soumaya
Enseignante-chercheuse

Date: Décembre 3, 2025

Introduction à la simulation à l'aide d'Arena

3.1 Introduction

La simulation est actuellement le seul outil disponible lorsqu'on désire étudier et comprendre le fonctionnement d'un système complexe que l'on ne parvient pas à formaliser par des méthodes analytiques. Ceci est souvent le cas des procédés industriels (par exemple la production d'une voiture) et/ou logistiques (par exemple le chargement d'un navire-container). Ces systèmes sont mieux représentés par une suite d'événements (par exemple *dépôt du châssis de la voiture sur un tapis roulant* ou *arrivée du navire au port*) que par un système d'équations.

La simulation à événements discrets est la simulation d'un système dont l'état ne peut changer que lors d'instantanés temporels distincts. En termes mathématiques nous pouvons dire que l'ensemble d'instantanés temporels auxquels le système peut changer est dénombrable. Un *événement* est la circonstance (entrée) qui permet au système de changer d'état. Notons que

- en absence d'événements l'état resterait indéfiniment le même, ce qui n'était pas forcément le cas dans les systèmes vus jusqu'ici.
- un événement pourrait ne pas déclencher un changement d'état.

En théorie, une simulation à événements discrets pourrait être menée de manière manuelle. En pratique, pour des systèmes avec un espace d'état de grande taille et avec un large ensemble d'événements, l'utilisation de l'ordinateur est incontournable.

Certains systèmes sont naturellement concevables comme étant discrets : c'est le cas d'un bureau de banque, d'un terminal maritime ou d'un atelier de production.

3.2 Définition des systèmes à événements discrets

Les **systèmes à événements discrets** (SED) sont des **modèles mathématiques** utilisés pour représenter et analyser des systèmes où les changements d'état se produisent à des moments discrets. Contrairement aux systèmes continus, où les changements s'effectuent de manière

continue et fluide, les SED changent d'état uniquement lors d'événements spécifiques, tels que l'arrivée d'un client dans un système de file d'attente ou le déclenchement d'un capteur.

3.2.1. Caractéristiques des SED

Les SED se caractérisent par plusieurs éléments clés :

- **Événements** : Un **événement** se produit d'une manière **instantanée** et provoque la **transition** de l'**état** d'une valeur discrète à une autre valeur
- **États** : étape dans l'évolution du système pendant laquelle il satisfait une condition, ou exécute une action ou attend un événement
- **Transitions** : Les changements entre les états.
- **Chronologie** : La séquence et le timing des événements peuvent fortement influencer le comportement du système.

La force des SED réside dans leur capacité à modéliser efficacement des systèmes tels que les chaînes de production, les réseaux informatiques, et les **systèmes de transport**.

Exemples :

1- Télerrupteur dans une maison

Lorsque l'on appuie sur un bouton poussoir, la réaction de l'éclairage associé dépend de son état courant (donc de son historique) : si la lumière est allumée, elle s'éteint, si elle est éteinte, elle s'allume.

Éléments du comportement du système

2 états : *Allumé* et *Eteint*.

2 transitions : *Allumé* → *Eteint* et *Eteint* → *Allumé*.

1 événement : *pression* sur un bouton poussoir.

2- Machine à café

Etats :

- Attente de pièce ;
- Descente du gobelet ;
- Versement de la poudre à café ;
- Versement de l'eau chaude ;
- Affichage "café est prêt"...

Evènements :

- Pièce insérée ;
- Gobelet présent ;
- Gobelet absent ;
- Timer terminé (compte-à-rebours)...

3.2.2. Types de modèles des systèmes à événements discrets

Il existe plusieurs formalismes de modélisation et d'analyse des SED, parmi lesquels on cite : les automates finis, les réseaux de Petri, grafcet, et l'algèbre des dioides, chacun ayant ses propres caractéristiques et applications :

- **Automates finis** : Représentent des systèmes où les états et les transitions sont clairement définis par des événements.
- **Réseaux de Pétri** : Utilisés pour modéliser les processus parallèles et synchronisés. Les réseaux de Pétri sont composés de places, de transitions et de jetons.
- **Graphes d'état et de transition** : Illustrent les relations entre différents états d'un système à travers des transitions déclenchées par des événements.
- **Processus de décision markoviens** : Permettent d'incorporer des décisions en modélisant les probabilités de transition entre différents états.

3.3. La simulation de Monte-Carlo

Le terme Monte-Carlo désigne généralement l'ensemble des méthodes et techniques d'estimation de quantités par échantillonnage itératif de modèles/équations représentés par ordinateur. De ce fait, cette terminologie est en quelque sorte synonyme de simulation informatique. Le terme Monte-Carlo tire son origine du casino de Monte-Carlo, en Principauté de Monaco, célèbre pour ses jeux de hasard. Il n'existe pas une seule méthode Monte-Carlo, mais plutôt un ensemble d'algorithmes et de techniques. En réalité, les concepts de génération de nombres aléatoires et de variables aléatoires, abordés précédemment, constituent le fondement des méthodes Monte-Carlo.

La méthode de Monte-Carlo a été inventée par John von Neumann et Stanislaw Ulam pendant la Seconde Guerre mondiale pour soutenir la prise de décision dans des conditions incertaines. Elle porte le nom d'un quartier d'une ville bien connue pour son casino, Monaco, car le hasard se trouve au cœur de cette approche de modélisation, comme à la roulette.

La simulation de Monte-Carlo est un type d'algorithme de calcul qui utilise des échantillons aléatoires répétés pour estimer la probabilité d'occurrence de différents résultats. En autre terme la simulation de Monte-Carlo est une technique mathématique utilisée pour estimer les résultats possibles d'un événement incertain.

Depuis leur introduction, les simulations de Monte-Carlo ont pu évaluer l'impact du risque dans de nombreux scénarios concrets, comme [l'intelligence artificielle](#), le cours des actions, les prévisions de vente, la gestion de projet et la tarification.

Elles offrent également un certain nombre d'avantages par rapport aux modèles prédictifs à entrées fixes, comme la possibilité d'effectuer une analyse de sensibilité ou de calculer les corrélations entre les entrées. Les analyses de sensibilité permettent aux décideurs de connaître l'impact des entrées individuelles sur un résultat donné, et les mises en corrélation leur permettent de comprendre les relations entre toutes les variables d'entrée.

3.4. Horloge a événements discrets

L'**horloge à événements discrets** constitue un élément central dans le fonctionnement d'une simulation à événements discrets. Elle permet de gérer l'évolution du temps simulé et de coordonner l'occurrence des événements qui provoquent les changements d'état du système. Contrairement à une horloge continue, l'horloge d'un système à événements discrets n'avance pas de manière uniforme, mais progresse par **sauts successifs**, d'un événement significatif au suivant.

3.4.1 Principe de fonctionnement

Dans un système à événements discrets, le temps simulé reste constant entre deux événements consécutifs. L'horloge avance uniquement lorsqu'un nouvel événement est exécuté. Ainsi, l'évolution temporelle du système est directement liée à la dynamique des événements, tels que :

- l'arrivée d'une entité dans le système,
- le début ou la fin d'un traitement,
- la libération ou la saisie d'une ressource,
- la sortie d'une entité du système.

L'horloge prend toujours comme valeur le **temps du prochain événement programmé**.

3.4.2 Liste des événements futurs

Le fonctionnement de l'horloge est étroitement lié à la **liste des événements futurs** (Future Event List – FEL). Cette liste contient l'ensemble des événements planifiés, chacun associé à une date d'occurrence.

À chaque étape de la simulation :

1. l'événement ayant la date la plus proche est sélectionné,
2. l'horloge est avancée à cette date,
3. l'événement est exécuté,
4. la liste des événements futurs est mise à jour.

Ce mécanisme garantit que les événements sont traités dans l'ordre chronologique.

3.4.3 Rôle de l'horloge dans la mise à jour de l'état du système

À chaque changement de l'horloge, l'état du système est mis à jour. Les variables d'état (nombre d'entités dans le système, état des ressources, longueur des files d'attente, etc.) ne sont modifiées que lors de l'exécution d'un événement. Entre deux événements, le système est considéré comme stable.

Ce principe permet :

- une modélisation efficace des systèmes complexes,
- une réduction du temps de calcul,
- une meilleure précision des résultats de simulation.

3.4.4 Horloge à événements discrets dans Arena

Dans Arena, la gestion de l'horloge à événements discrets est **entièrement automatique** et transparente pour l'utilisateur. Le logiciel se charge de :

- planifier les événements générés par les modules (Create, Process, Dispose, etc.),
- gérer la liste des événements futurs,
- faire avancer l'horloge simulée au moment opportun.

L'utilisateur définit uniquement les lois de temps (temps inter-arrivées, temps de service, délais, etc.), et Arena traduit ces informations en événements planifiés dans le moteur de simulation.

3.4.5 Exemple

Considérons un système de file d'attente avec un seul serveur :

- un client arrive à l'instant $t = 3$ minutes,
- la fin de son service est planifiée à $t = 7$ minutes.

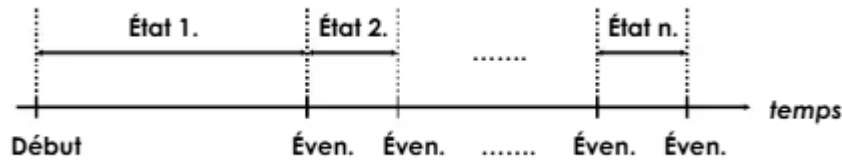
L'horloge simulée reste à $t = 3$ jusqu'à l'exécution de l'événement « fin de service ». Elle saute ensuite directement à $t = 7$, sans considérer les instants intermédiaires. Ce saut illustre le principe fondamental de l'horloge à événements discrets.

3.5. Modélisation d'un système à événements discrets

La modélisation d'un Système à Événements Discrets (SED) consiste à construire une représentation simplifiée mais fidèle de la réalité, où l'évolution du système est dictée par une succession d'événements.

3.4.1. Définition

- Modélisation des systèmes dont les états changent seulement à des instants discrets comme résultats d'évènements spécifiques.



- La simulation à événements discrets (SED) permet de modéliser des systèmes dynamiques dont le comportement est caractérisé par la circulation des entités au cours du temps.

Exemple:

- Ateliers ou d'usines au sein desquels circulent des produits, des transporteurs, des ouvriers,
- Services dans lesquels circulent des clients, des documents, des informations,
- Bâtiments ou plans en général dans lesquels circulent des clients, etc.

3.5.1. Composants d'un système

Les composants du système à définir :

- **Entité**: Objet d'intérêt au sein d'une simulation dont les activités sont modélisées.
- **Attribut**: Dans un système, il peut y avoir de nombreux types d'entités, chacun ayant des caractéristiques uniques, appelés attributs. Les attributs représentent des valeurs définies par l'utilisateur associées à des entités individuelles, comme le type de client, la taille du produit, etc. Toutes les entités ont le même ensemble d'attributs (priorité, heure d'arrivée) avec des valeurs différentes.
- **Activité** : Opération s'effectuant sur une période de temps déterminée.
- **État** : Collection de variables nécessaire pour décrire le système à tout moment par rapport aux objectifs de l'étude.
- **Événement** : Réalisation instantanée qui correspond à un changement d'état d'objet.

3.5.2. Etapes de la modélisation d'un SED

La modélisation d'un système à événements discrets suit une démarche méthodique composée de plusieurs étapes :

a) Analyse du système réel

Cette étape consiste à observer et comprendre le fonctionnement du système étudié. Elle inclut l'identification des flux, des contraintes, des règles de gestion et des interactions entre les différents éléments.

b) Définition des entités

Les entités représentent les objets dynamiques qui circulent dans le système (clients, pièces, produits, dossiers, etc.). Chaque entité peut posséder des attributs qui décrivent ses caractéristiques.

c) Identification des ressources

Les ressources sont les moyens nécessaires au traitement des entités (machines, opérateurs, serveurs). Leur disponibilité et leur capacité influencent directement les performances du système.

d) Définition des événements

Les événements sont les instants où l'état du système change (arrivées, débuts et fins de service, pannes, sorties). Ils structurent l'évolution temporelle du modèle.

e) Modélisation des files d'attente

Les files d'attente apparaissent lorsque la demande dépasse la capacité des ressources. Leur discipline de gestion (FIFO, priorité, etc.) doit être clairement définie.

f) Définition des indicateurs de performance

Les indicateurs permettent d'évaluer le système, tels que :

4. temps d'attente moyen,
5. temps de séjour dans le système,
6. taux d'utilisation des ressources,
7. longueur moyenne des files.

7.3.1. Structure et composants du modèle de simulation à événements discrets

La modélisation sous Arena repose sur une distinction claire entre les flux dynamiques et les paramètres de contrôle. Nous pouvons diviser ces composants en cinq catégories essentielles :

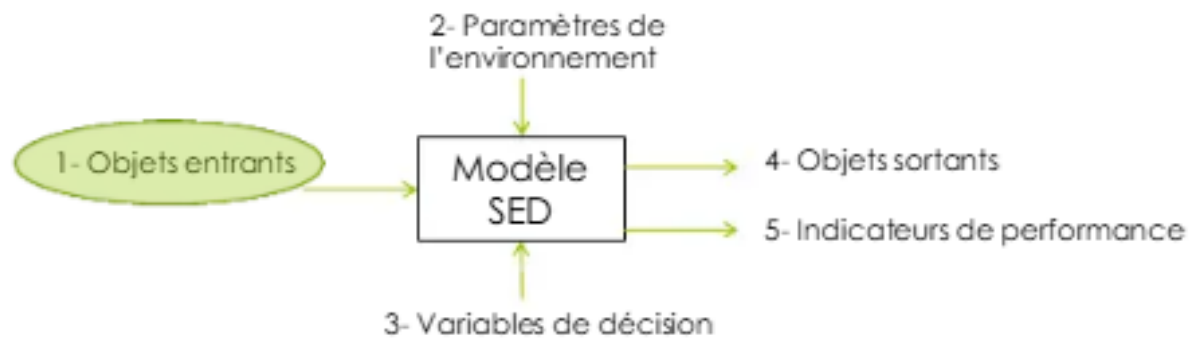


Figure 3.1. Modèle du simulation a événements discrets

- 1- Les objets entrants sont spécifiés par les entités, ils constituent les composants dynamiques qui circulent dans le modèle (pièces, clients, ordre de fabrication, etc.).
Ces entités sont caractérisées par des données appelées attributs (couleur, taille, poids, référence, type, etc.).
La création (génération) des entités dans le modèle doit être spécifiées (suivant des lois de probabilité, suivant un Schedule, etc.).
- 2- Paramètres de l'environnement : déterminent des grandeurs sur lesquels on considère n'avoir aucune influence (non contrôlable) mais pouvant influencer le système (distances séparant des cellules d'usinages, des taux de panne, etc.)
- 3- Variables de décision sont les grandeurs dont on veut étudier l'influence sur le système, (Le nombre de guichets de douane, le nombre de chariots élévateurs, règles de routage, etc.)
- 4- Les objets sortants sont des objets sur lesquels on cherchera à faire des statistiques. Dans le cas d'une usine les pièces usinées, ou dans un aéroport, les passagers ayant récupérés leurs valises.
- 5- Les indicateurs de performances permettent de compléter les observations sur les objets sortants pour << juger >> de la performance du modèle observé, (longueurs de files d'attente, des temps de séjour, etc.)

7.4. Implémentation exécution du modèle à l'aide d'Arena

L'implémentation dans Arena est le passage du modèle conceptuel (logigramme) au modèle numérique. Cette étape repose sur l'assemblage de modules graphiques qui encapsulent la logique de simulation et les lois de probabilité.

3.6.1. Construction du flux logique (Flowchart View)

L'implémentation suit généralement une séquence de modules du panneau **Basic Process** :

1. **Module Create** : C'est le point d'entrée. On y définit la loi de probabilité des inter-arrivées (ex: EXPO(10) pour une moyenne de 10 minutes).
2. **Module Process** : C'est le cœur de l'activité. On y définit la règle de saisie des ressources. La méthode la plus rigoureuse est le **Seize Delay Release** :
 - **Seize** : L'entité occupe la ressource (ex: un guichet).
 - **Delay** : L'entité subit un délai de traitement (ex: TRIA(5, 10, 15) pour une loi triangulaire).
 - **Release** : L'entité libère la ressource pour l'entité suivante dans la file d'attente.
3. **Module Decide** : Utilisé pour les branchements (ex: séparation des entités pour contrôle qualité).
4. **Module Dispose** : Marque la fin de vie de l'entité et déclenche l'enregistrement des statistiques de sortie.

3.6.2. Configuration des données (Data View)

En parallèle du flux, l'implémentation nécessite le paramétrage des feuilles de calcul (*Spreadsheets*) pour définir les objets statiques :

- **Resource** : Définition de la capacité (ex: 2 serveurs disponibles).
- **Variable** : Définition de variables globales (ex: coût unitaire).
- **Schedule** : Planification des ressources (ex: pauses déjeuner ou changements d'équipes).

3.6.3. Paramétrage de l'exécution (Run Setup)

Avant de lancer la simulation, il est impératif de configurer le contrôle de l'exécution dans le menu **Run > Setup**. En Master de Mathématiques Appliquées, trois paramètres sont cruciaux pour la validité statistique :

- **Replication Length** : La durée d'une simulation (ex: 8 heures de travail).
- **Number of Replications** : Pour obtenir une convergence statistique et un intervalle de confiance fiable, il faut exécuter le modèle plusieurs fois avec des graines aléatoires différentes.
- **Warm-up Period** : Si l'on étudie le système en régime permanent, on définit une période de "chauffe" pour éliminer le biais dû à l'état initial (système vide au départ).

3.6.4. Le processus d'exécution

Lors de l'exécution, Arena compile le modèle en langage **SIMAN**. Le moteur de simulation gère alors :

1. La génération des nombres pseudo-aléatoires.
2. Le déplacement des entités à travers les modules.
3. La mise à jour de l'horloge **TNOW** à chaque événement.

Pendant l'exécution, Arena affiche une animation visuelle qui permet de vérifier la validité du flux (détection visuelle de goulots d'étranglement ou de files d'attente anormales).

7.5. Analyse des résultats (Output Analysis)

L'analyse des résultats est l'étape finale qui permet de transformer les données brutes de simulation en informations d'aide à la décision. Dans Arena, cela passe par l'examen des rapports (Reports) et l'utilisation d'outils statistiques pour valider la précision des résultats.

7.5.1. Les indicateurs de performance clés (KPI)

À la fin de l'exécution, Arena génère automatiquement un rapport structuré présentant les moyennes, les valeurs minimales et maximales pour :

- **Les Entités** : Temps total dans le système (*Cycle Time*), temps d'attente (*Wait Time*).
- **Les Files d'attente** : Nombre moyen d'entités en attente, temps d'attente maximum.
- **Les Ressources** : Taux d'utilisation instantané et moyen. Une utilisation proche de 100% indique souvent un goulot d'étranglement.

7.5.2. L'outil "Output Analyzer"

Arena propose un outil complémentaire nommé **Output Analyzer**. Il permet de :

- Comparer deux scénarios (ex: deux configurations d'usine) via des tests de comparaison de moyennes (Test-t).
- Visualiser la distribution des données sous forme d'histogrammes.
- Vérifier la corrélation entre les variables.

7.6. Conclusion

Dans un contexte de simulation avancée, la qualité de la modélisation conditionne directement la pertinence des résultats obtenus. Un modèle bien conçu permet :

- de tester différentes configurations sans perturber le système réel,
- d'identifier les goulots d'étranglement,
- d'optimiser l'allocation des ressources,
- d'aider à la prise de décision.

Arena offre des outils puissants pour représenter des systèmes complexes, intégrer des lois statistiques, et analyser en profondeur le comportement des systèmes à événements discrets.

