

Modèles, systèmes, hétérogénéité

Soutenance d'HdR

Frédéric Boulanger

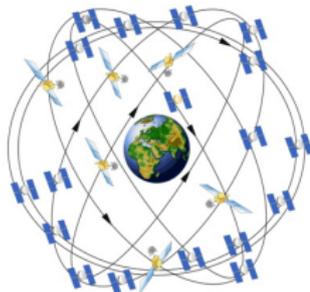
Supélec E3S – Département Informatique

8 juin 2011

Contexte

Modélisation et validation de systèmes logiciels enfouis

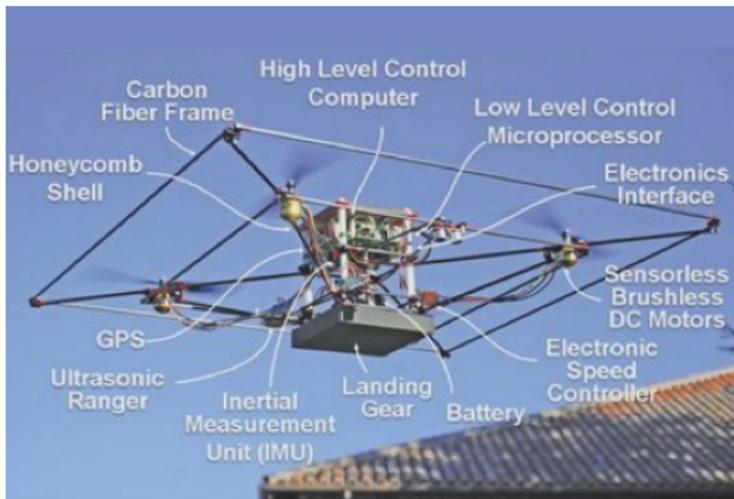
- systèmes réactifs
- fort lien avec l'environnement d'exécution



Hétérogénéité

- Différents métiers
- Différentes méthodes et différents outils de modélisation

Starmac (Stanford/Berkeley)



Équations différentielles

Fonctions du plan complexe

Mécanique

Systèmes logiques

Automatique

Traitement du signal

Télécommunications

Automates

Aérodynamique

Énergie

Événements discrets

Systèmes échantillonnés

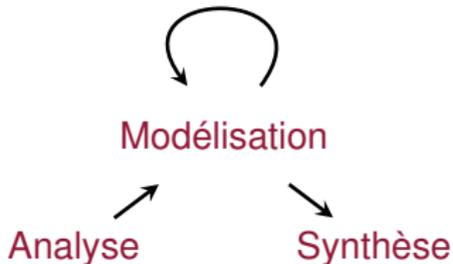
Nécessité d'un modèle de la totalité du système (+ environnement)

- Complexité croissante des systèmes
 - Validation au plus tôt de l'architecture
- ⇒ Vision globale du système, intégration des modèles métiers

Modélisation multi-paradigme

Application de l'ingénierie dirigée par les modèles à des systèmes modélisés selon plusieurs approches ou méthodes de modélisation

abstraction, raffinement, transformation



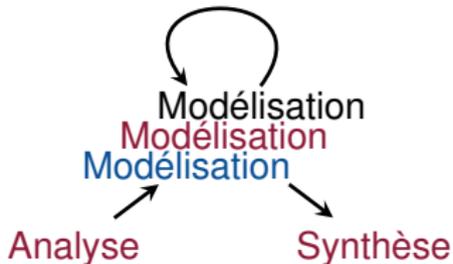
Nécessité d'un modèle de la totalité du système (+ environnement)

- Complexité croissante des systèmes
 - Validation au plus tôt de l'architecture
- ⇒ Vision globale du système, intégration des modèles métiers

Modélisation multi-paradigme

Application de l'ingénierie dirigée par les modèles à des systèmes modélisés selon plusieurs approches ou méthodes de modélisation

abstraction, raffinement, transformation



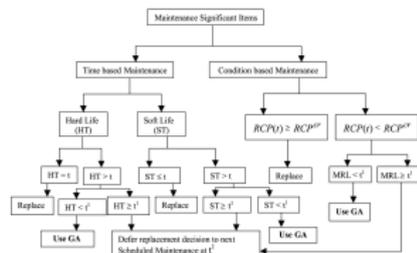
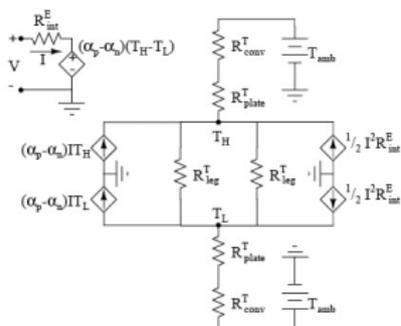
Système

- Simple objet d'étude ? Entité modélisée ?
- Entité structurée, obtenue par assemblage de composants ?
- Définir un système = isoler une partie du monde pour l'étudier
 - Interface traversée par des flux de matière, d'énergie et d'information
 - Interactions entre les composants dans la structure



Modèle

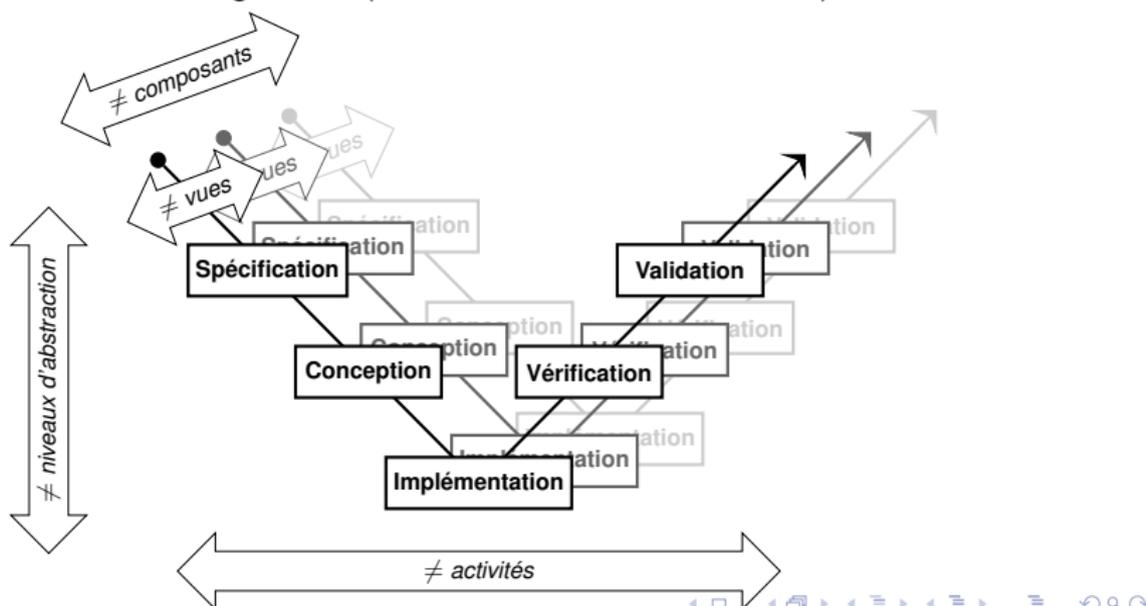
- Représentation abstraite d'un système pour un objectif particulier
- Retient les caractéristiques du système pertinentes pour l'objectif
- Ignore le plus possible les détails non pertinents



Modèle hétérogène

Modèle qui fait appel à différents langages de modélisation

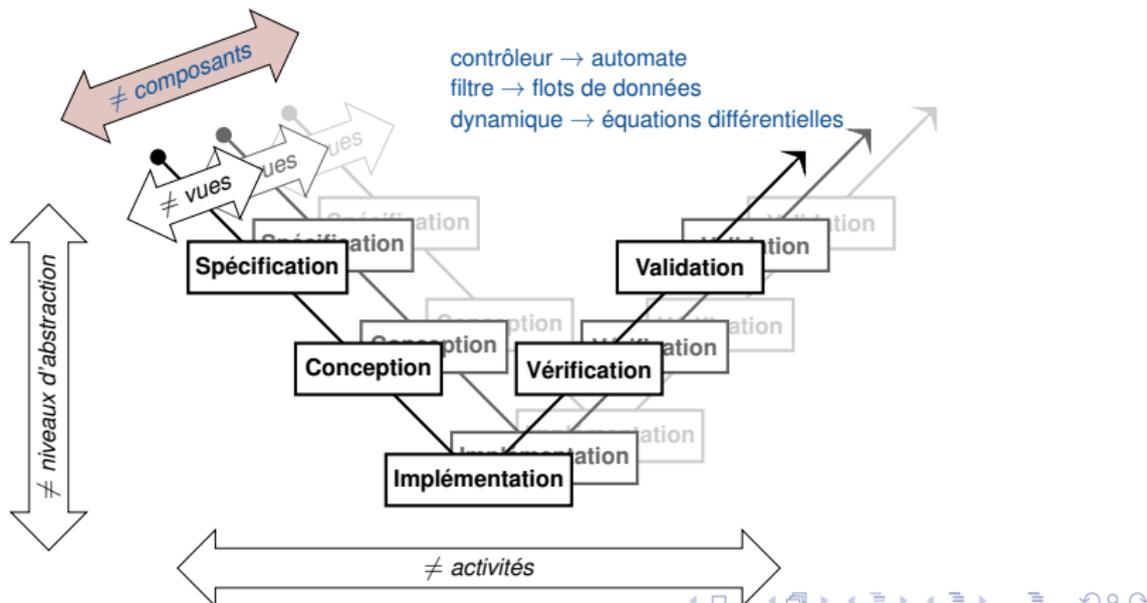
Quatre sources d'hétérogénéité (thèse de Cécile Hardebolle)



Modèle hétérogène

Modèle qui fait appel à différents langages de modélisation

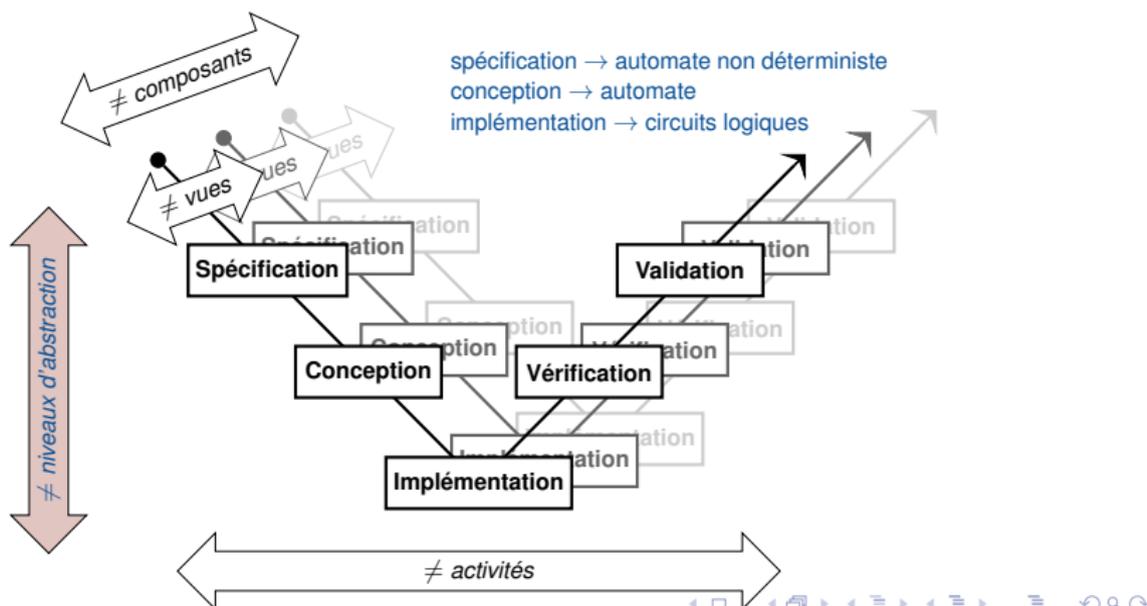
Différents composants font appels à différents paradigmes



Modèle hétérogène

Modèle qui fait appel à différents langages de modélisation

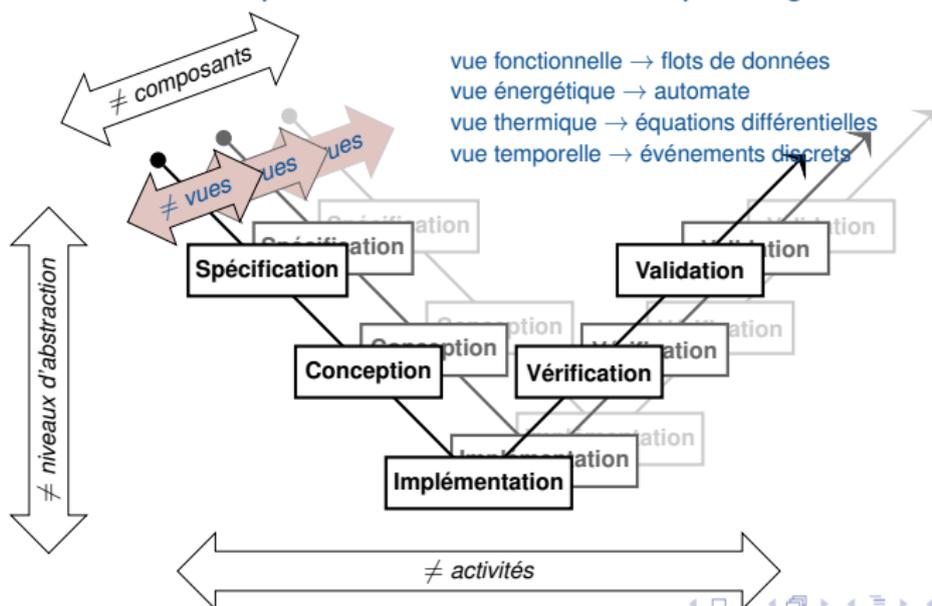
Différents formalismes sont utilisés à différents niveaux d'abstraction



Modèle hétérogène

Modèle qui fait appel à différents langages de modélisation

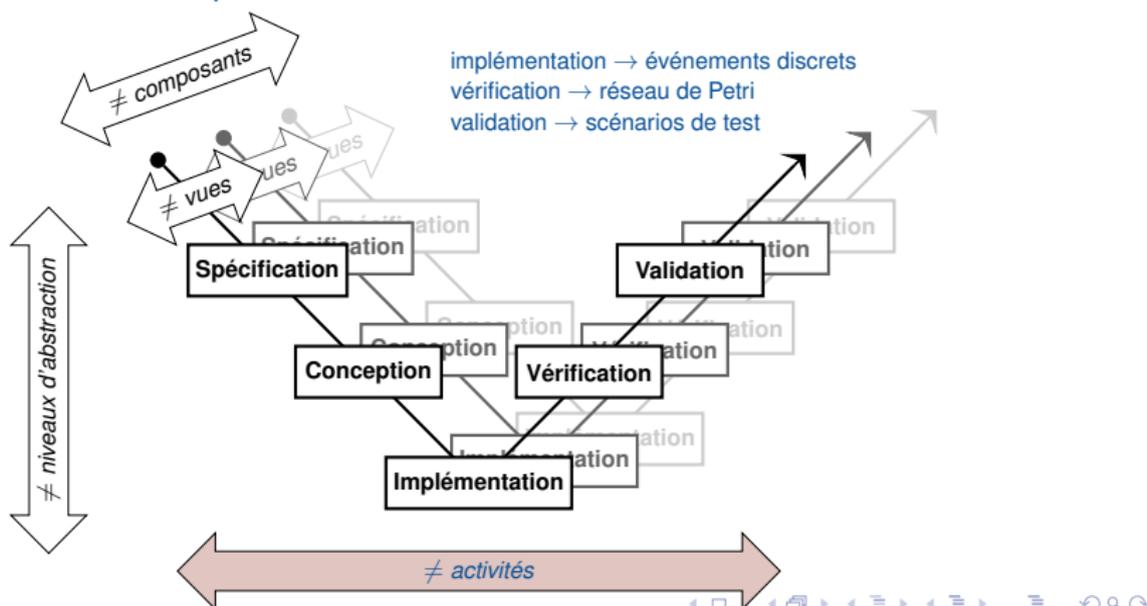
Différentes vues d'un composant utilisent différents paradigmes



Modèle hétérogène

Modèle qui fait appel à différents langages de modélisation

Différentes activités requièrent des formalismes différents



Modèle hétérogène

- Prise en compte de plusieurs langages
- Principe : langage \rightarrow modèle de calcul (syntaxe commune)

Modèle de calcul

- Ensemble de règles pour la composition du comportement des composants d'un modèle. Sémantique de la structure du modèle
- Algorithme de résolution des règles \Rightarrow modèle d'exécution

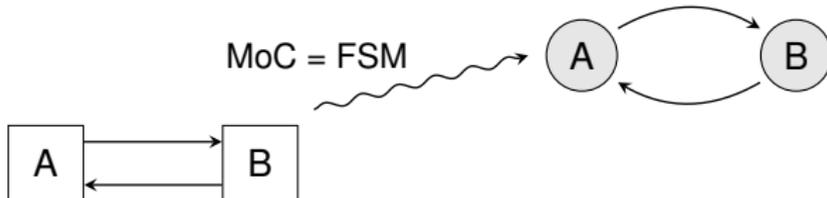


Modèle hétérogène

- Prise en compte de plusieurs langages
- Principe : langage \rightarrow modèle de calcul (syntaxe commune)

Modèle de calcul

- Ensemble de règles pour la composition du comportement des composants d'un modèle. Sémantique de la structure du modèle
- Algorithme de résolution des règles \Rightarrow modèle d'exécution

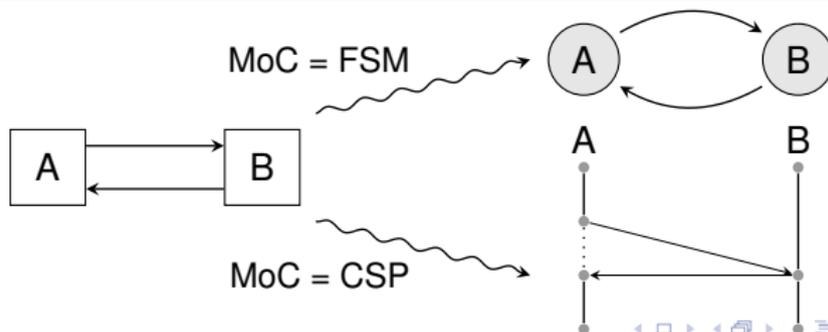


Modèle hétérogène

- Prise en compte de plusieurs langages
- Principe : langage \rightarrow modèle de calcul (syntaxe commune)

Modèle de calcul

- Ensemble de règles pour la composition du comportement des composants d'un modèle. Sémantique de la structure du modèle
- Algorithme de résolution des règles \Rightarrow modèle d'exécution



Machine d'exécution (libSync)

- Environnement nécessaire à l'exécution correcte d'un modèle
- Communication entre le modèle et son environnement



Rôles d'une machine d'exécution (thèse M. Feredj)

Préservation de la sémantique

- Fournir aux composants un environnement qui respecte la sémantique du langage de modélisation
- Combiner le comportement des composants du modèle

Adaptation sémantique

- Gérer l'interface entre le modèle et son environnement, qui n'obéit pas nécessairement aux mêmes règles

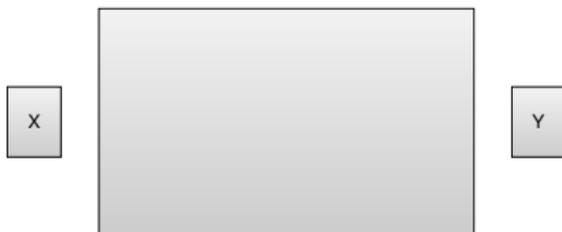
De Ptolemy (Berkeley) à ModHel'X (Supélec)

- Plateforme expérimentale pour l'exécution de modèles hétérogènes
- Généralise les travaux précédents, s'affranchit des limitations de Ptolemy

ModHel'X (Projet Usine Logicielle, thèse de C. Hardebolle)

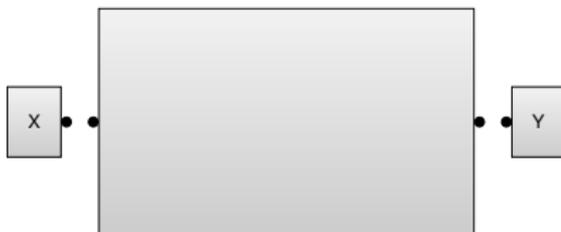
- Meta-modèle + moteur d'exécution générique :
 - `schedule` choix du prochain composant à observer
 - `update` mise à jour de l'interface d'un composant
 - `propagate` propagation des informations aux autres composants
- Comportement d'un modèle = point fixe de $propagate \circ update \circ schedule$
- Existence, unicité, atteignabilité par itération du point fixe ?

Unité de comportement : **Block**, « boîte noire » avec interface



Unité de comportement : **Block**, « boîte noire » avec interface

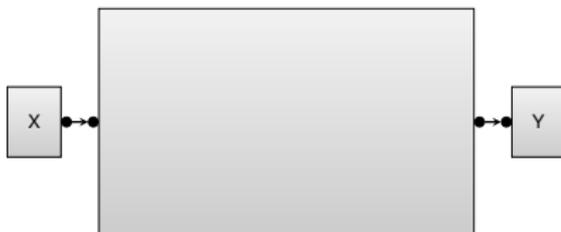
Unité d'interface : **Pin**, permet de fournir et d'obtenir des informations



Unité de comportement : **Block**, « boîte noire » avec interface

Unité d'interface : **Pin**, permet de fournir et d'obtenir des informations

Structure : **Relations** entre pins, sémantique définie par le MoC



Unité de comportement : **Block**, « boîte noire » avec interface

Unité d'interface : **Pin**, permet de fournir et d'obtenir des informations

Structure : **Relations** entre pins, sémantique définie par le MoC

Modèle : **Model** = structure + MoC



Unité de comportement : **Block**, « boîte noire » avec interface

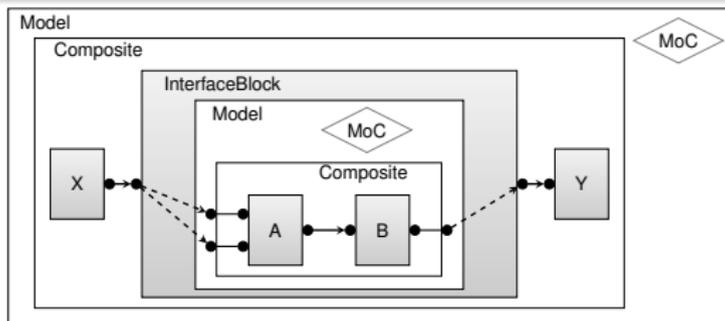
Unité d'interface : **Pin**, permet de fournir et d'obtenir des informations

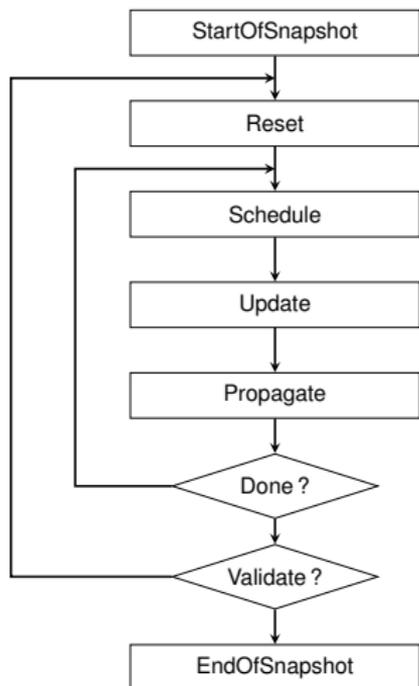
Structure : **Relations** entre pins, sémantique définie par le MoC

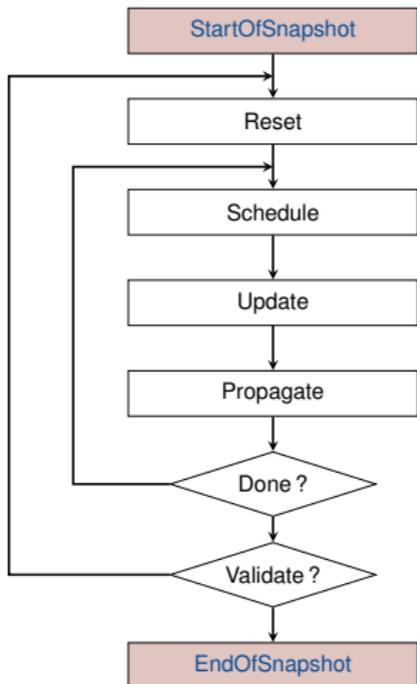
Modèle : **Model** = structure + MoC

Hétérogénéité hiérarchique

InterfaceBlock : comportement décrit par un **Model**



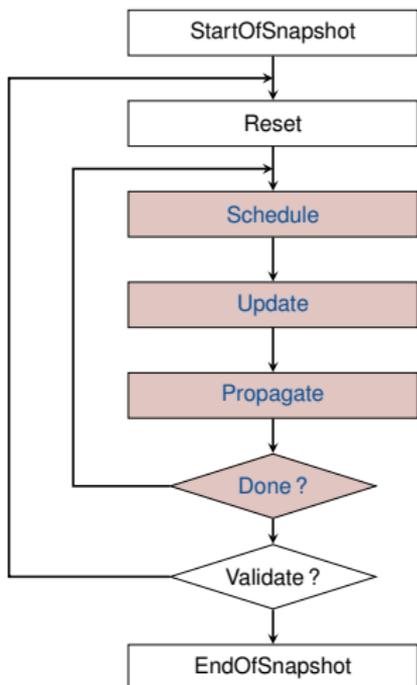




Comportement d'un modèle = suite d'observations

Approche synchrone de l'observation des modèles :

- aucune communication avec l'environnement pendant le snapshot ;
- aucune modification de l'état interne des blocs.

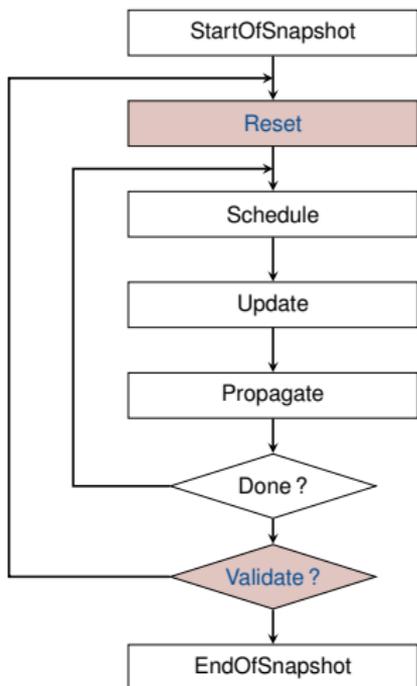


Calcul de point fixe par itération :

- observation séquentielle des blocs ;
- mise à jour de leur interface ;
- propagation des informations selon les relations entre pins.

Schedule, **Propagate** et **Done** sont les trois opérations qui définissent un MoC.

Update représente le comportement observable des blocs.

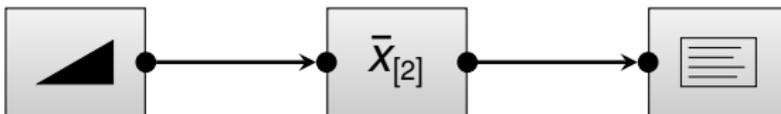


Possibilité d'invalider le point fixe atteint.

Recherche un autre point fixe à partir d'autres conditions initiales.

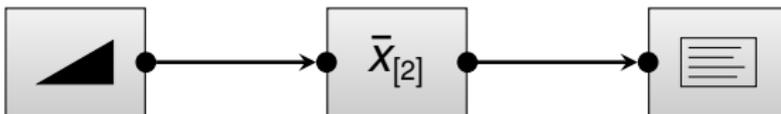
Exemple d'un MoC à flots de données synchrones

- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Exemple d'un MoC à flots de données synchrones

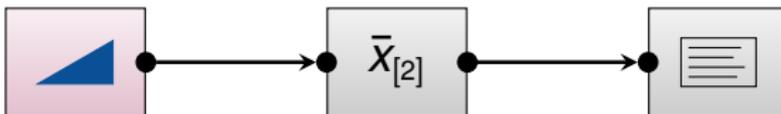
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



StartOfSnapshot

Exemple d'un MoC à flots de données synchrones

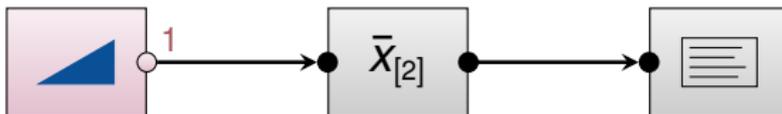
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Schedule → rampe

Exemple d'un MoC à flots de données synchrones

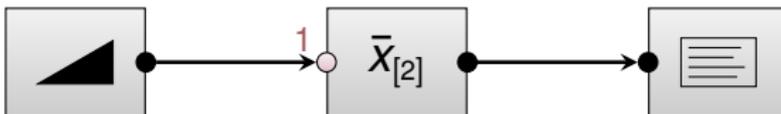
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Update de la rampe

Exemple d'un MoC à flots de données synchrones

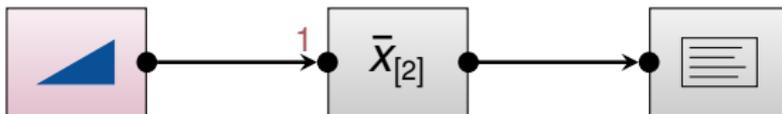
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Propagate

Exemple d'un MoC à flots de données synchrones

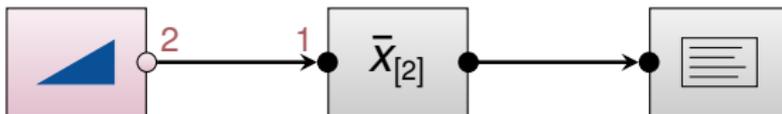
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Schedule \rightarrow rampe

Exemple d'un MoC à flots de données synchrones

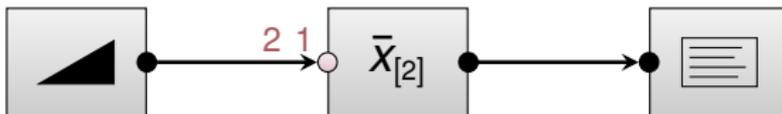
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Update de la rampe

Exemple d'un MoC à flots de données synchrones

- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Propagate

Exemple d'un MoC à flots de données synchrones

- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Schedule → moyenne

Exemple d'un MoC à flots de données synchrones

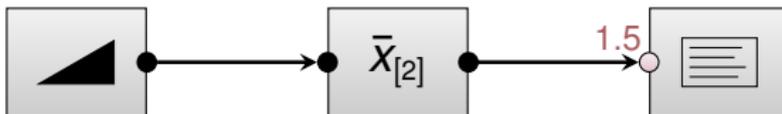
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Update de la moyenne

Exemple d'un MoC à flots de données synchrones

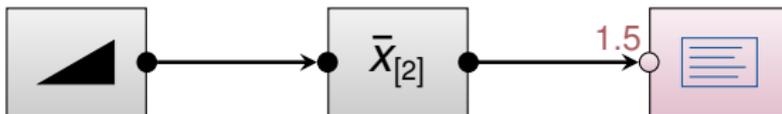
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Propagate

Exemple d'un MoC à flots de données synchrones

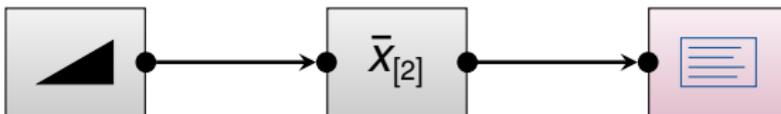
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Schedule → afficheur

Exemple d'un MoC à flots de données synchrones

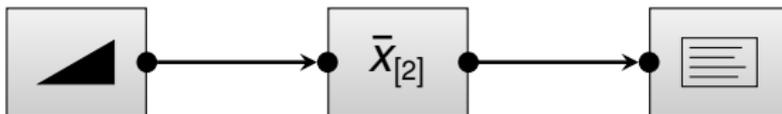
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Update de l'afficheur

Exemple d'un MoC à flots de données synchrones

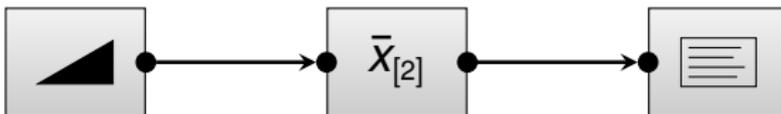
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Done

Exemple d'un MoC à flots de données synchrones

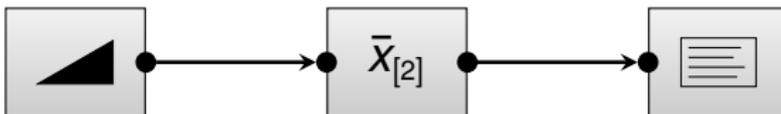
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



Validate

Exemple d'un MoC à flots de données synchrones

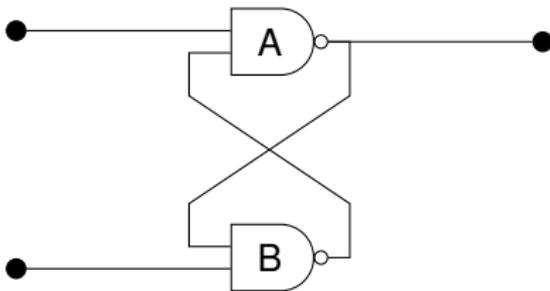
- Propagation instantanée des données
- Consommation et production d'un nombre fixe de données
- Snapshot = cycle minimal de production consommation



EndOfSnapshot

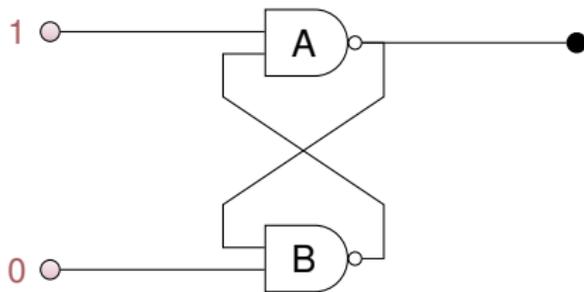
Exemple d'un MoC réactif synchrone

- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Exemple d'un MoC réactif synchrone

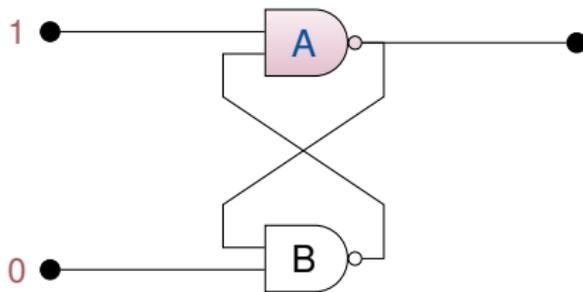
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



StartOfSnapshot

Exemple d'un MoC réactif synchrone

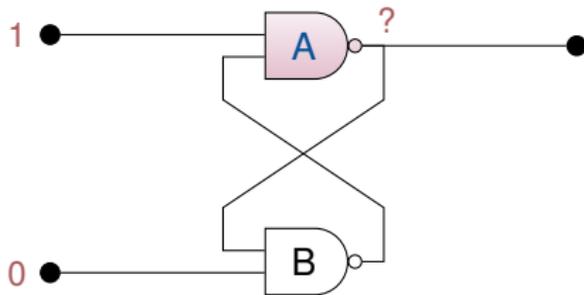
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Schedule \rightarrow A

Exemple d'un MoC réactif synchrone

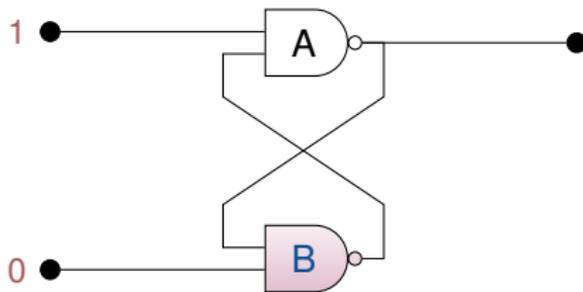
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Update A

Exemple d'un MoC réactif synchrone

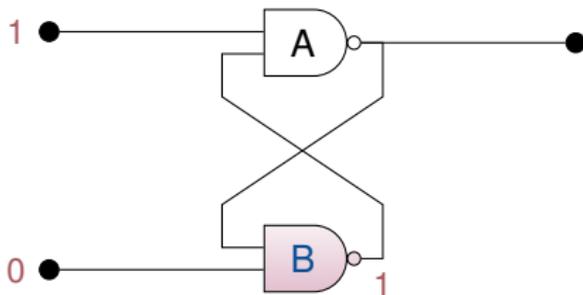
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Schedule \rightarrow B

Exemple d'un MoC réactif synchrone

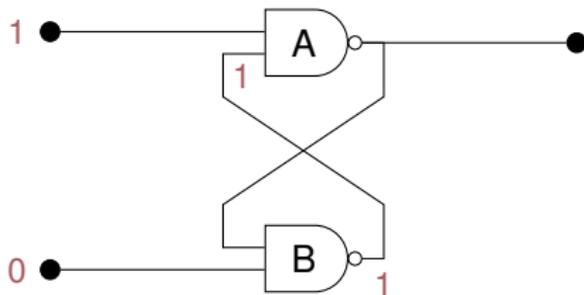
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Update B

Exemple d'un MoC réactif synchrone

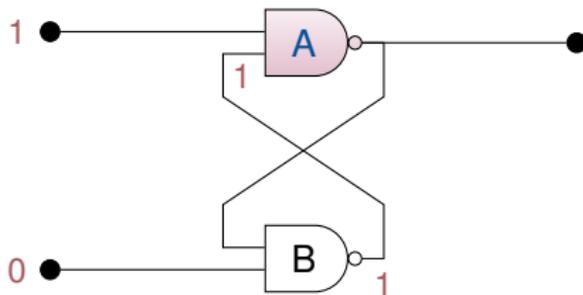
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Propagate

Exemple d'un MoC réactif synchrone

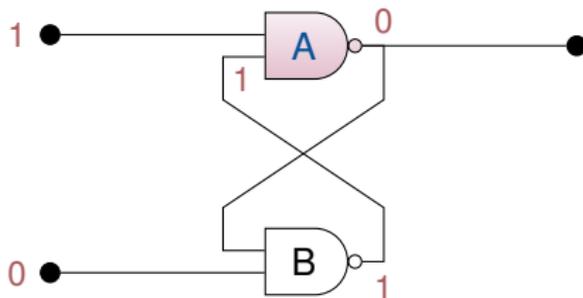
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Schedule \rightarrow A

Exemple d'un MoC réactif synchrone

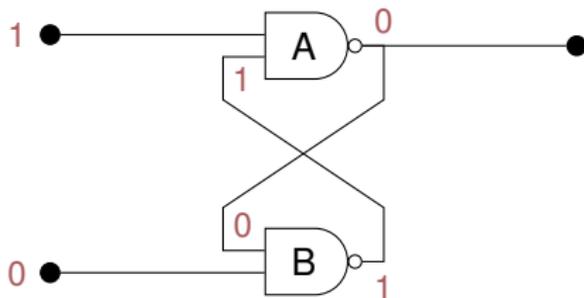
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Update A

Exemple d'un MoC réactif synchrone

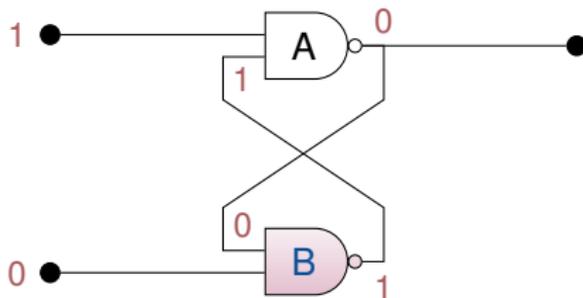
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Propagate

Exemple d'un MoC réactif synchrone

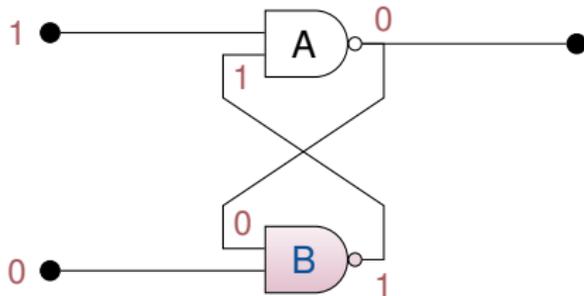
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Schedule \rightarrow B

Exemple d'un MoC réactif synchrone

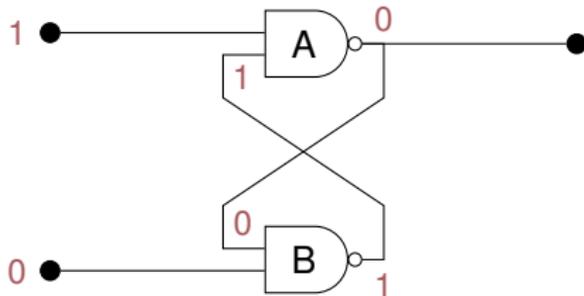
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Update B

Exemple d'un MoC réactif synchrone

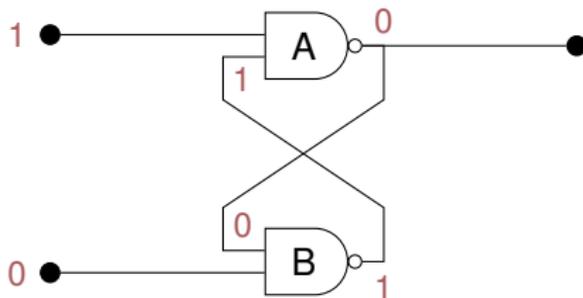
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Done

Exemple d'un MoC réactif synchrone

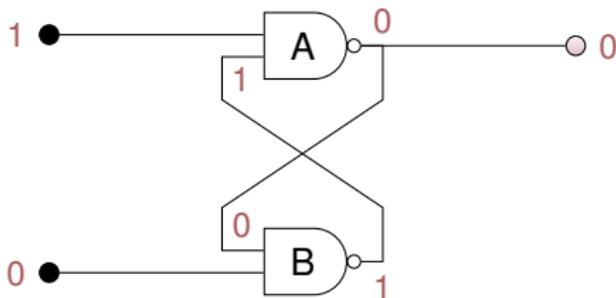
- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



Validate

Exemple d'un MoC réactif synchrone

- Propagation instantanée des données
- Blocs non stricts (réaction à des entrées partielles)
- Snapshot = tous les signaux sont déterminés



EndOfSnapshot

Modélisation hétérogène \Rightarrow description des différents MoC

Interactions entre des modèles utilisant des modèles de calcul **différents** ?

Comment combiner

- Automates
- Schémas blocs
- Réseaux de processus
- Systèmes discrets
- Systèmes continus

Modélisation hétérogène \Rightarrow description des différents MoC

Interactions entre des modèles utilisant des modèles de calcul **différents** ?

Comment combiner

- Automates
- Schémas blocs
- Réseaux de processus
- Systèmes discrets
- Systèmes continus

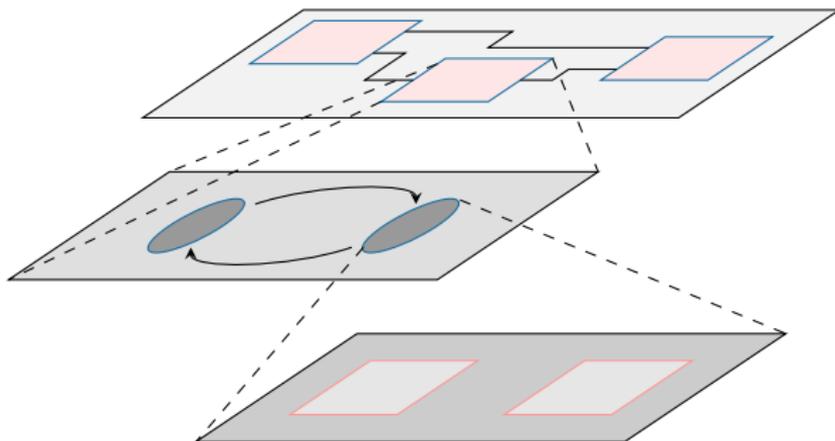


Deux types de chimères

- Composition à plat des modèles de calcul
- Composition hiérarchique

Composition hiérarchique

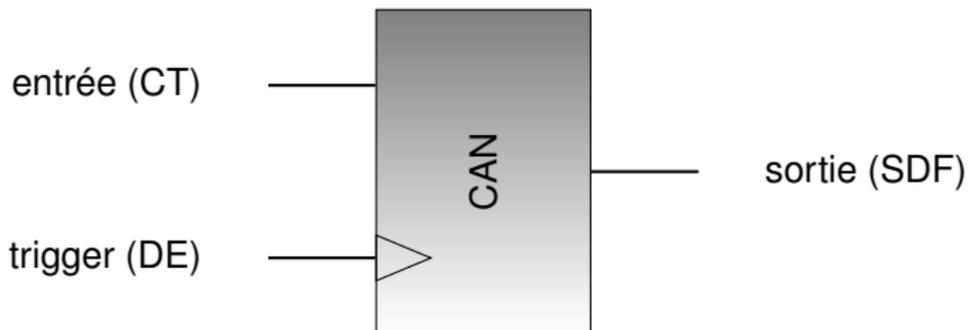
- Approche utilisée dans Ptolemy et ModHel'X
- Chaque niveau hiérarchique ne fait intervenir qu'un seul MoC
- Permet de ne considérer que les combinaisons de MoC 2 à 2



Adaptation sémantique hiérarchique modulaire (thèse de M. Feredj)

Composition à plat

- Plusieurs paradigmes peuvent intervenir à un niveau hiérarchique
- Approches à connecteurs (limitent les combinaisons)
- Composants à interface hétérogène



Modèle plat \rightsquigarrow modèle hiérarchique (thèse de M. Mbobi)

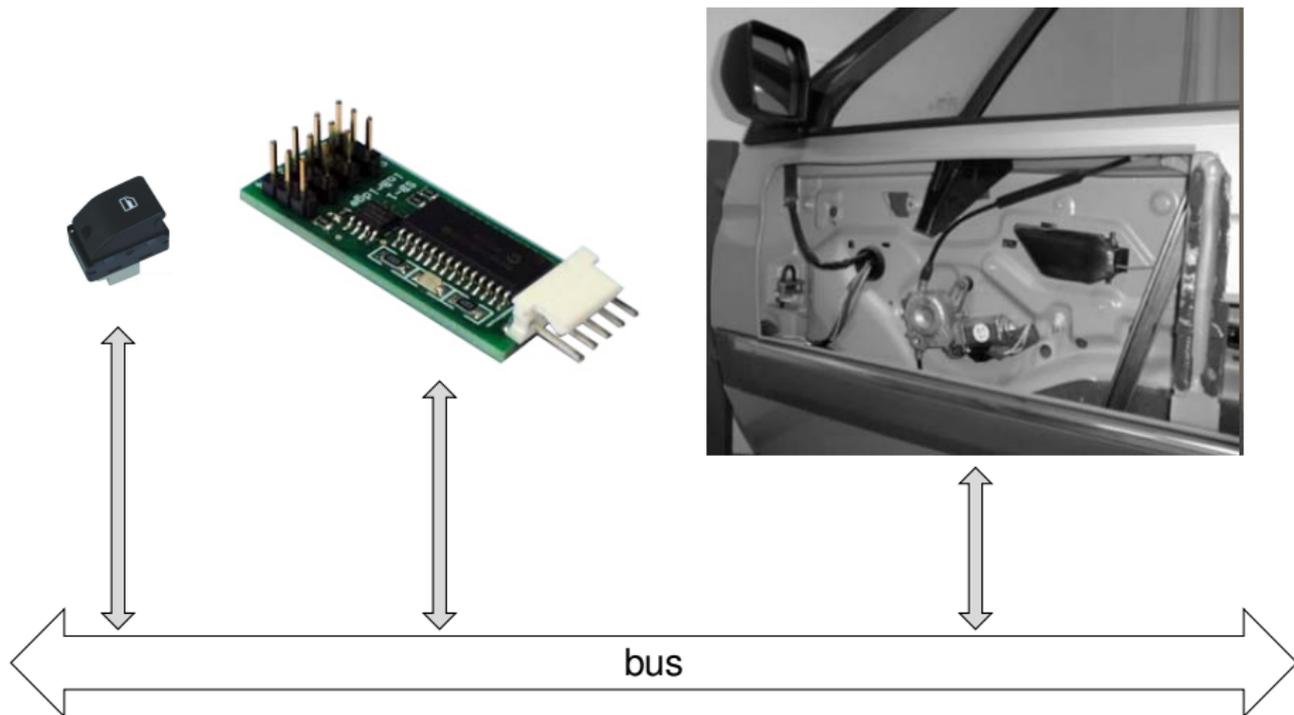
ModHel'X : adaptation sémantique caractérisée selon trois axes

- Adaptation des **données** :
différents MoC peuvent utiliser différentes formes de données
- Adaptation du **temps** :
les notions de temps diffèrent d'un MoC à l'autre
- Adaptation du **contrôle** :
les instants auxquels un modèle doit être observé dépendent des MoC

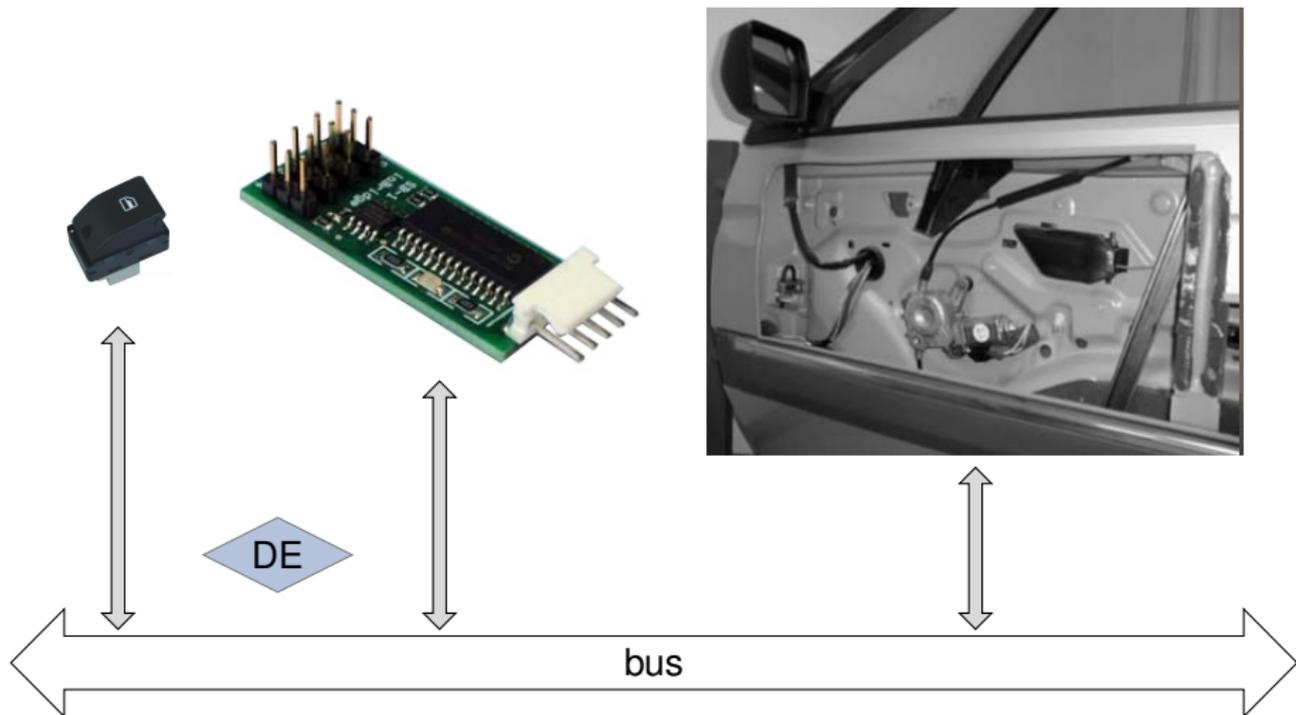
Projet EDONA

- Formalisation de MoC du domaine automobile
- Définition de patrons d'adaptation sémantique paramétrables
- Exemple du lève-vitre électrique

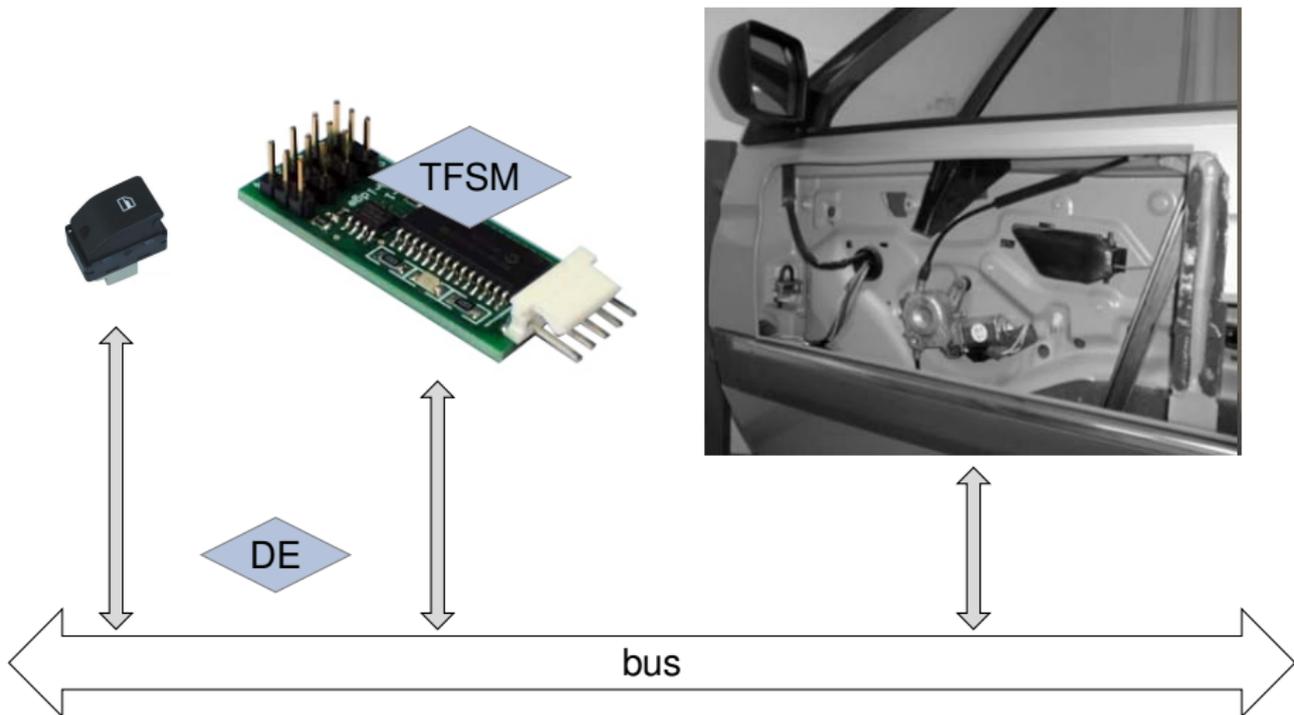
Exemple : lève-vitre



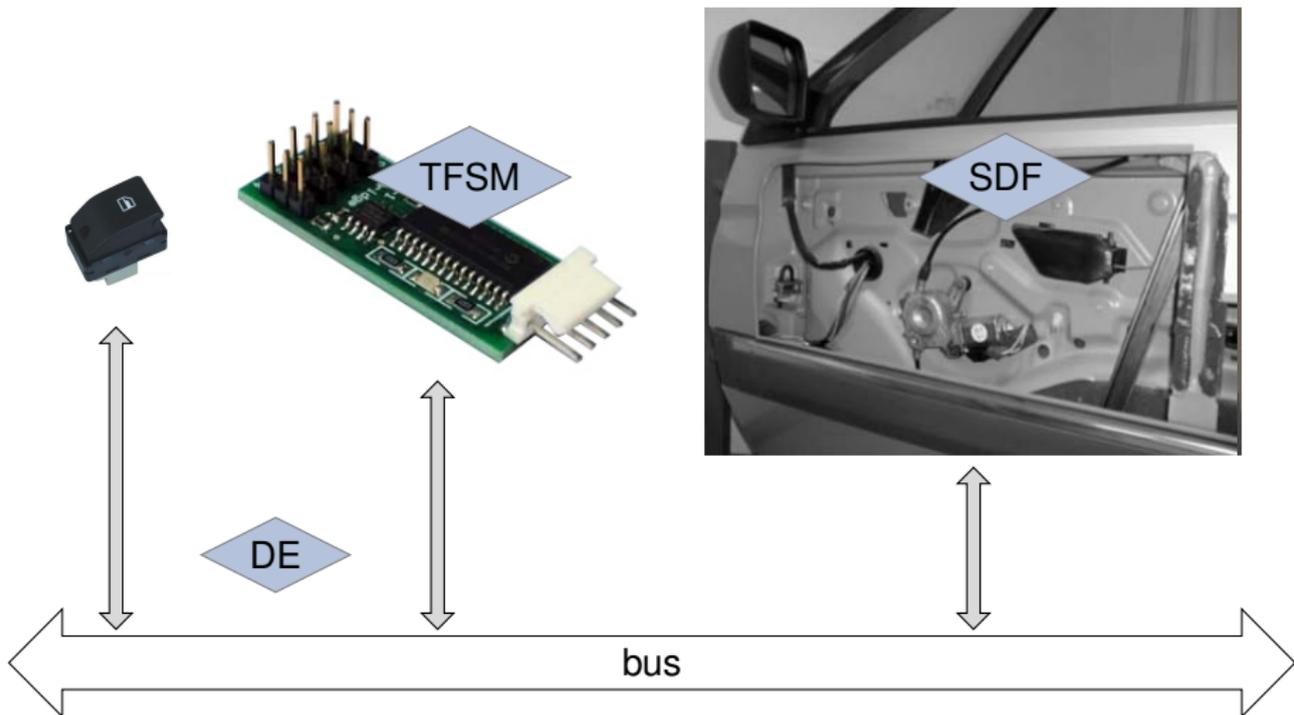
Exemple : lève-vitre



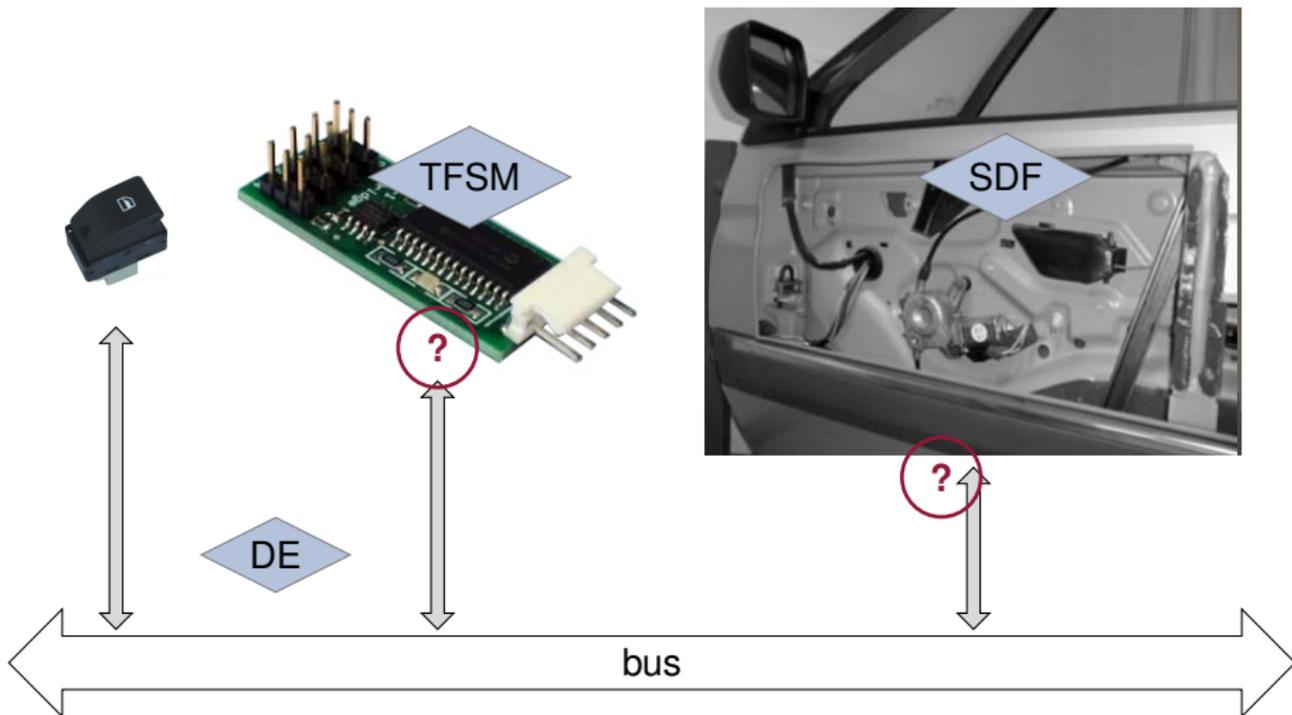
Exemple : lève-vitre



Exemple : lève-vitre



Exemple : lève-vitre



Adaptation des données entre DE et TFSM

valeur
date

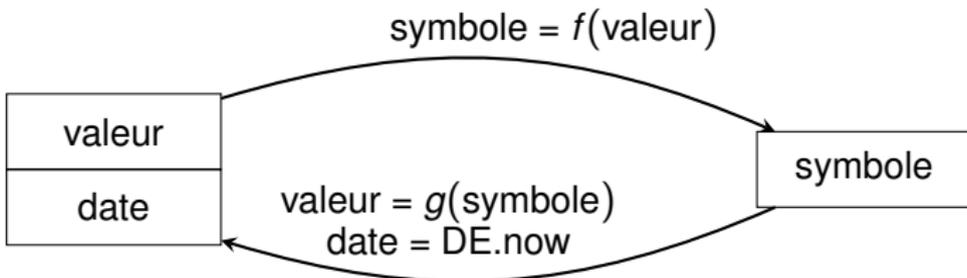
symbole

Adaptation des données entre DE et TFSM

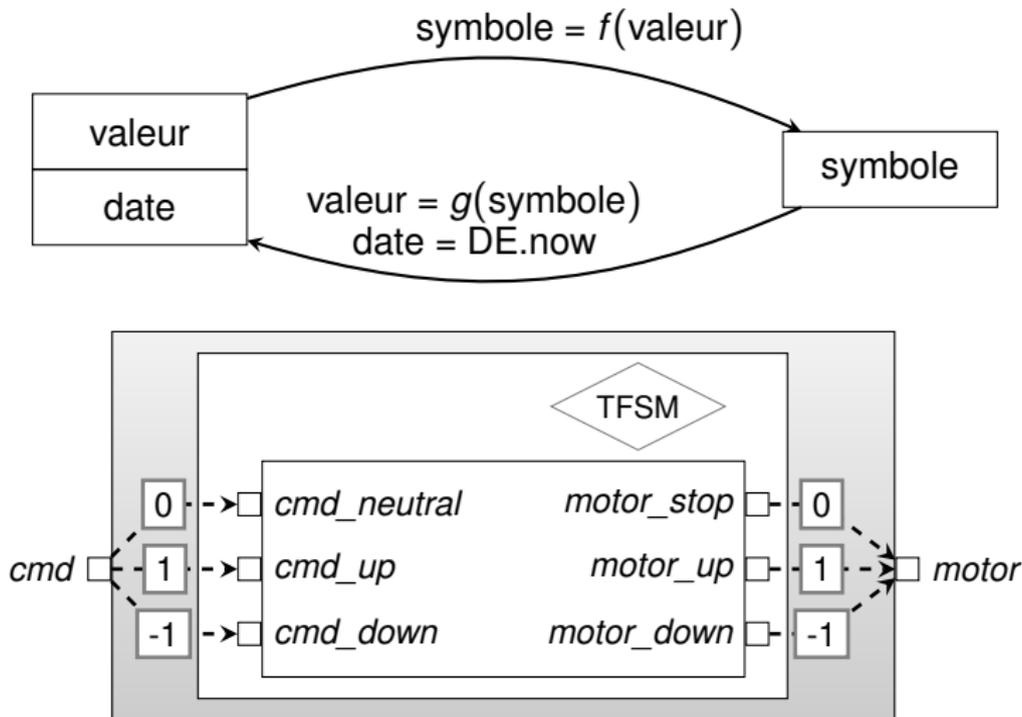
symbole = $f(\text{valeur})$



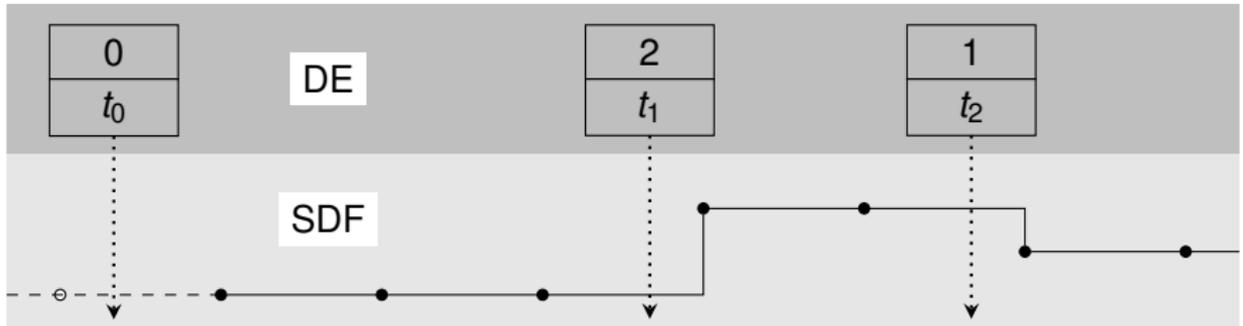
Adaptation des données entre DE et TFMS



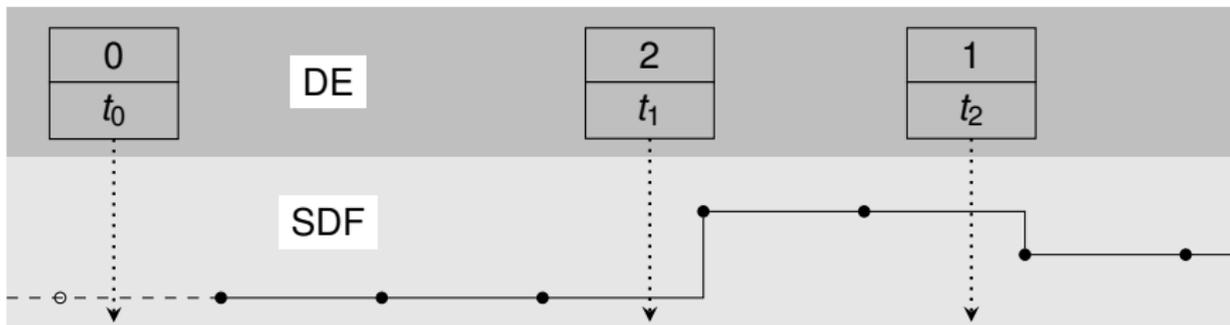
Adaptation des données entre DE et TFMSM



Adaptation des données entre DE et SDF

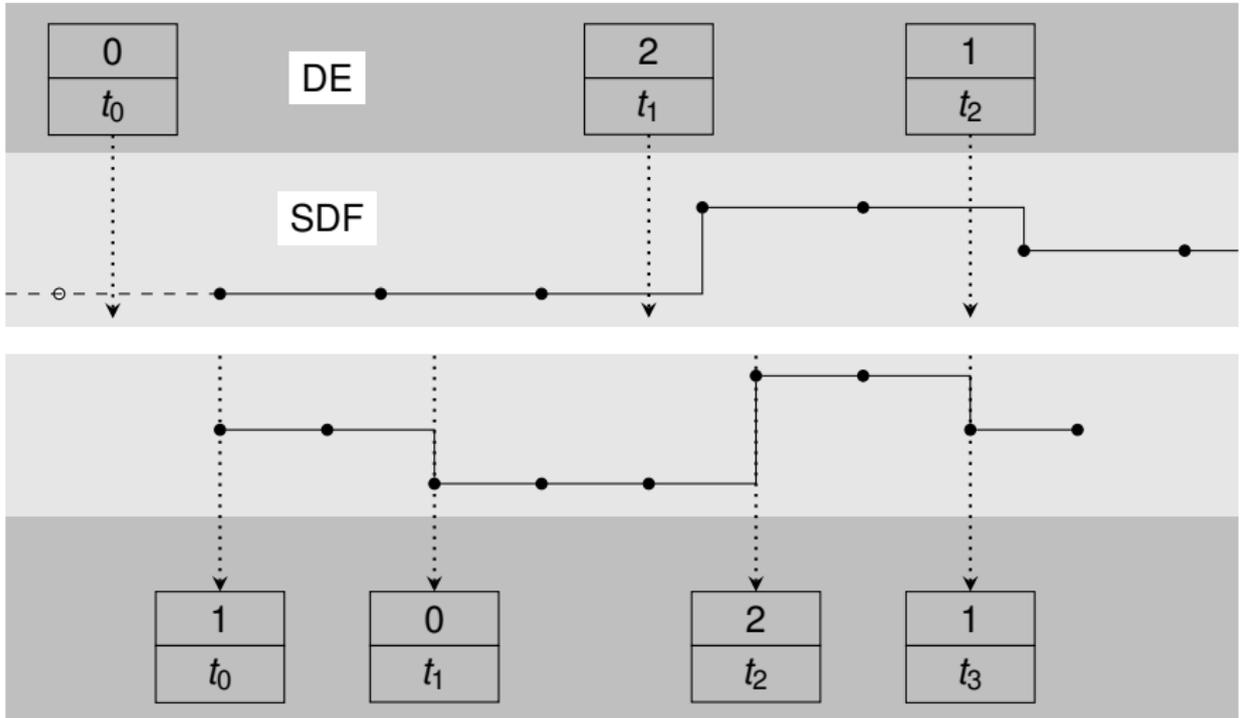


Adaptation des données entre DE et SDF



- Position relative événements / échantillons \Rightarrow adaptation du temps
- Occurrence des échantillons \Rightarrow adaptation du contrôle

Adaptation des données entre DE et SDF



Le temps dans les modèles de calcul

Formes du temps

- Étiquettes sur les données (date)
- Durées (chronométrage)

Rôles du temps

- Ordre partiel sur les données (causalité)
- Durées, échéances (contrôle)

Adaptation du temps

Le temps de DE

- Étiquettes temporelles sur $\mathbb{R} \times \mathbb{N}$
- Synchronisation, causalité
- Contrôle du traitement des événements

Le temps de TFMSM

- Mesure du temps écoulé dans un état
- Pas de sens donné à la valeur absolue d'une date

Adaptation DE – TFMSM

- Correspondance entre durée TFMSM et différence entre dates DE
- Conséquences sur le contrôle

Adaptation du temps

Le temps de DE

- Étiquettes temporelles sur $\mathbb{R} \times \mathbb{N}$
- Synchronisation, causalité
- Contrôle du traitement des événements

Le temps de SDF

- Séquence d'échantillons, pas de véritable notion de date
- Pas de notion de durée

Adaptation DE – SDF

- Affectation d'une date DE aux échantillons SDF
- Période d'échantillonnage, conséquences sur le contrôle

Adaptation du contrôle

TFSM 

DE 

SDF 

Adaptation du contrôle

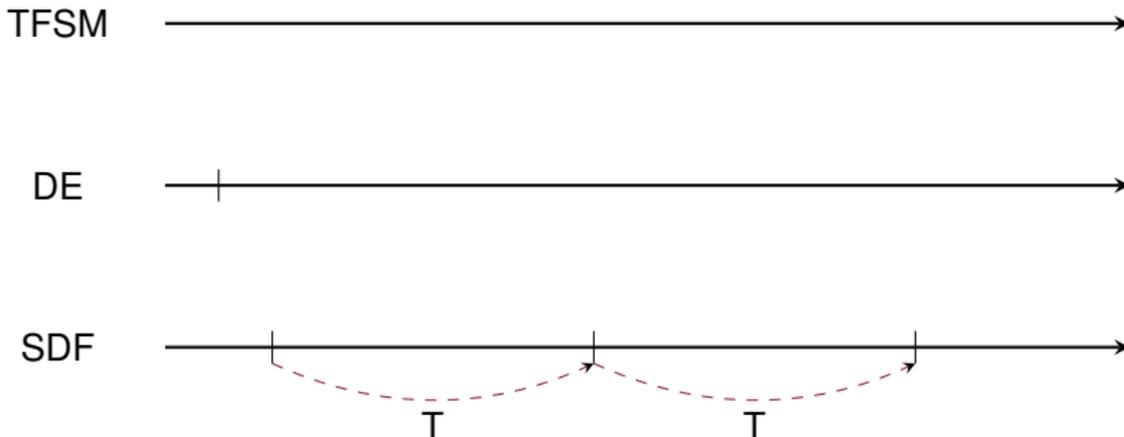
TFSM 

DE 

SDF 

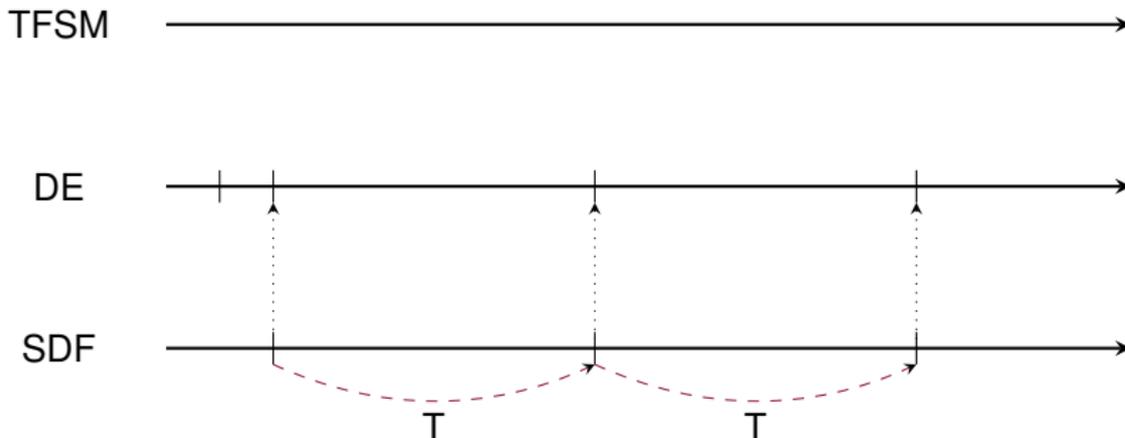
Événement dans DE, pas d'entrée pour l'automate

Adaptation du contrôle



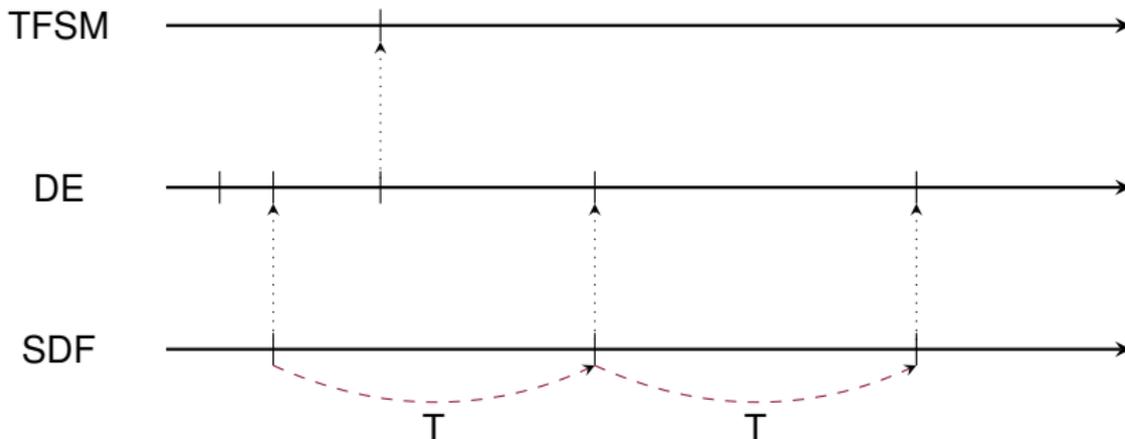
Contrôle périodique de SDF \Rightarrow contrôle dans DE

Adaptation du contrôle



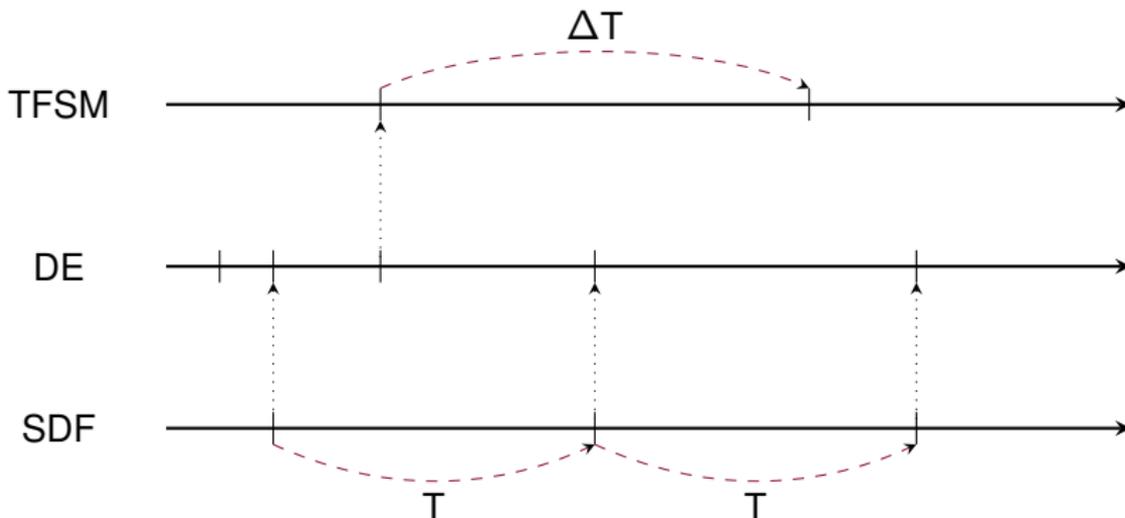
Contrôle périodique de SDF \Rightarrow contrôle dans DE

Adaptation du contrôle



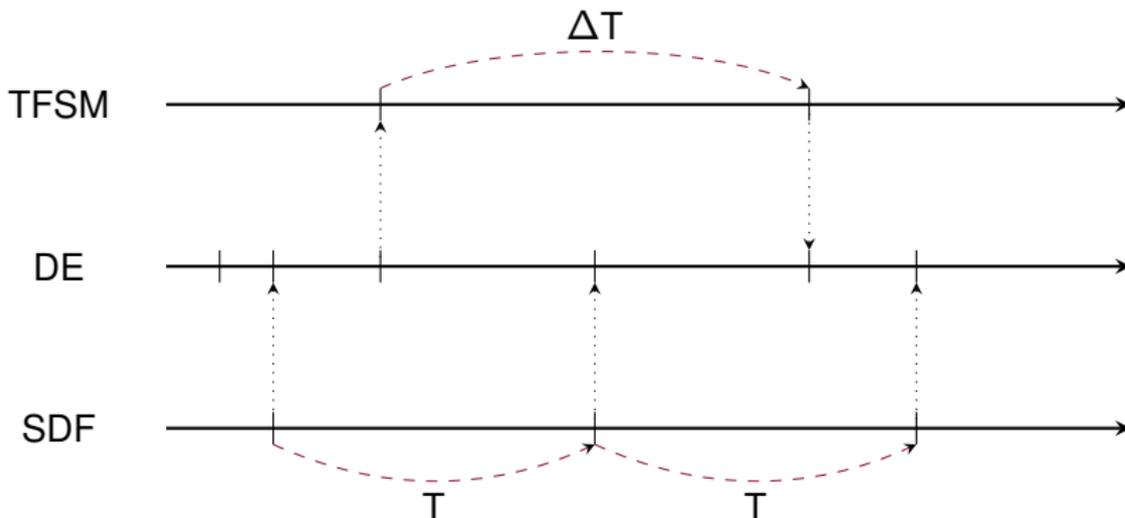
Événement dans DE, entrée pour l'automate
⇒ contrôle dans TFSM

Adaptation du contrôle



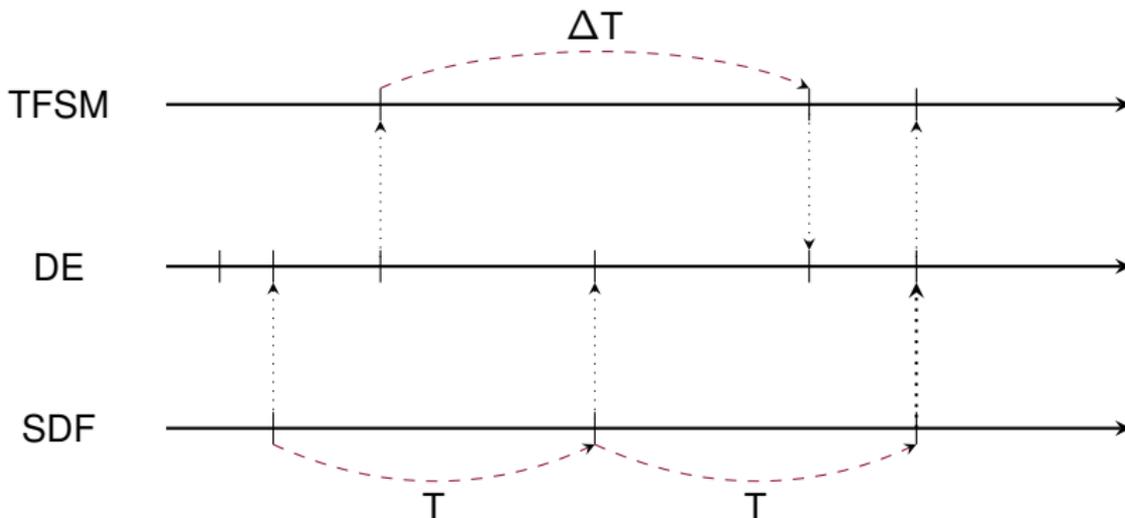
Expiration d'une transition temporisée de l'automate
 \Rightarrow contrôle dans DE

Adaptation du contrôle



Expiration d'une transition temporisée de l'automate
 \Rightarrow contrôle dans DE

Adaptation du contrôle

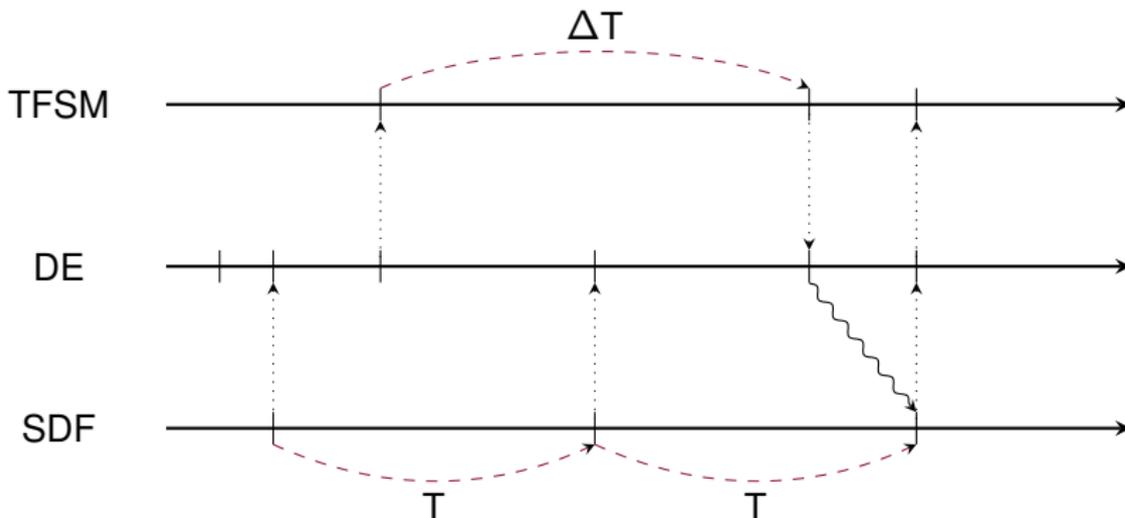


Observation périodique de SDF

⇒ contrôle dans DE + événement pour l'automate

⇒ contrôle dans TFSM

Adaptation du contrôle



Les données ne créent jamais de contrôle pour SDF
 \Rightarrow mémorisation des données à l'interface

Actuellement

- Adaptation des données : patrons d'adaptation paramétrables
- Adaptation du temps : gestion ad hoc dans les blocs d'interface
- Adaptation du contrôle : contraintes sur la date du prochain snapshot

Évolution (thèse d'Ayman Dogui)

- Adaptation des données : MoC particulier ?
- Adaptation du temps et du contrôle : calcul d'horloge (Clock Constraint Specification Language)

Modélisation explicite de l'adaptation sémantique (Ayman Dogui)

- Modèles pour l'adaptation des données
- Calcul d'horloges pour l'adaptation du temps et du contrôle

Application à l'exécution de modèles UML (Abderraouf Benyahia)

- Exécution de modèles UML
- Prise en compte des profils du domaine temps-réel embarqué

Test de conformité des systèmes hétérogènes (Bilal Kanso)

- Modélisation coalgébrique des composants
- Généralisation d'un algorithme de génération de tests de conformité
- Définition de connecteurs

Dans ModHel'X

Modèles hétérogènes des aspects fonctionnels

⇒ **hétérogénéité des composants uniquement**

Trois autres axes d'hétérogénéité à explorer

- Hétérogénéité des vues : modélisation multi-vues
- Hétérogénéité des niveaux d'abstraction : conformité d'un modèle à un autre, conformité d'un modèle d'exécution à un modèle de calcul
- Hétérogénéité des activités : cohérence entre les modèles utilisés pour différentes activités (validation/génération de code)

Validation

- Model-checking hétérogène : comment combiner des model-checkers pour vérifier des propriétés sur un composant connaissant les propriétés valides sur son environnement
- Définition formelle des modèles d'exécution/de calcul
- Composition de propriétés exprimées selon des MoC différents, correction par construction.

- Mokhoo Mbobi, soutenue le 17 décembre 2004
Université Paris-Sud 11, Guy Vidal-Naquet
Modélisation hétérogène non hiérarchique
- Mohamed Feredj, soutenue le 2 décembre 2005
Université Paris-Sud 11, Guy Vidal-Naquet
Composants domaine-polymorphes
- Cécile Hardebolle, soutenue le 28 novembre 2008
Université Paris-Sud 11, Guy Vidal-Naquet
Formalisation des modèles d'exécution et de leurs interactions
- Bilal Kanso, débutée en 2008
École centrale Paris, Marc Aiguier
Théorie du test de conformité pour les modèles hétérogènes
- Abderraouf Benyahia, débutée en 2008
Supélec - CEA, François Terrier
Sémantiques d'exécution de modèles UML profilés pour le temps-réel
- Ayman Dogui, débutée en 2010
Supélec, Yolaine Bourda
Modélisation explicite de l'adaptation sémantique entre MoC

Approche synchrone et objets

Intégration de l'approche synchrone dans des programmes classiques

- Nécessité de fournir un environnement d'exécution compatible avec la sémantique synchrone
- Nécessité d'adapter cette sémantique à l'environnement logiciel

Domaines et acteurs

Modélisation par acteur, existence de différentes lois de combinaison

- Notion de modèle de calcul
- Notion d'adaptation sémantique entre modèles

Aujourd'hui

La modélisation multi-paradigme est un domaine actif

- Simulink Stateflow, COMSOL, VHDL-AMS, System-C AMS
- Ptolemy, ModHel'X, Rosetta

Approche fédératrice ouverte pour

- Décrire des modèles hétérogènes
- Décrire des modèles d'exécution
- Définir l'adaptation sémantique entre modèles d'exécution
- Exécuter des modèles hétérogènes

Points clefs

- Accepter l'hétérogénéité des modèles
- Utiliser conjointement les approches et outils existants
- En privilégiant la modularité

Merci de votre attention

► Communauté CAMPaM

► Projets

► Non validation

► Publications

► Non validation CT

► Modélisation multi-vues

► Temps super dense

- Cécile Hardebolle, Frédéric Boulanger. *Exploring Multi-Paradigm Modeling Techniques*. SIMULATION, November 2009. Impact factor : 0.404 (2007), Rank : B (CORE 2009).
- Cécile Hardebolle, Frédéric Boulanger. *Multi-Formalism Modelling and Model Execution*. International Journal of Computers and their Applications, July 2009. Rank : C (CORE 2009).
- Mohamed Feredj, Frédéric Boulanger, Aimé Mokho Mbohi. *A model of domain-polymorph component for heterogeneous system design*. Journal of Systems and Software, January 2009. Impact factor : 1.340 (2009), Rank : A (CORE 2010).
- Frédéric Boulanger, Christophe Jacquet, Cécile Hardebolle, Elyes Rouis. *Modeling Heterogeneous Points of View with ModHel'X*. Models in Software Engineering : Workshops at MoDELS 2009, LNCS 6002, 2010.
- Cécile Hardebolle, Frédéric Boulanger. *ModHel'X : A Component-Oriented Approach to Multi-Formalism Modeling*. Models in Software Engineering : Workshops at MoDELS 2007, LNCS 5002, 2008.
- Guy Vidal-Naquet, Frédéric Boulanger. *Integration of Synchronous Modules in an Object-Oriented Language*. Information Systems – Correctness and Reusability, August 1995.
- Bilal Kanso, Marc Aiguier, Frédéric Boulanger, Assia Touil. *Testing of Abstract Components*. Proceedings of ICTAC 2010, LNCS 6255, 2010.
- Abderraouf Benyahia, Arnaud Cuccuru, Safouan Taha, François Terrier, Frédéric Boulanger, Sébastien Gérard. *Extending the Standard Execution Model of UML for Real-Time Systems*, volume 329 of IFIP Advances in Information and Communication Technology, 2010.

2005-2007 Usine Logicielle (System@tic)

- Composant CP-xUML du sous-projet OpenDev Factory avec le CEA LIST
- Langage de description d'architecture d'applications à contrôle vérifiable dans le sous-projet MoDriVal

2007-2010 EDONA (System@tic, Num@tec Automotive)

- Description formelle des modèles de calcul utilisés dans l'informatique embarquée des automobiles (SDF, TFSM, DE, EAST-ADL)
- Conception de patrons d'adaptation sémantique entre ces modèles de calcul

2008-2011 LAMBDA (System@tic)

- Intégration des approches synchrones dans la modélisation SysML/UML/MARTE

2009-2012 TASCCC (ANR)

- Formalisation de propriétés de sécurité pour le test des systèmes hétérogènes
- Analyse des relations entre les propriétés de sécurité et la modélisation comportementale pour la génération de tests

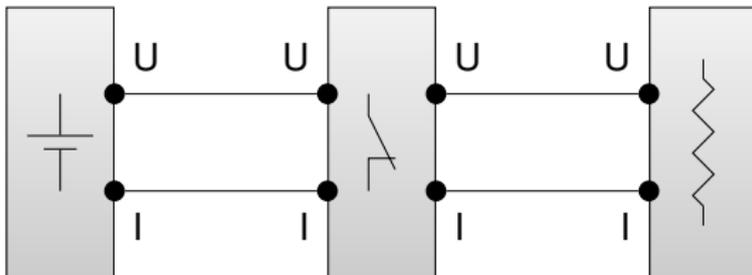
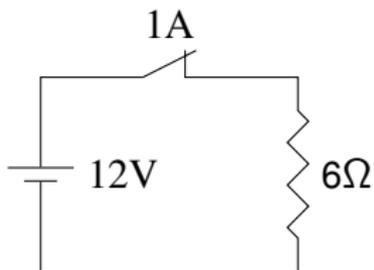
Computer Assisted Multi Paradigm Modeling

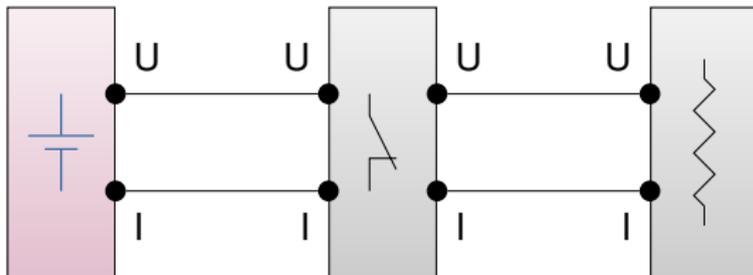
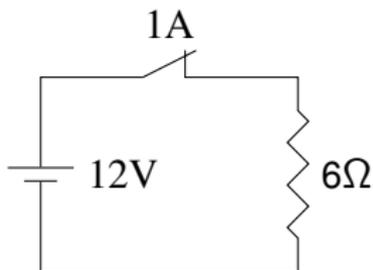
- Supélec : ModHel'X
- CEA LIST : LEM
- THeSys : Supélec, CEA, Centrale
- UC Berkeley : Ptolemy (Edward A. Lee)
- Mc Gill University, Montréal (Hans Vangheluwe)
- The Mathworks (Pieter J. Mosterman)
- Vanderbilt University, Nashville (Gabor Karsai)
- Verimag : BIP (Joseph Sifakis)
- University of Kansas : Rosetta (Perry Alexander)

Manifestations

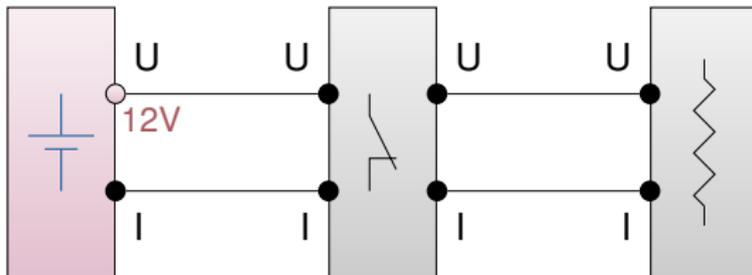
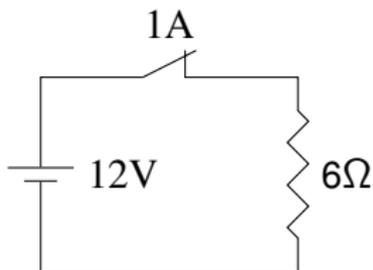
- Workshops MPM à MODELS, N° spécial de SIMULATION
- Workshop MoVaH à ICST
- Journée Hétérogénéo (RTRA Digiteo)



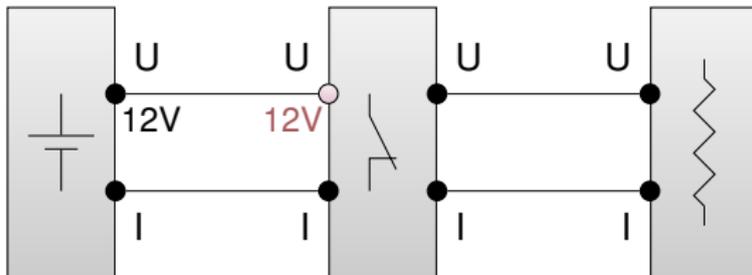
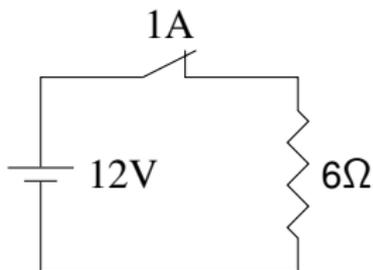




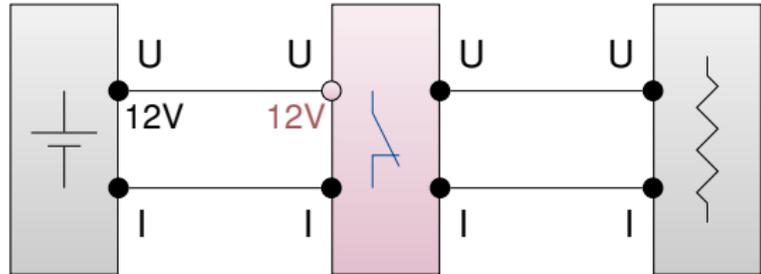
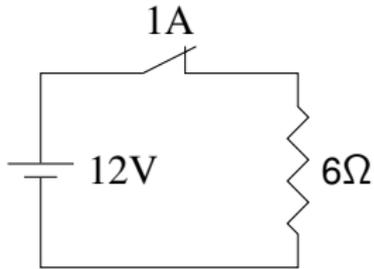
Schedule → batterie



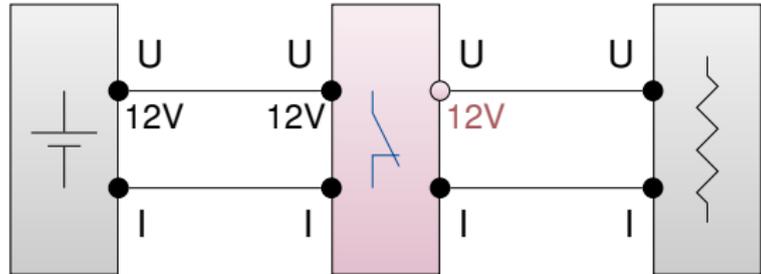
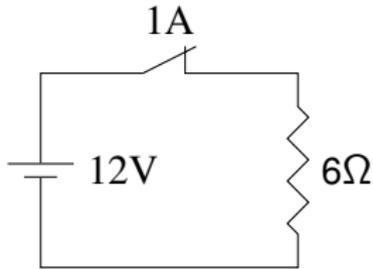
Update batterie



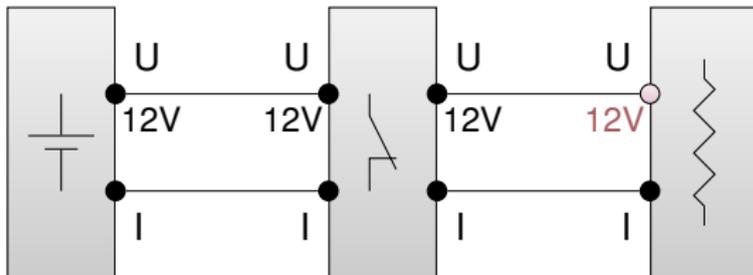
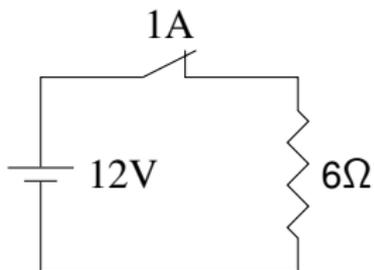
Propagate



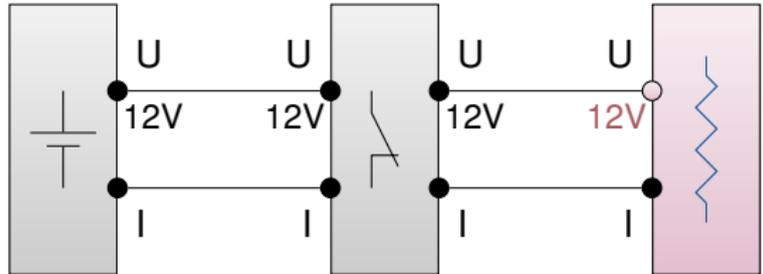
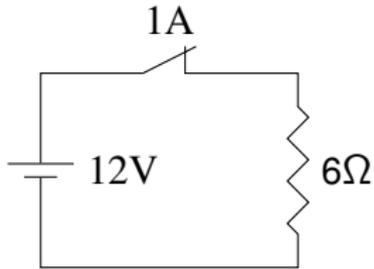
Schedule → fusible



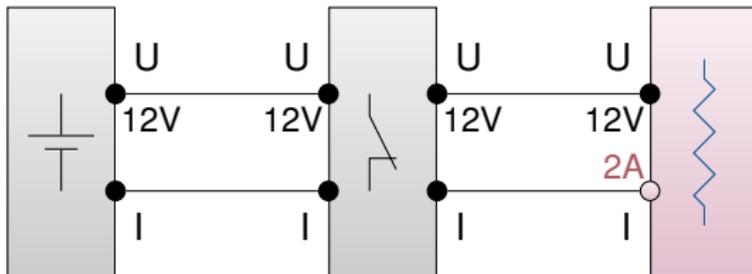
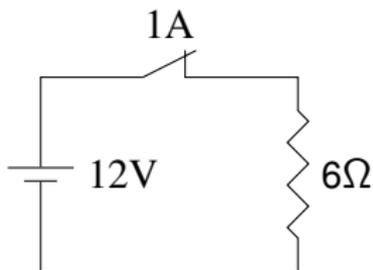
Update fusible



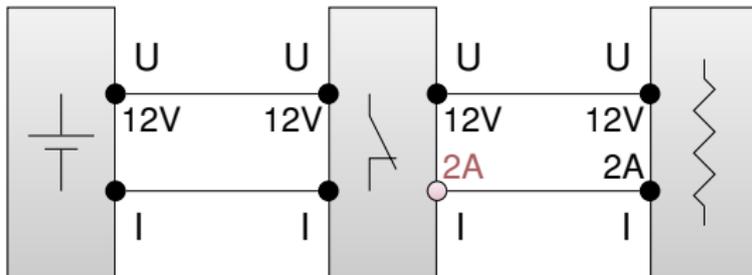
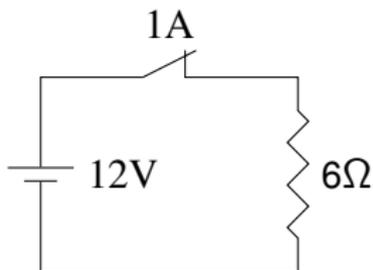
Propagate



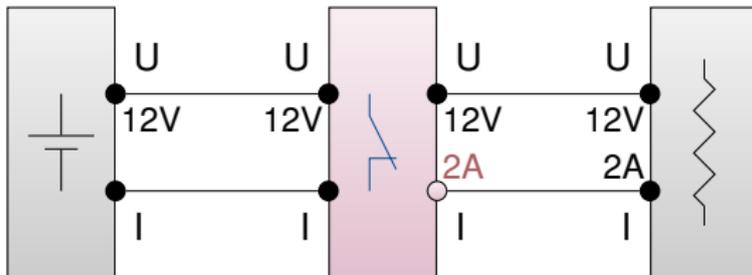
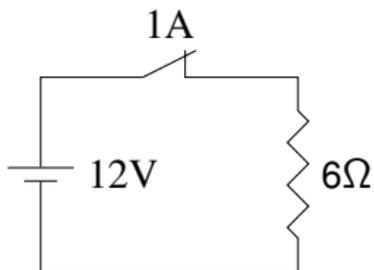
Schedule → résistance



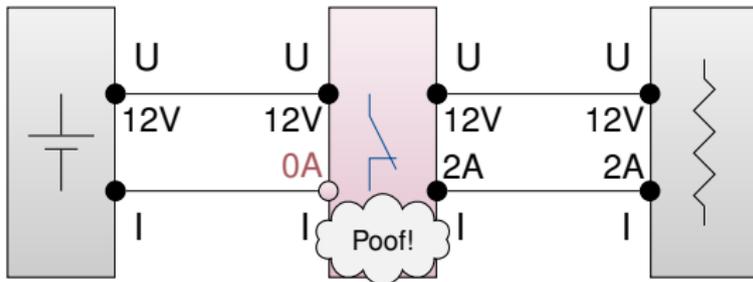
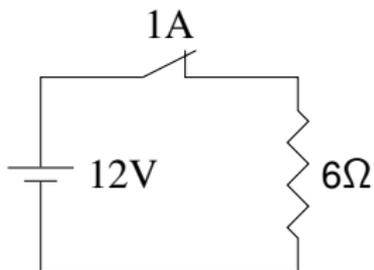
Update résistance



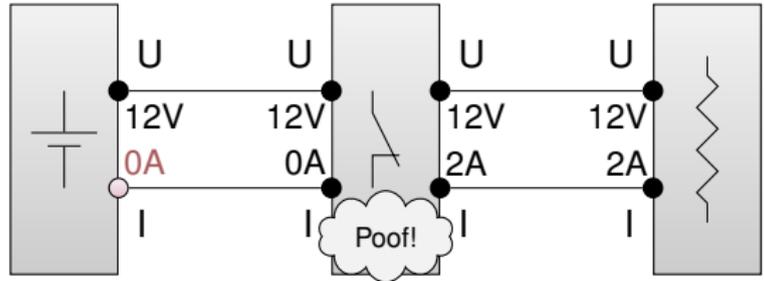
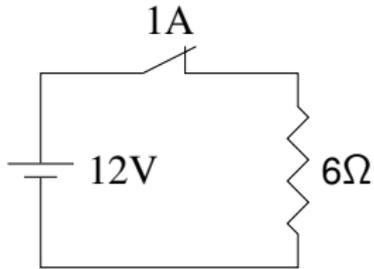
Propagate



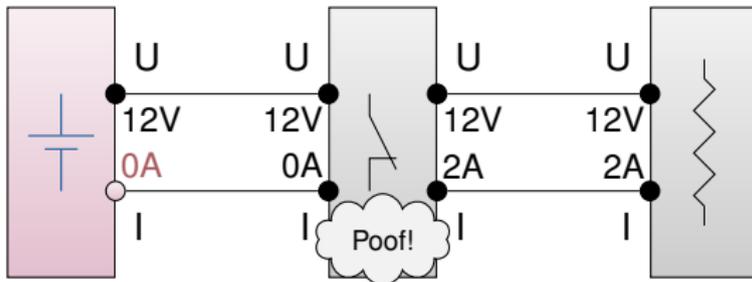
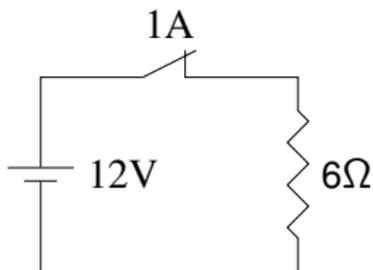
Schedule → fusible



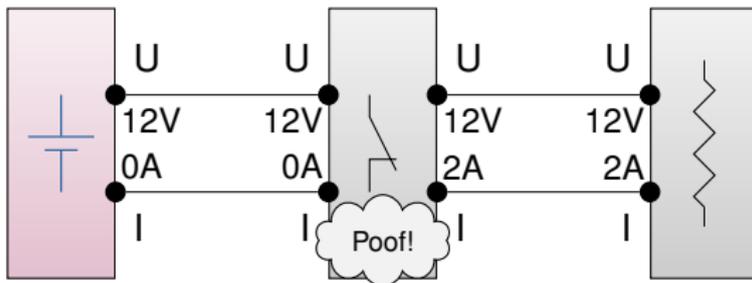
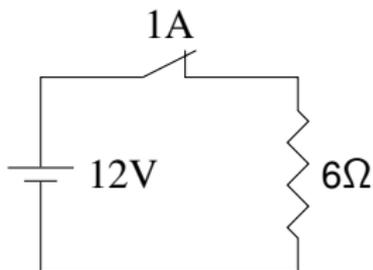
Update fusible



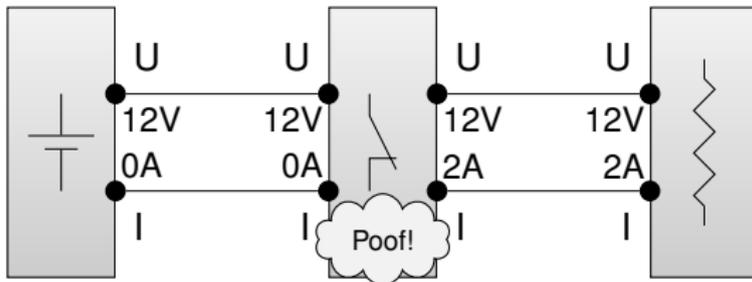
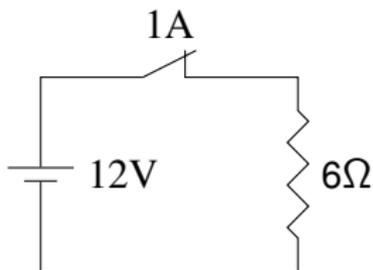
Propagate



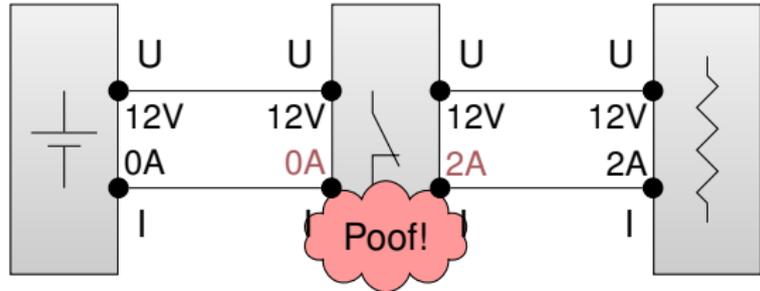
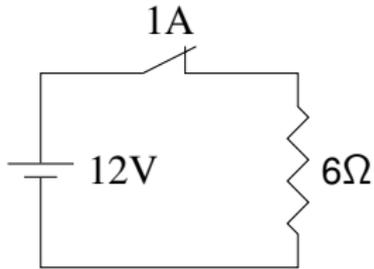
Schedule → batterie



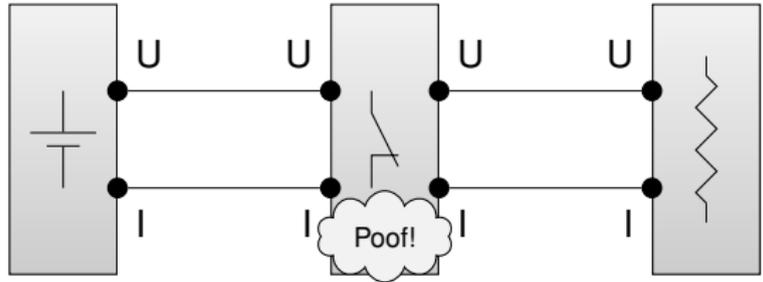
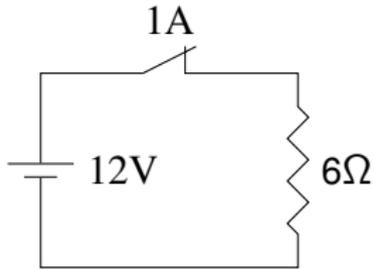
Update batterie



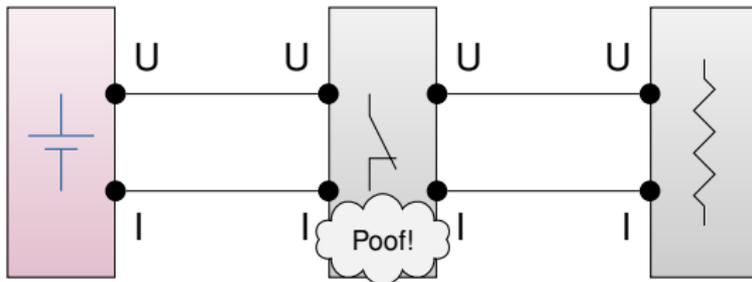
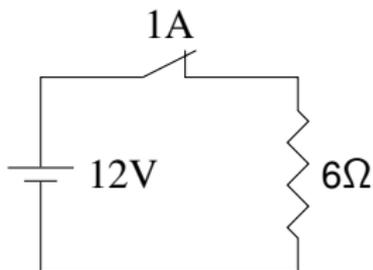
Done



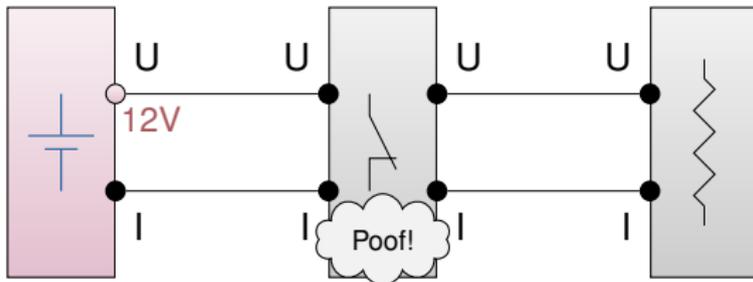
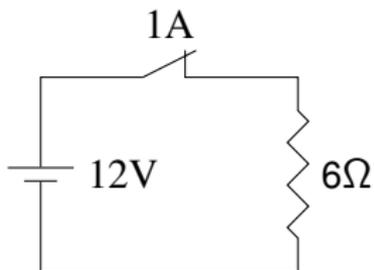
Validate



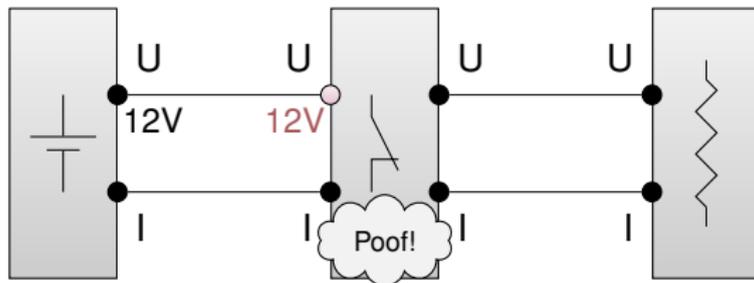
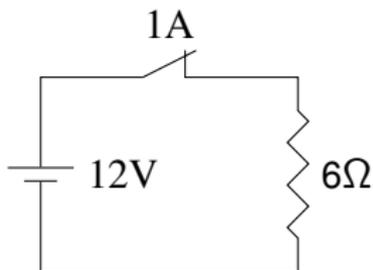
Reset



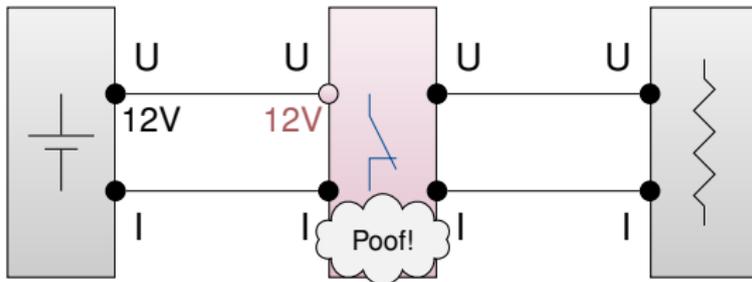
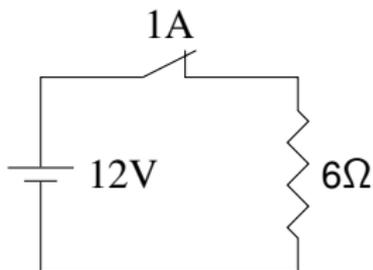
Schedule → batterie



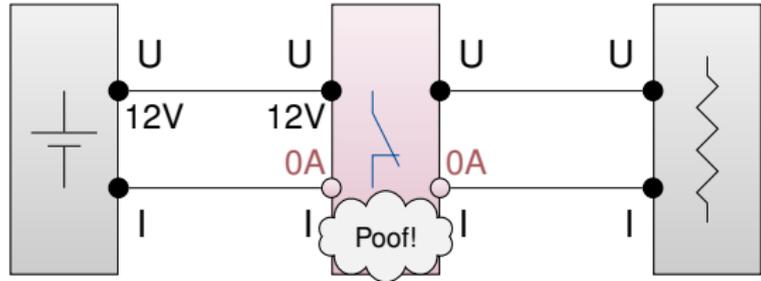
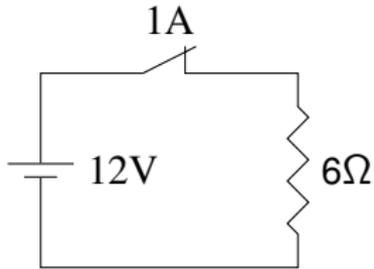
Update batterie



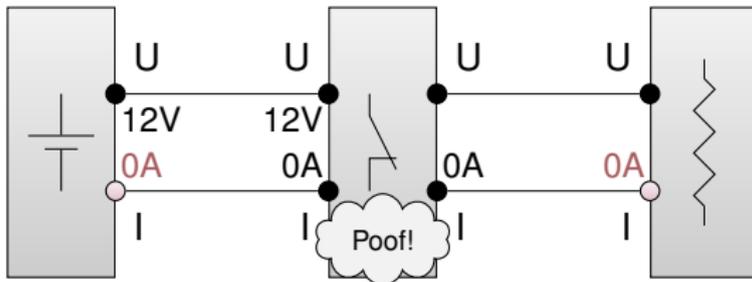
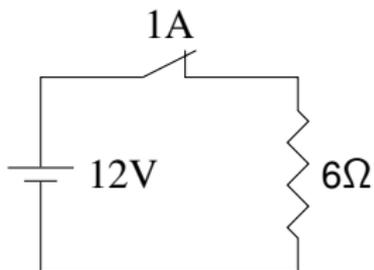
Propagate



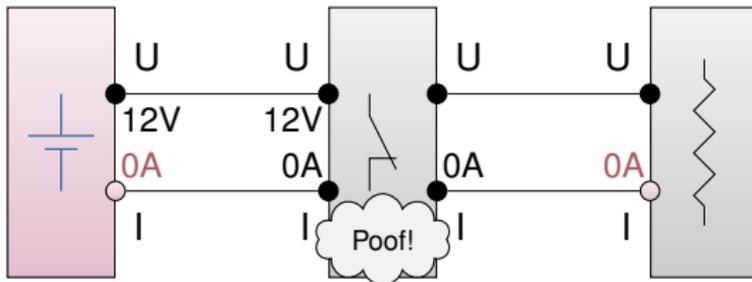
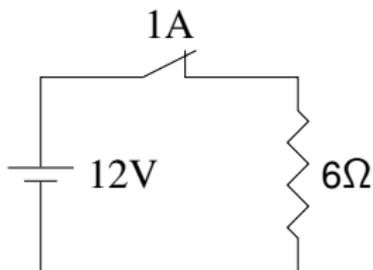
Schedule → fusible



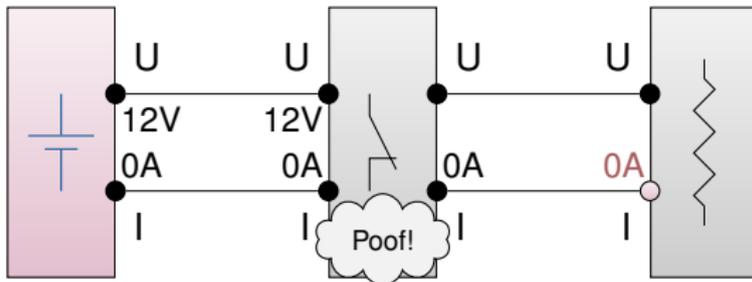
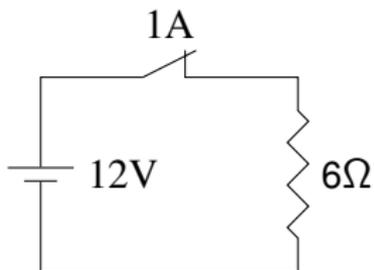
Update fusible



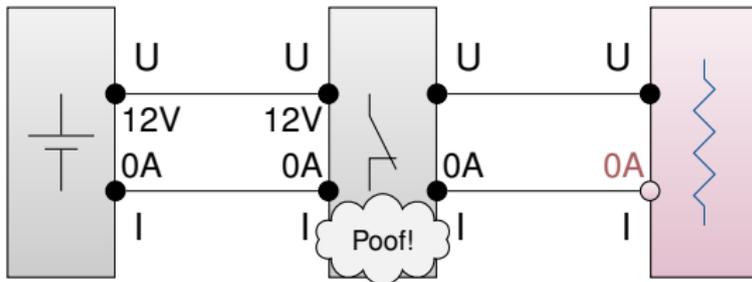
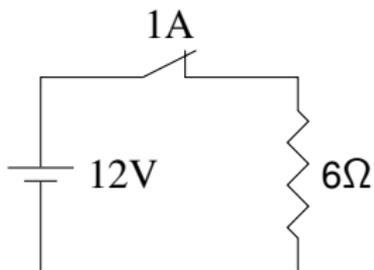
Propagate



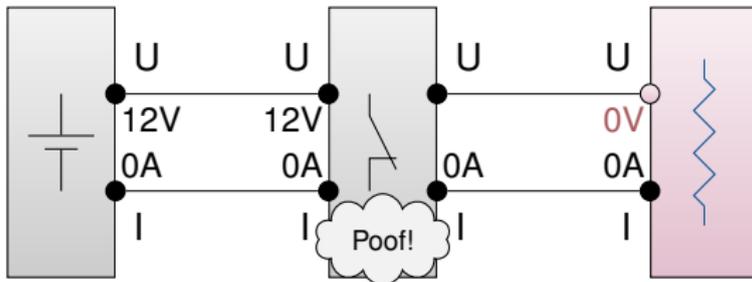
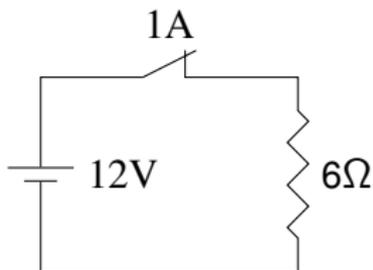
Schedule → batterie



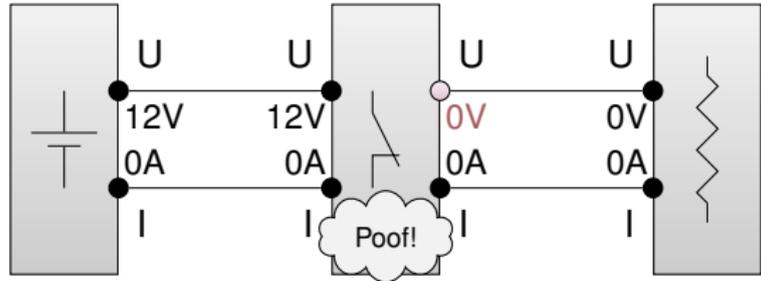
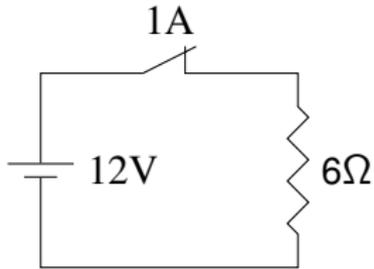
Update batterie



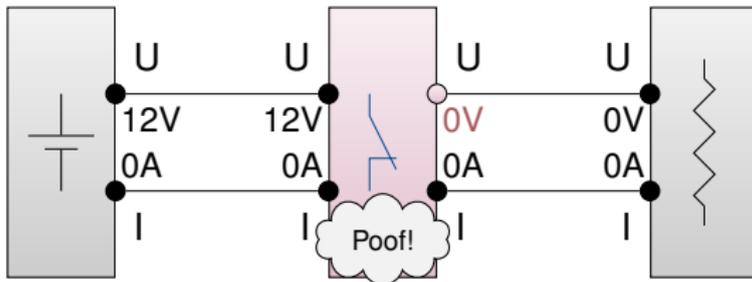
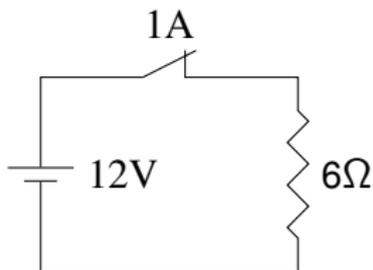
Schedule → résistance



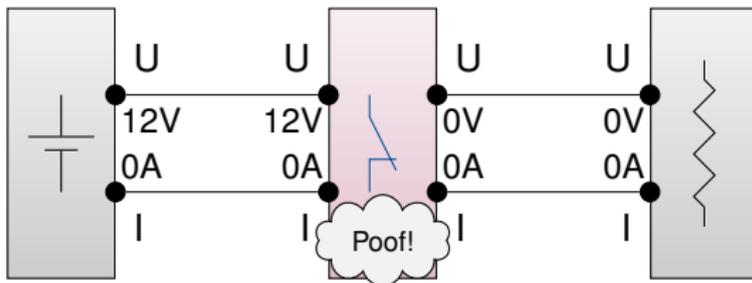
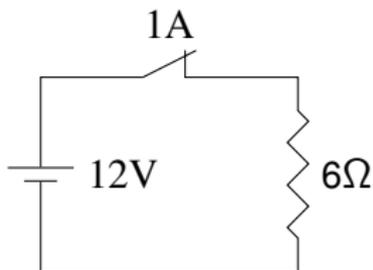
Update résistance



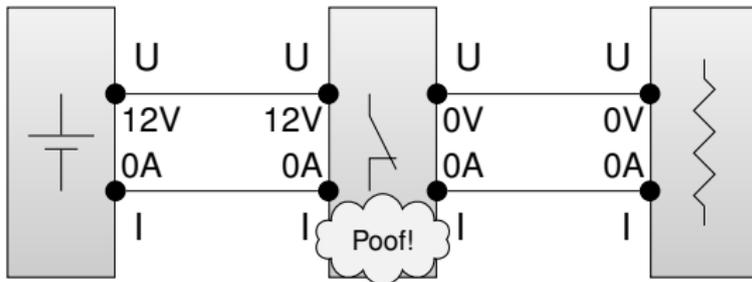
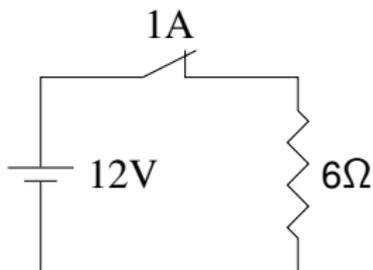
Propagate



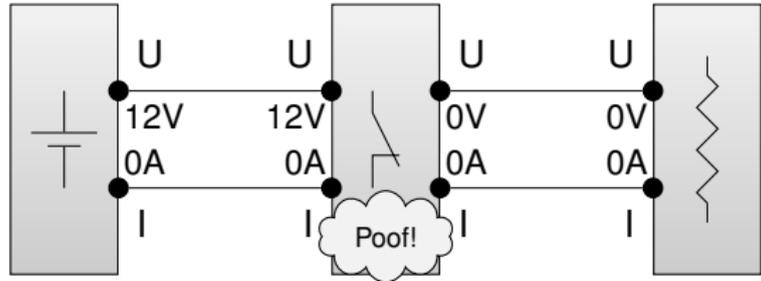
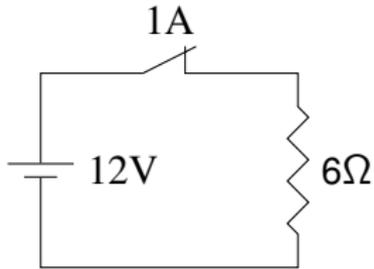
Schedule → fusible



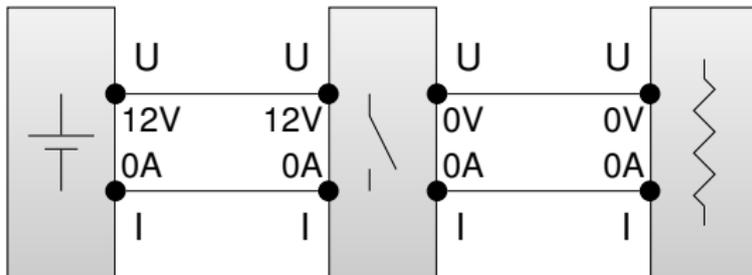
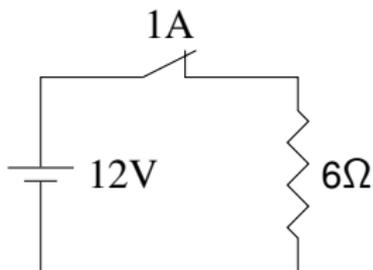
Update fusible



Done



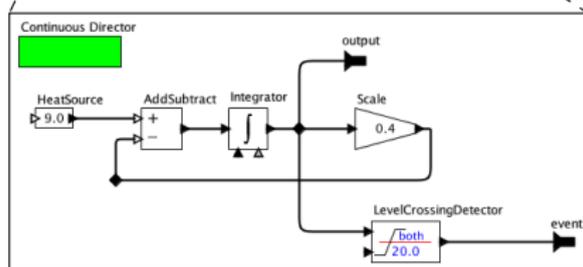
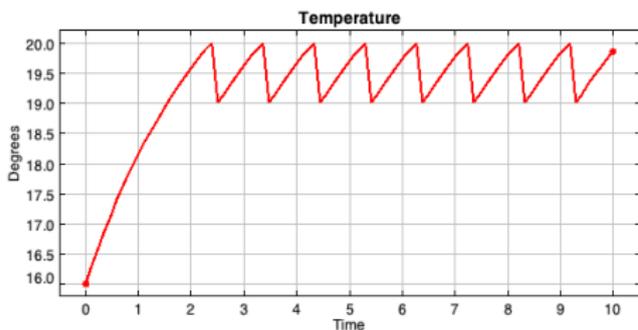
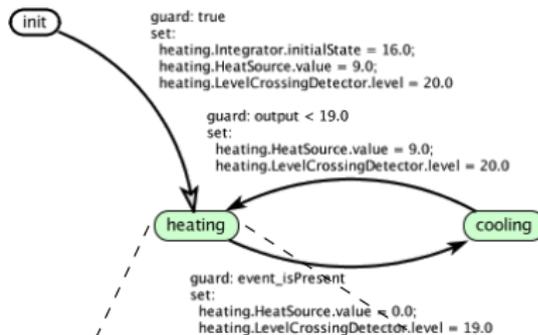
Validate

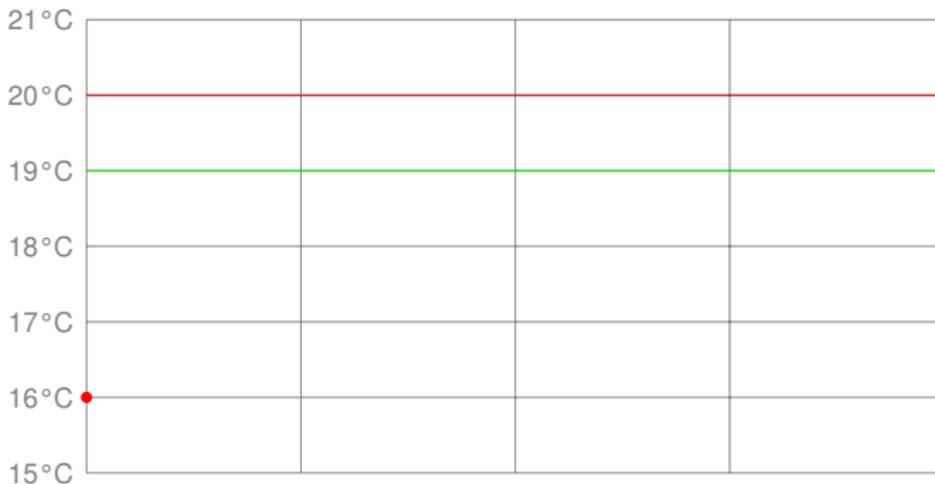


EndOfSnapshot

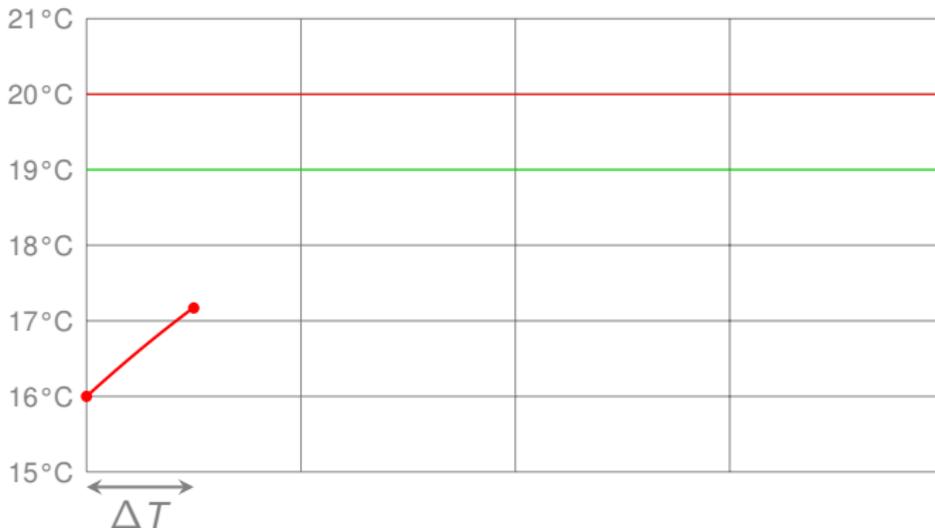


Exemple CT avec non validation





État = chauffage



État = chauffage

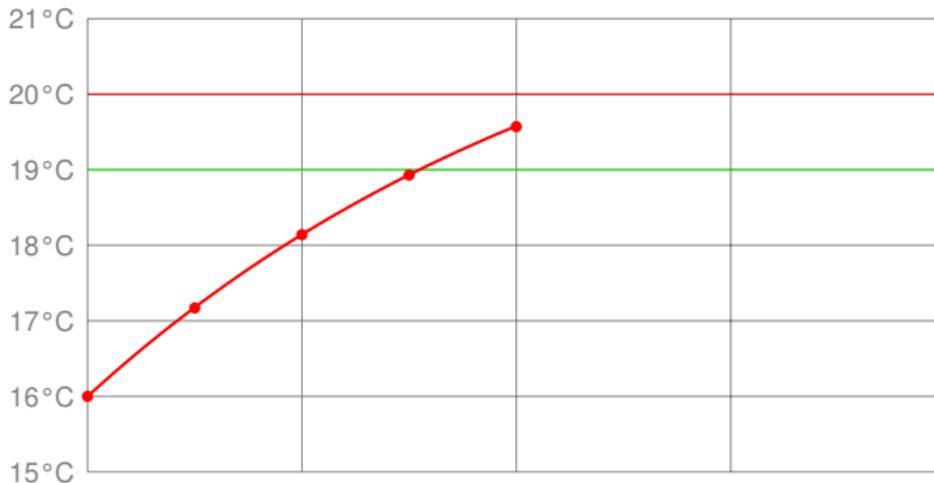
ΔT déterminé selon la précision sur la valeur de T



État = chauffage



État = chauffage

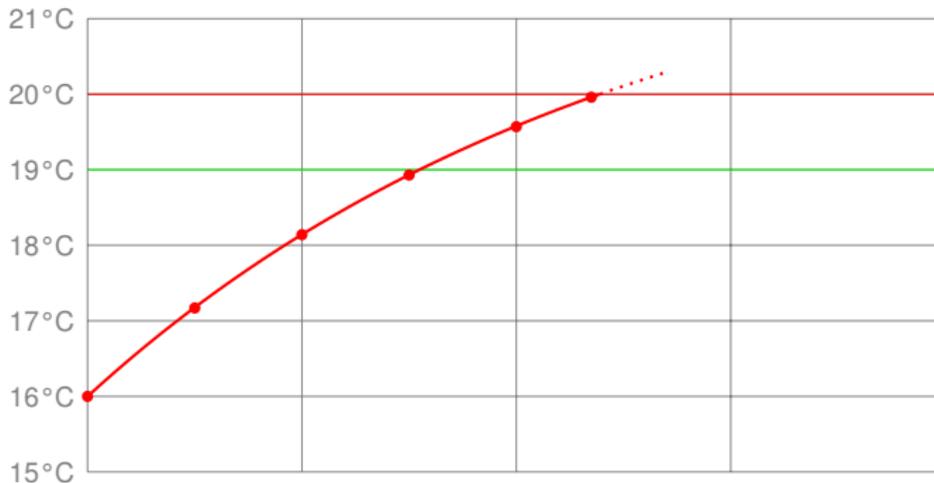


État = chauffage

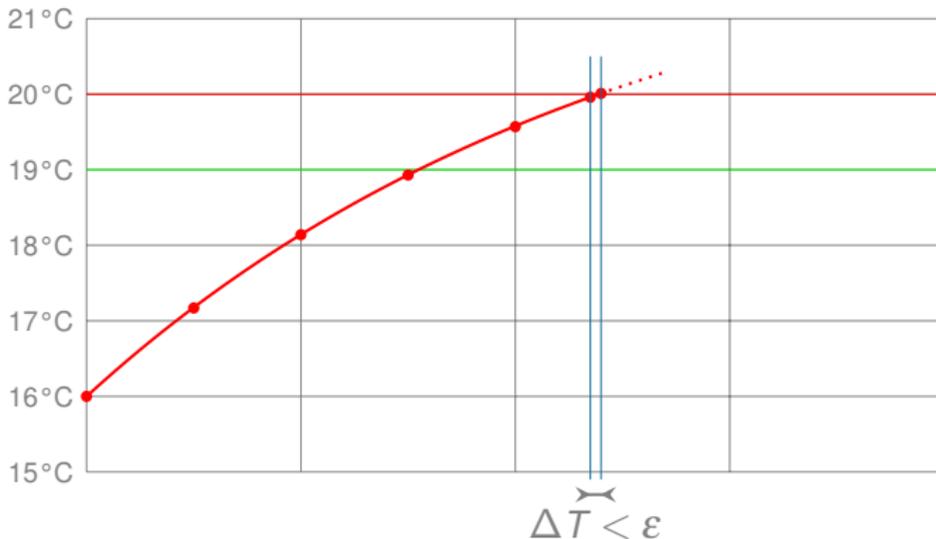


État = chauffage

Le seuil haut est franchi, mais on ne sait pas quand.
Le snapshot n'est pas valide

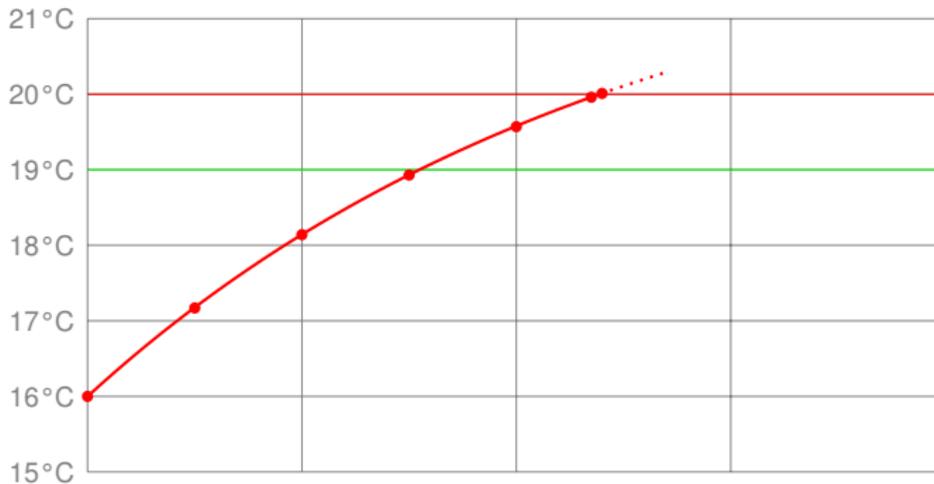


État = chauffage

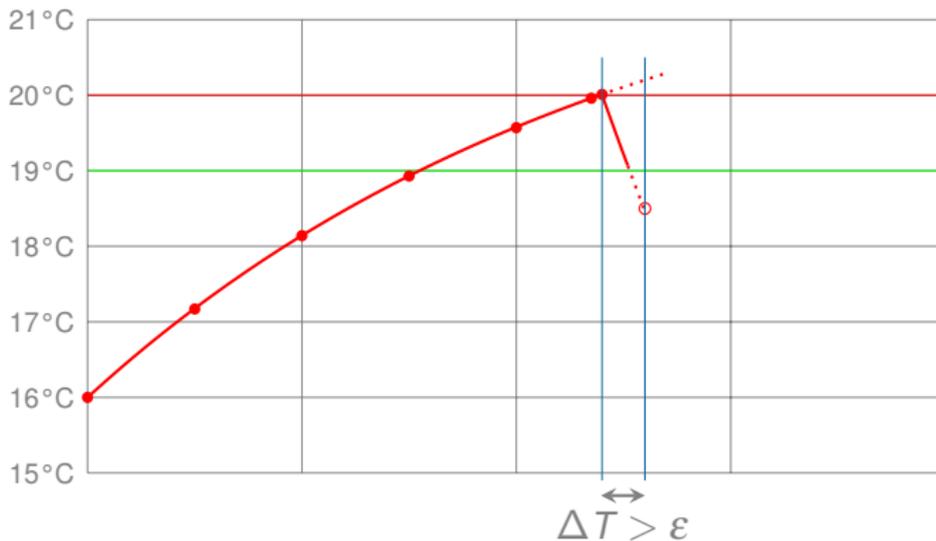


État = chauffage

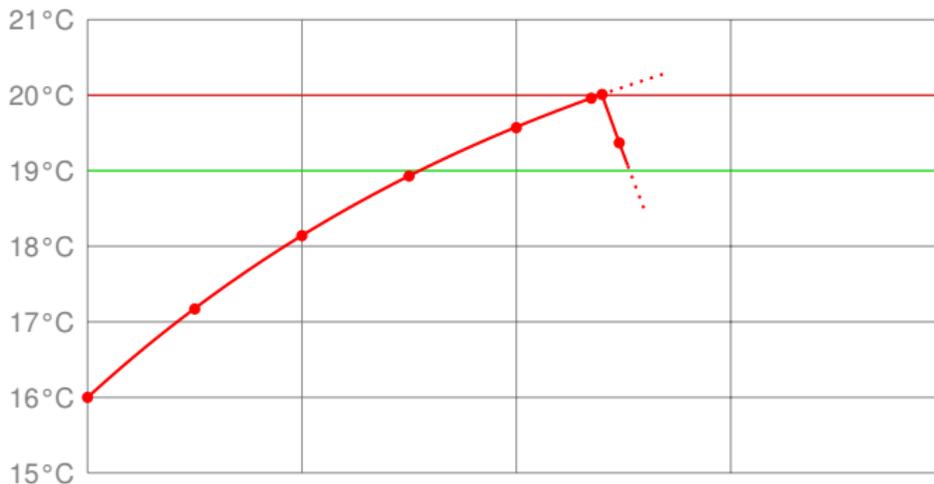
Le seuil haut est franchi, la date est suffisamment précise.
On change d'état



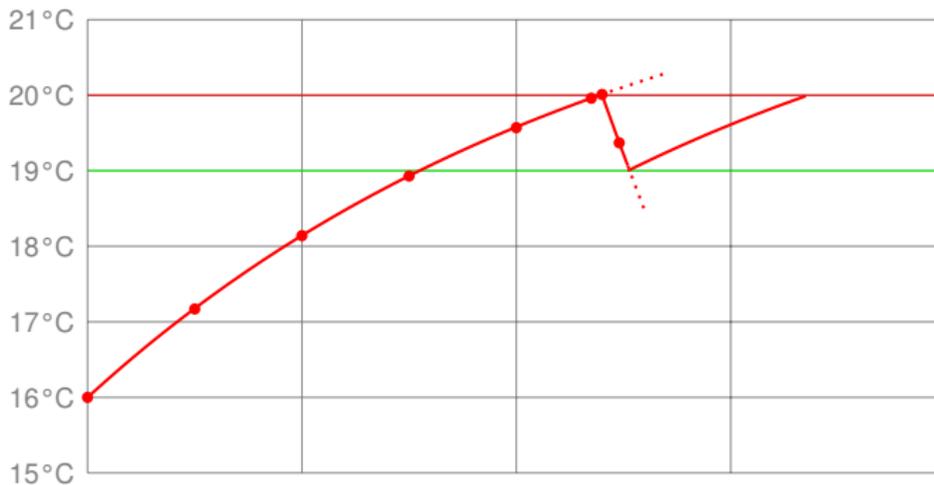
État = refroidissement



État = refroidissement



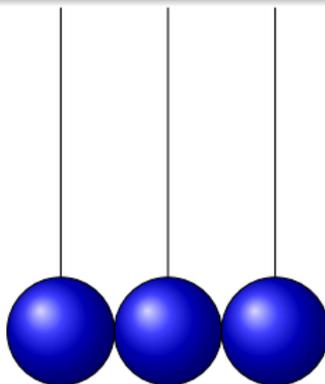
État = refroidissement



Objectif

Modéliser la causalité entre événements se produisant à une même date

Mise en séquence d'actions instantanées (idem synchrone)

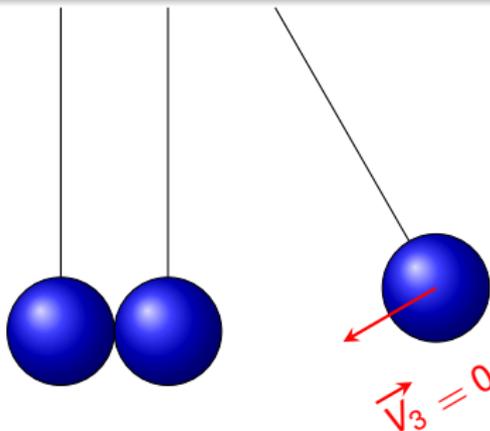


$$t \in \mathbb{R} \times \mathbb{N}$$

Objectif

Modéliser la causalité entre événements se produisant à une même date

Mise en séquence d'actions instantanées (idem synchrone)

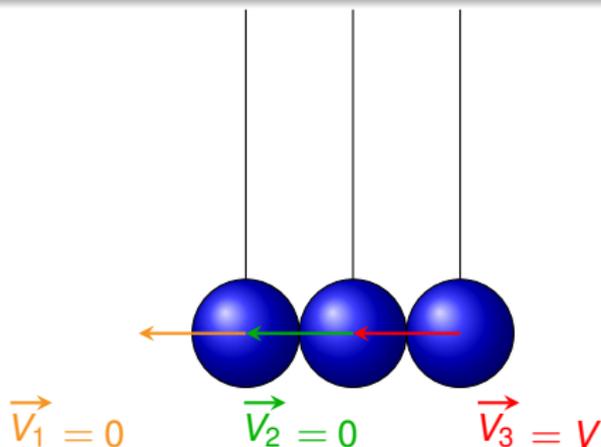


$$t = \langle t_0, 0 \rangle$$

Objectif

Modéliser la causalité entre événements se produisant à une même date

Mise en séquence d'actions instantanées (idem synchrone)

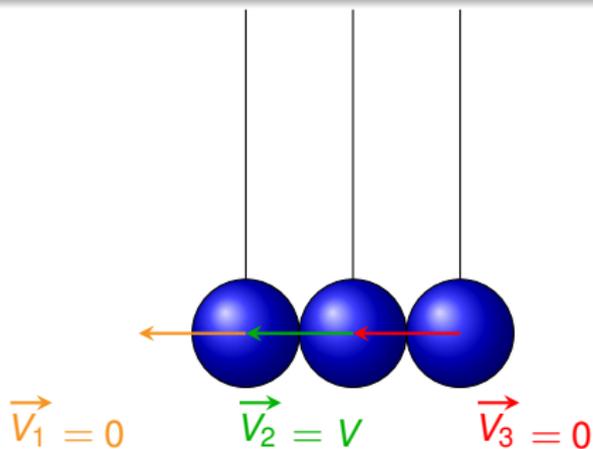


$$t = \langle t_1, 0 \rangle$$

Objectif

Modéliser la causalité entre événements se produisant à une même date

Mise en séquence d'actions instantanées (idem synchrone)

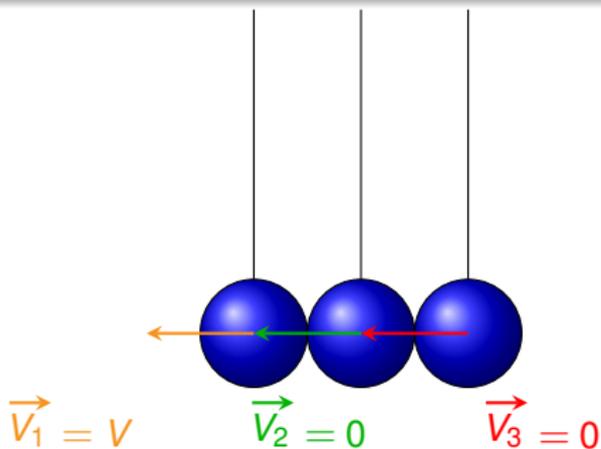


$$t = \langle t_1, 1 \rangle$$

Objectif

Modéliser la causalité entre événements se produisant à une même date

Mise en séquence d'actions instantanées (idem synchrone)

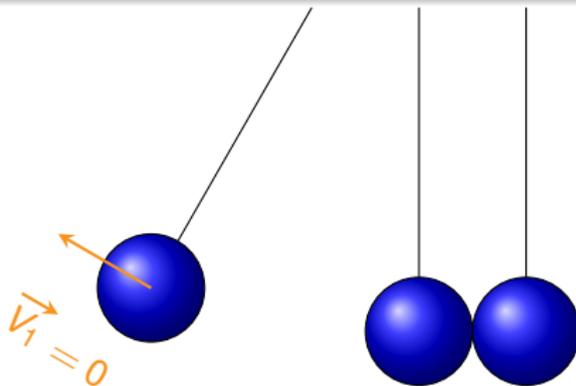


$$t = \langle t_1, 2 \rangle$$

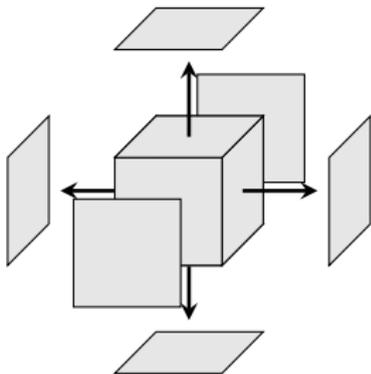
Objectif

Modéliser la causalité entre événements se produisant à une même date

Mise en séquence d'actions instantanées (idem synchrone)

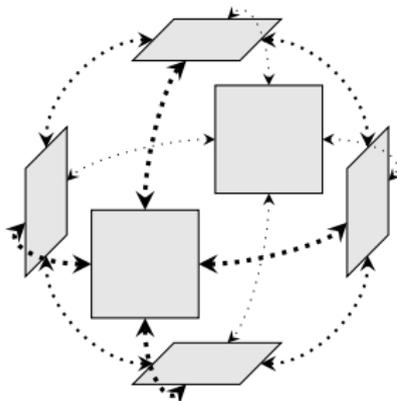


$$t = \langle t_2, 0 \rangle$$



Modèle global explicite

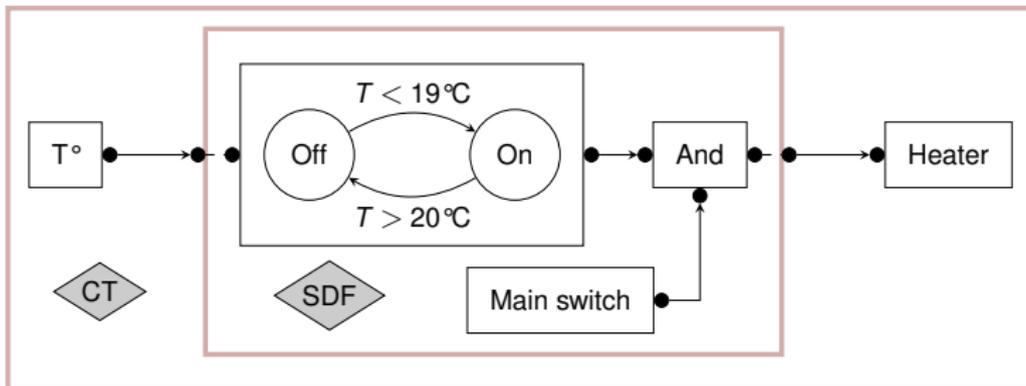
- + consistance
- + intégration
- langage
- ouverture



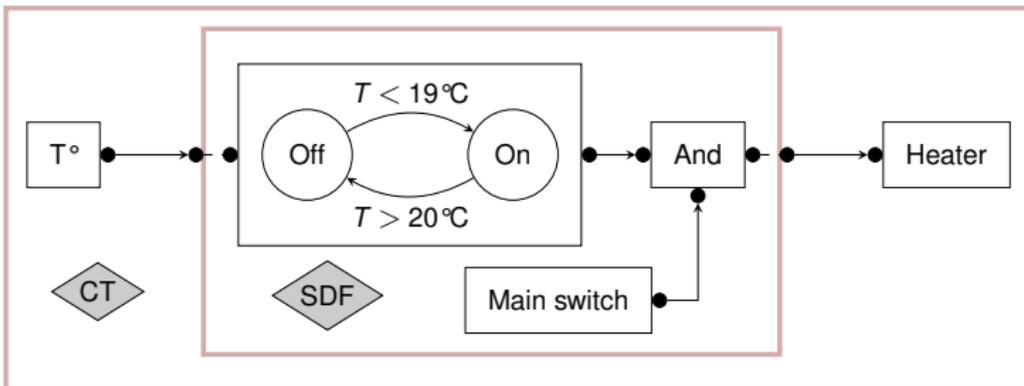
Modèle global virtuel

- consistance
- intégration/articulation
- + langages
- + ouverture

Functional view



Functional view



Power consumption view

