



Un DSL pour représenter et analyser des systèmes à événements discrets

MSR 2025

Pascal André, Hugo Locteau, Yasmina Dali Youcef, Erwan Bousse, Olivier Cardin

LS2N, Nantes Université, France

Sommaire

2

- Contexte
- Besoins et solutions
- DSL RMS
- Illustration
- Conclusion

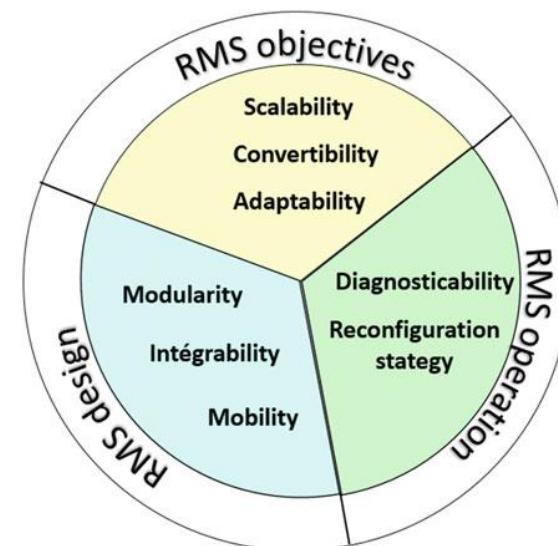
□ Modélisation et analyse de systèmes à événements discrets



Contexte

Systèmes manufacturiers reconfigurable (RMS : reconfigurable manufacturing systems)

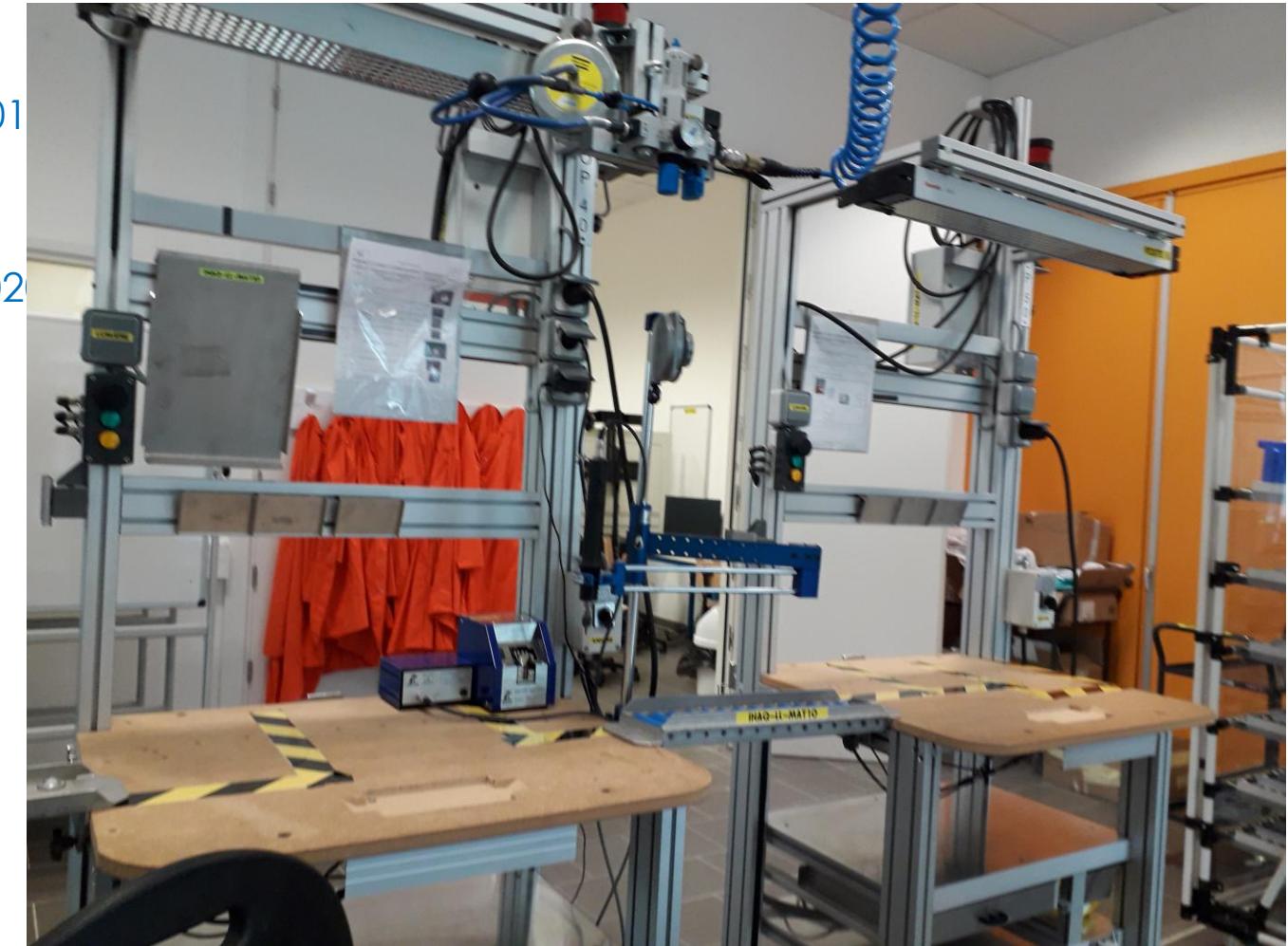
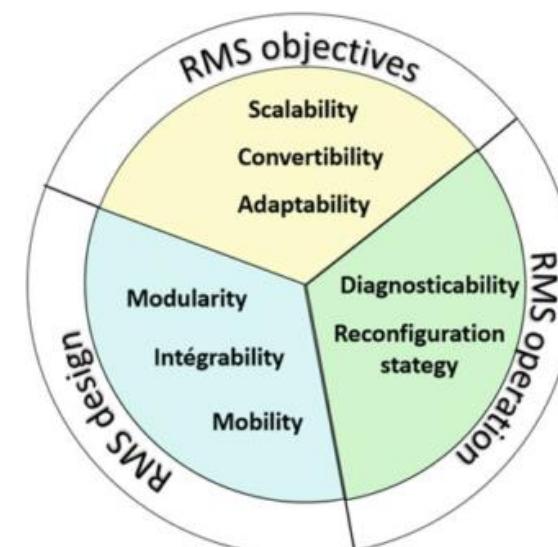
- From FMS to RMS [Koren et al 1999, 2010]
- Application scenario [Bi et al 2008]
- Classification [Capawa Fotsoh et al 2020]



Contexte

Systèmes manufacturiers reconfigurable (RMS : reconfigurable manufacturing systems)

- From FMS to RMS [Koren et al 1999, 2011]
- Application scenario [Bi et al 2008]
- Classification [Capawa Fotsoh et al 2020]



Focus sur l'usine/la partie prenante de la ligne

Contexte

5



Cible

- PME
- Production personnalisée, évolutive
- Configuration de courte durée, réactivité
- Faible expertise en Optimisation/Simulation
- Solution peu onéreuse

Rôles

- Planificateur (ligne de production)
- Ouvrier (opérateur de machine)

Performance

- Evaluation
- Reconfiguration

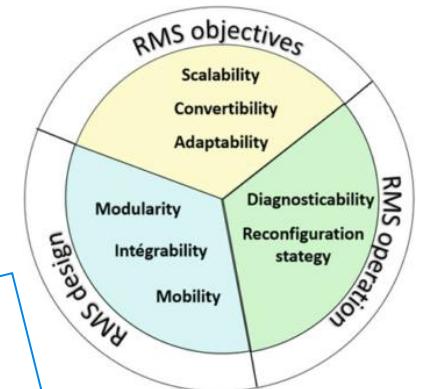


<https://digitalshoestring.net/>

Kaiser et al. - 2025 - Effective integration of low-cost digital manufacturing systems

Il manque un outil pour raisonner sur les configurations et reconfigurations

La notion de configuration est réduite à son expression la plus simple pour réduire la complexité des calculs



Sommaire

Un DSL pour représenter et analyser des systèmes à événements discrets

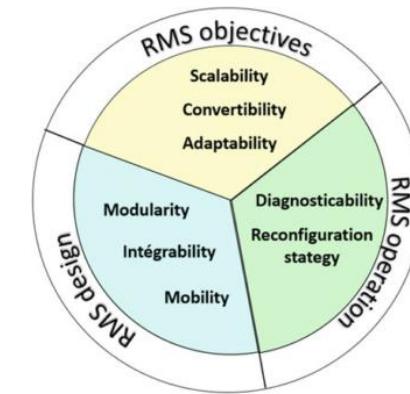
- Contexte
- Besoins et solutions
- DSL RMS
- Illustration
- Conclusion



Besoins

Un DSL pour représenter et analyser des systèmes à événements discrets

- Modèles conceptuels pour les RMS
 - 1. Modélisation
 - 1. Structure (modulaire)
 - 2. Dynamique
 - 3. Fonctions
 - 2. Analyse
 - 1. Vérification formelle
 - 2. Simulation
 - 3. Expressivité (utilisateurs non-experts)
 - 4. Scalabilité



Solutions outils numériques

- Outils de vérification formelle
 - Réseaux de petri
 - Automates communiquants
- -> Romeo, Tina, CPN Tools, UPAAL
- Simulateurs à évènements discrets
 - Arena
 - FlexSim

Manque de modularité ou d'expressivité

Requiert un niveau d'expertise élevé



Besoins et solutions

liste non exhaustive

Un DSL pour représenter et analyser des systèmes à événements discrets

H. Locteau - MSR'25

Solution	Modelling (expressivity)				User skill		Verification		Performance evaluation			Outils
	Structure	Operations	Dynamics	Customisation	Expertise	Effort	Proving system	KPI def.	Simulation	Comparison		
MES												plugged, extraction
PetriNets (Romeo, Tina, CPNTools)												editors, model checking
process algebra (uppaal)												editors, model checking
DES (FlexSim)												editors, simulator, boards
Optimisation methods												simulator, optimisation
DSL												editors, checker, runner

Sommaire

10

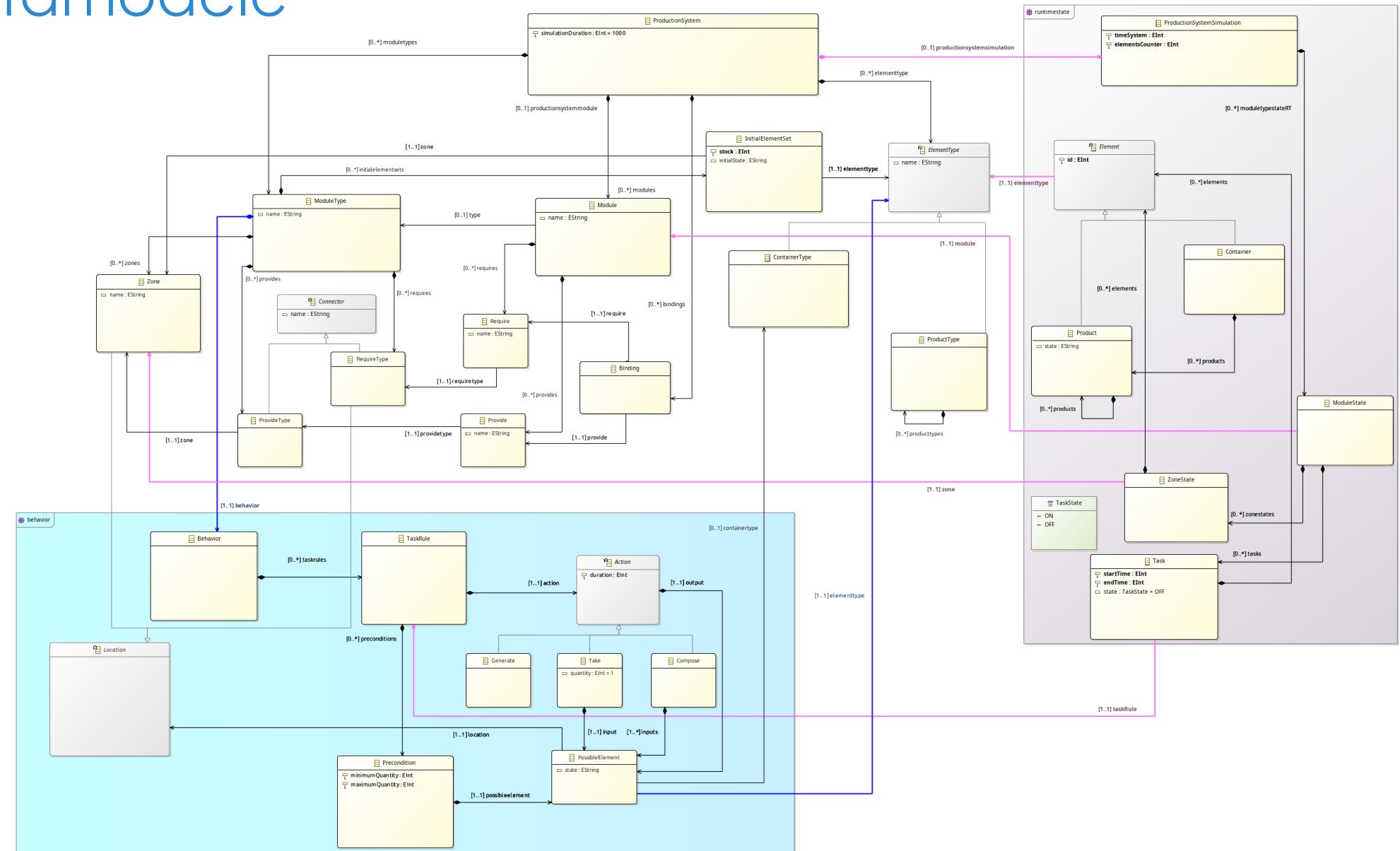


- Contexte
- Besoins et solutions
- DSL RMS
 - métamodèle
 - sémantique
 - exécution
- Illustration
- Conclusion



DSL RMS métamodèle

11



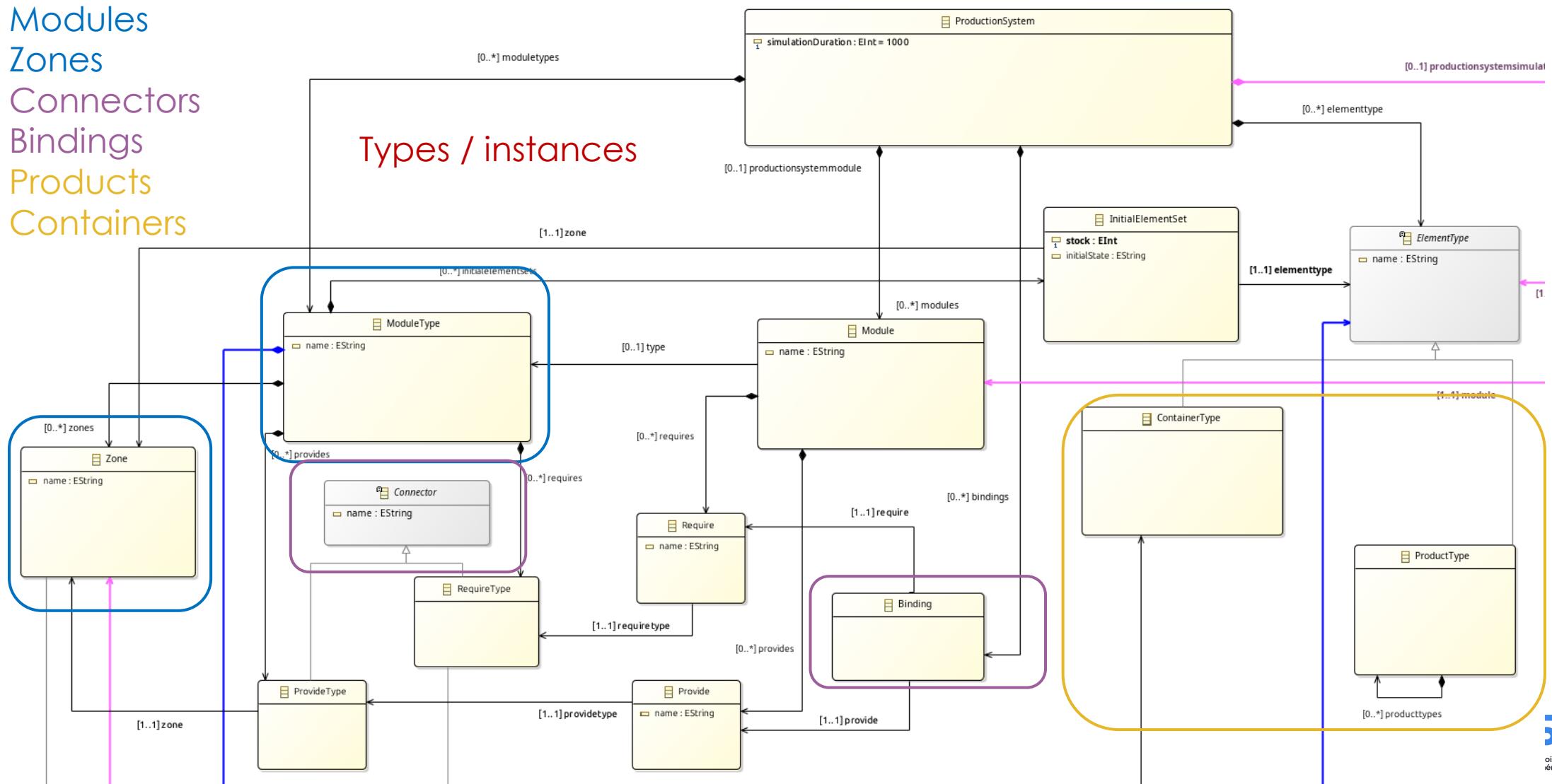
DSL RMS

métamodèle / structure

Un DSL pour représenter et analyser des systèmes à événements discrets

Modules
Zones
Connectors
Bindings
Products
Containers

Types / instances



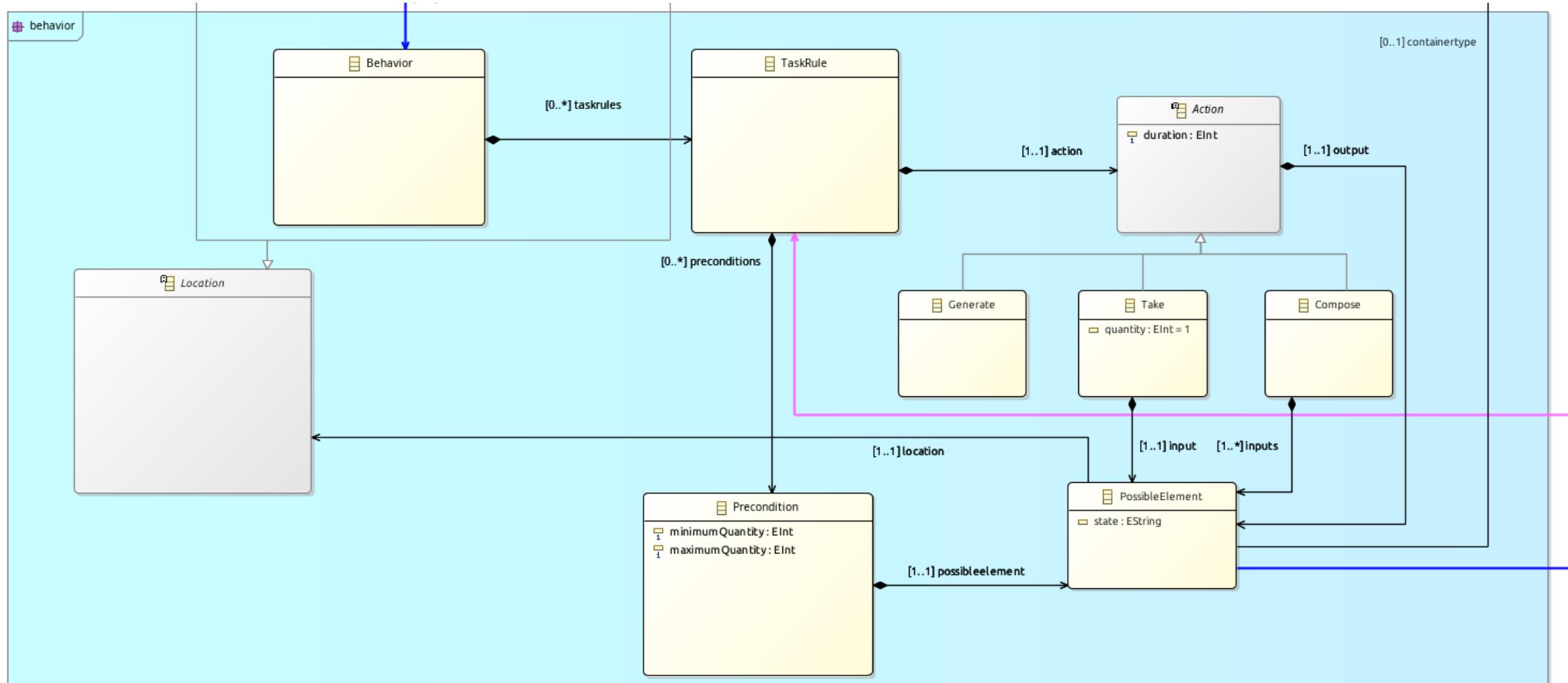
DSL RMS

métamodèle / comportement

Tasks rules
Actions

Un DSL pour représenter et analyser des systèmes à événements discrets

H. Locteau - MSR'25



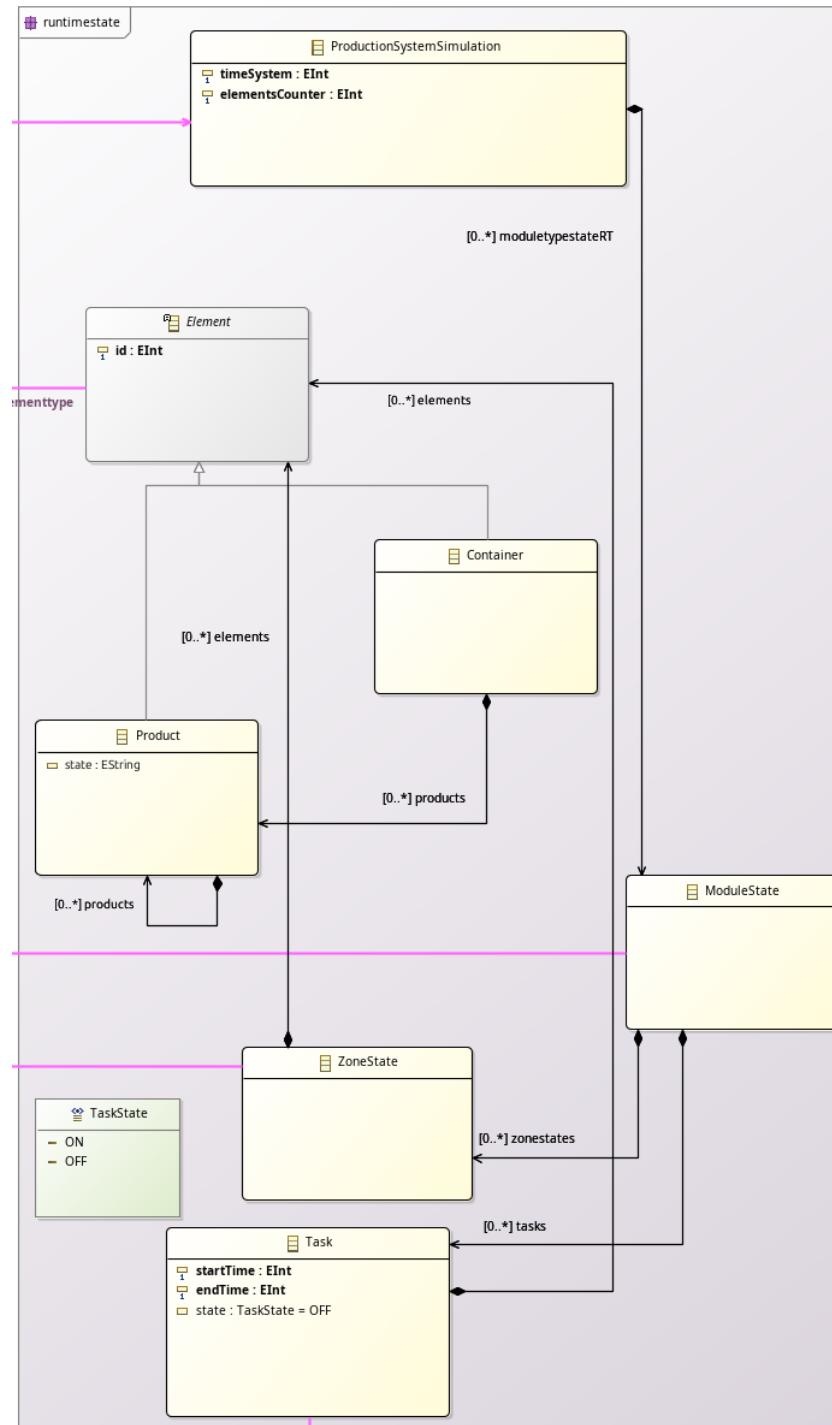
RMS DSL

métamodèle / runtime

Etats des

- Modules
- Zones
- Products / Containers
- Tasks

Sémantique opérationnelle:
java code généré + java
code



RMS DSL

De la définition à l'exécution

Un DSL pour représenter et analyser des systèmes à événements discrets

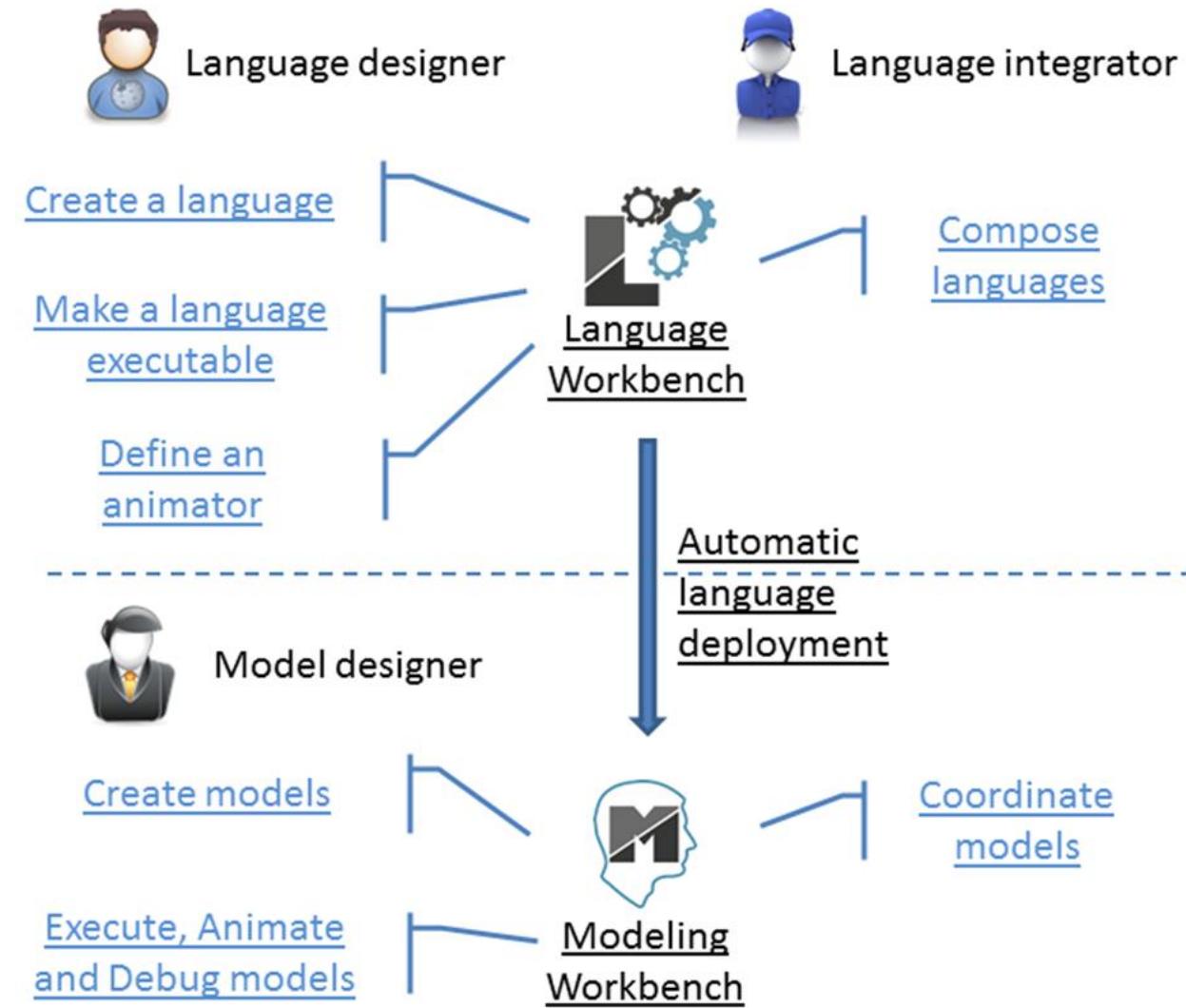
H. Locteau - MSR'25

Gemoc Studio

- Support Eclipse
 - e.g. AIRD, Xtend, Xtext, Sirius
- Génération de code Java/ code custom
- Addons
 - Plugin de trace

<https://gemoc.org/>

Language Driven Engineering



Sommaire

- Contexte
- Besoins et solutions
- DSL RMS
- Illustration
- Conclusion

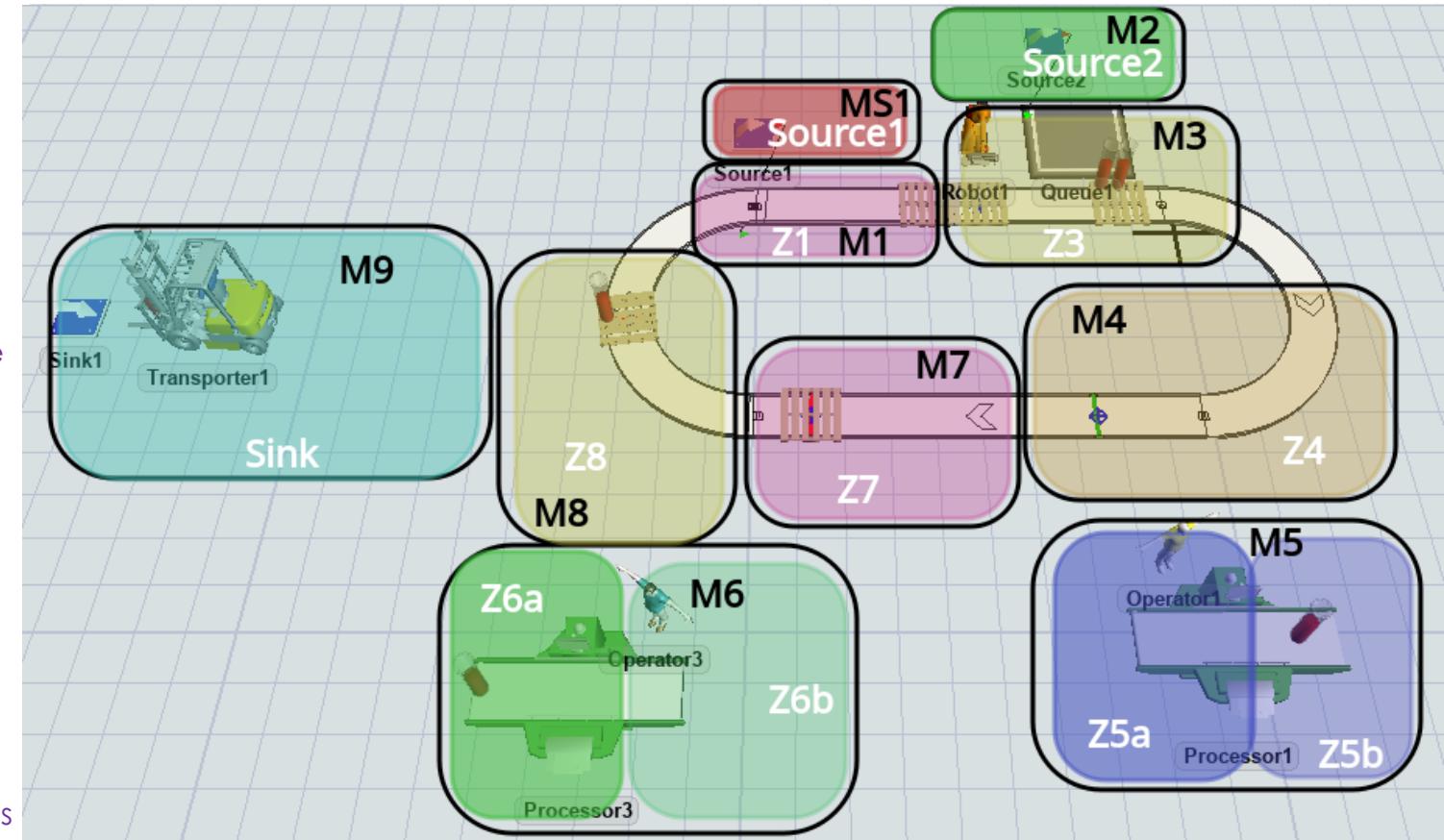


Exemple : cas d'étude
simplifié

Cas d'étude simplifié

Organisation de la ligne

1. Tapis roulant (convoyeur)
2. source 1 = positionne des palettes sur le convoyeur
3. Un robot pose deux composants sur une palette
4. Un opérateur humain prend l'un des composants pour qu'une machine (Processeur 1) le traite
5. Un opérateur humain emmène le deuxième composant à la seconde machine (Processeur 2)
6. Les 2 opérateurs ramènent les composants qui ont subi le traitement
7. Un chariot (Transporteur 1), transporte les composants un à un vers la sortie



organisation modulaire =
définit les périmètres de responsabilités



Cas d'étude simplifié

Un DSL pour représenter et analyser des systèmes à événements discrets

Cas d'étude simplifié

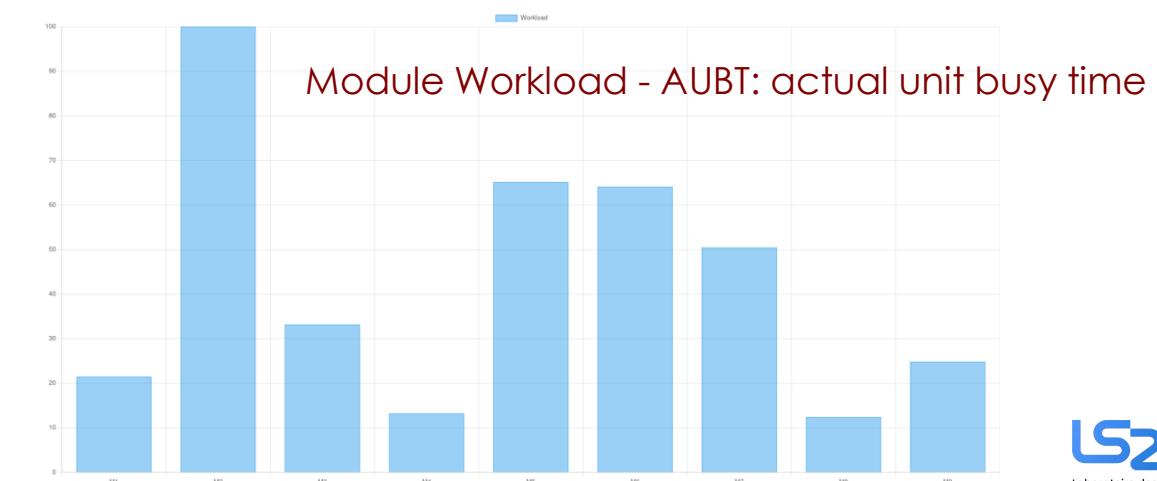
Résultats de simulation

- Logs et Trace d'exécution
- Calcul de KPI :
- A partir de l'instrumentation du code (observers) :
 - (i) temps du système en secondes (0 à l'état initial),
 - (ii) mouvements des conteneurs et produits d'une zone à l'autre,
 - (iii) Stockage des conteneurs ou produits,
 - (iv) Changement d'état des produits.
- A partir de l'analyse de la trace d'exécution :
 - (a) 76 produits ont atteint la zone Sink-M9
 - (b) 96 produits attendent dans le stock Source2-M2
 - (c) Le module M5 a fonctionné 1982 secondes
 - (d) Le module M6 a fonctionné 1558 secondes
 - (e) Le délai de production du produit 20 est de 140 secondes. C'est le temps mis par ce produit pour quitter la Source2 et atteindre la zone Sink

Moteur de simulation Java

19

```
Default MessagingSystem console
-----
1838
Product Component with id 64 has been removed from Source2 (state: initial)
Product Component with id 61 has been removed from Container 0 (state: initial)
Container Pallet with id 0 moved to Zone Z7
Product Component with id 60 has been removed from Z5b (state: completed)
-----
1844
Container Pallet with id 1 moved to Zone Z4
Product Component with id 62 has been removed from Container 1 (state: initial)
Product Component with id 58 moved to Zone Sink (state: completed)
Product Component with id 59 has been removed from Container 3 (state: completed)
```



Interface Web – Cas Usine ecole

Un DSL pour représenter et analyser des systèmes à événements discrets

Configuration de la ligne

Rechercher un module

Category

- Stock
 - Stock 1
 - Stock 2
 - Stock 3
- Workstation 1
 - Workstation 1 1
 - Workstation 1 2
 - Workstation 1 3
- Workstation 2
 - Workstation 2 1
 - Workstation 2 2

Gamme

Exporter

Importer

Configuration 01 - Prototype

Configuration 01 - Prototype

Ajouter configuration

Vérifier

Simuler

Paramètres

Liaisons

Stock 1	+
Workstation 1 1	+
Workstation 2 1	+
Workstation 3 1	+
Workstation 4 1	+
Workstation 5 1	+

Sommaire

21

- Contexte
- Besoins et solutions
- DSL RMS
- Illustration
- Conclusion



Conclusion

Un DSL pour modéliser des RMS et évaluer leur performance

- La cible sont les utilisateurs spécialistes des RMS (et pas des techniques de modélisation)
- Les outils numériques existants (vérifications formelles, simulateurs) manquent de modularité, d'expressivité, ou demandent un haut niveau d'expertise
- Le DSL s'adapte au domaine visé, notamment pour nommer les concepts du langages



Conclusion

Un DSL pour modéliser des RMS et évaluer leur performance

- La cible sont les utilisateurs spécialistes des RMS (et pas des techniques de modélisation)
- Les outils numériques existants (vérifications formelles, simulateurs) manquent de modularité, d'expressivité, ou demandent un haut niveau d'expertise
- Le DSL s'adapte au domaine visé, notamment pour nommer les concepts du langages
- **Le DSL n'est pas un concurrent des autres outils, qui sont bien plus adaptés dans d'autres contextes**



Conclusion

Un DSL pour modéliser des RMS et évaluer leur performance

- La cible sont les utilisateurs spécialistes des RMS (et pas des techniques de modélisation)
- Les outils numériques existants (vérifications formelles, simulateurs) manquent de modularité, d'expressivité, ou demandent un haut niveau d'expertise
- Le DSL s'adapte au domaine visé, notamment pour nommer les concepts du langages
- **Le DSL n'est pas un concurrent des autres outils, qui sont bien plus adaptés dans d'autres contextes**

Etat actuel

1. Spécification du langage (DSL) & implémentation avec Gemoc
2. Vérification et évaluation (simulation et mesures de KPI)
3. Instanciation sur différents cas d'études



Conclusion perspectives

Travaux futurs

- **DSL**
 - Etendre le langage d'actions
 - Ajouter la notion de scénario / à séparer des paramètres de configuration
- **Méthodologie**
 - Ajouter des bibliothèques de modules
 - Comparer des configurations, décider la prochaine reconfiguration
- **Outilage**
 - Améliorer l'interface web pour les utilisateurs – modélisation visuelle, abstraction vs layout
 - Assistance au design de configuration
- **Application** à d'autres contextes et cas d'études





ANR Rodic



Un DSL pour représenter et analyser des systèmes à événements discrets

Merci pour votre
attention

Cet travail est suporté par l'Agence Nationale de la Recherche (ANR)
sous le projet RODIC, numéro ANR-21-CE10-0017.



appendix