



AUTO-FORMATION BUILDSYSPRO 1. PRÉ-REQUIS

Support de formation de BuildSysPro
Dernière révision : janvier 2016

EDF R&D
Département Enerbat (Energie dans les Bâtiments et les Territoires)
Groupe « Simulation énergétique et bâti »





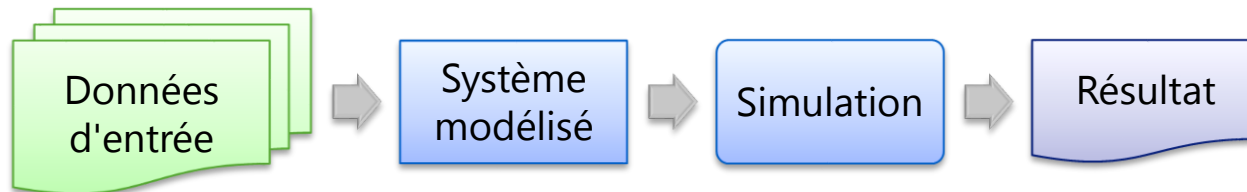
1. INTRODUCTION / PRÉ-REQUIS

- Pré-requis sur la simulation numérique (vocabulaire)
- Spécificités du langage Modelica
- Programmation orientée objet
- Programmation acausale

PRÉ-REQUIS SUR LA SIMULATION NUMÉRIQUE

■ Simulation numérique

- C'est un procédé permettant de reproduire sur ordinateur des phénomènes physiques
- Elle sert à étudier le fonctionnement et les propriétés d'un système et à en prédire l'évolution sans avoir recours à l'expérience (irréalisable, coûteuse, ...)
- Elle repose sur la programmation de modèles adaptés aux moyens numériques



■ Modèle

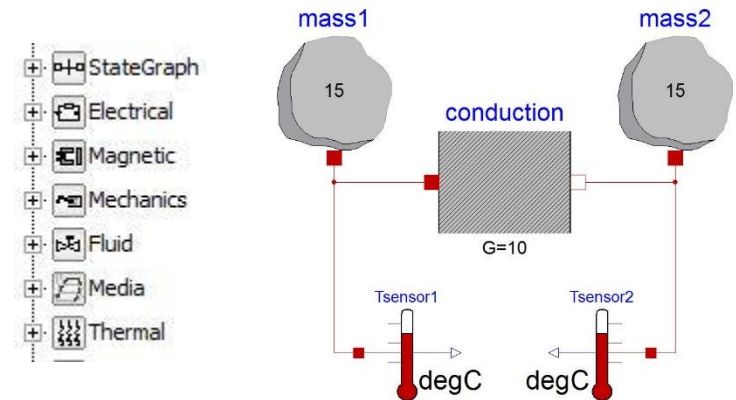
- C'est la traduction mathématique d'un ou plusieurs phénomènes physiques
- Il peut être plus ou moins précis dans sa description du phénomène

■ Système

- C'est l'objet qu'on souhaite étudier et qui est représenté par un ensemble de modèles interdépendants

SPÉCIFICITÉS DU LANGAGE MODELICA

- Modélisation multi-physique (électrique, thermique, mécanique,...)
 - Modélisation acausale avec visualisation des relations entre composants
 - Écriture formelle des équations (dont les équations différentielles)
 - Modélisation hybride : temps continu / événements discrets
- ➔ Développement de modèles plus rapides que dans d'autres langages
- ➔ Modèles accessibles aux non-informaticiens car guidés par la physique grâce à l'interface graphique et la programmation orientée objet



$$G \frac{dT}{dt} = Q_{flow}$$

equation

$Q_flow - G*dT = 0;$

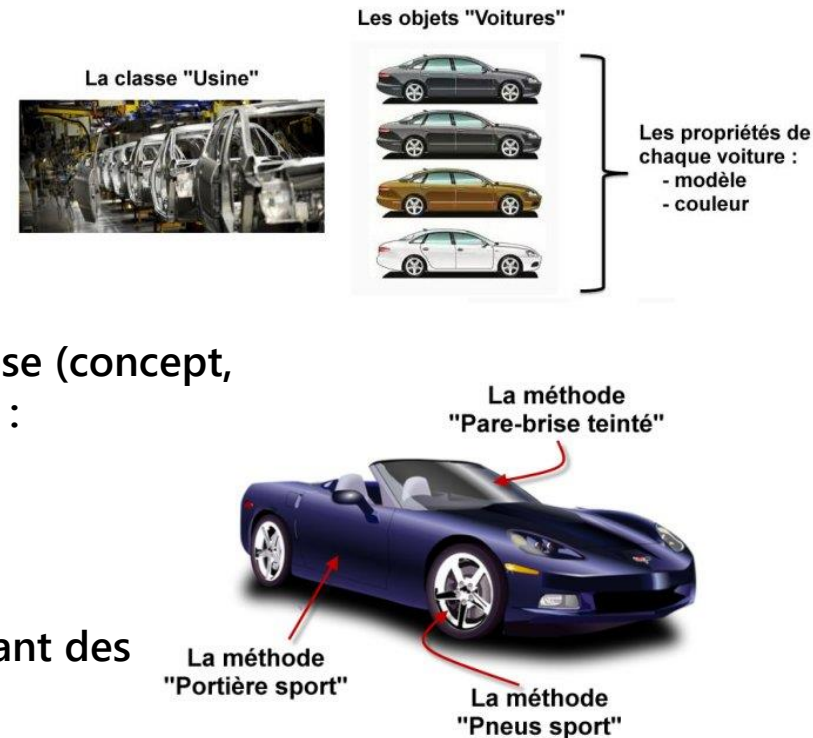
equation

$G*dT = Q_flow;$

```
model ThermalConductor
  extends Interfaces.Element1D;
  parameter ThermalConductance G
  "Constant thermal conductance of
  material";
  equation
    Q_flow = G*dT;
end ThermalConductor;
```

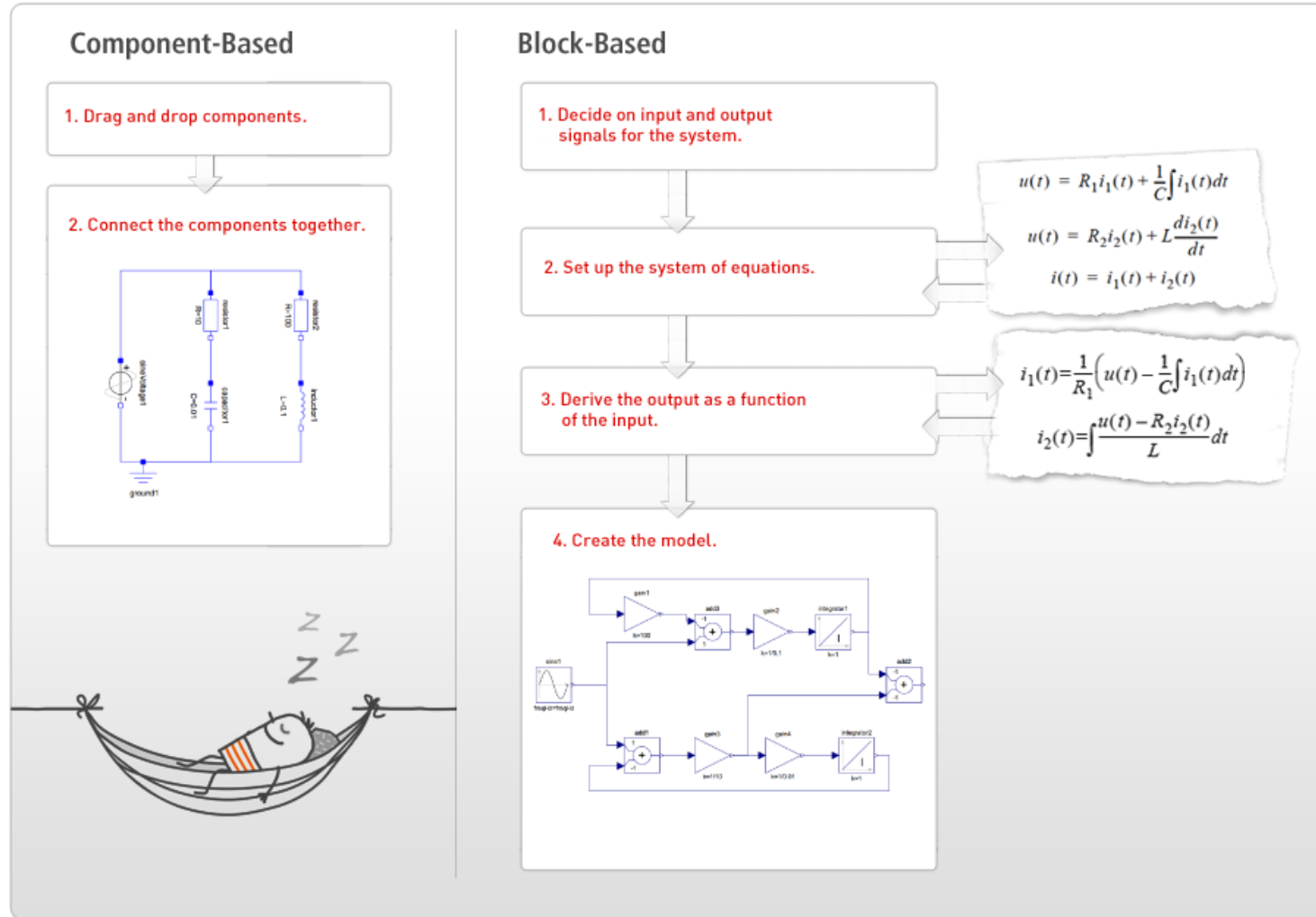
PROGRAMMATION ORIENTÉE OBJET (POO)

- Consiste en la définition et l'interaction de briques logicielles (objets)
 - S'oppose à la programmation procédurale
- Classe = Représentation abstraite de quelque chose (concept, idée, entité du monde physique), caractérisée par :
 - Des propriétés (attributs)
 - Des comportements (méthodes)
- Objet (ou instance) = Exemple d'une classe, ayant des propriétés spécifiques
 - Exemple : On déclare une classe « Pays », contenant des attributs comme « superficie », « drapeau », « monnaie ». A partir de cette classe on peut créer un objet « France », en remplissant les attributs de classe avec les valeurs spécifiques à ce pays.
- Avantages : Héritage multiple, encapsulation, souplesse de développement

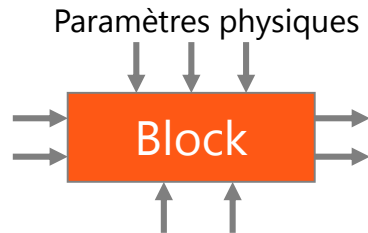


PROGRAMMATION ACAUSALE

« COMPONENT BASED »



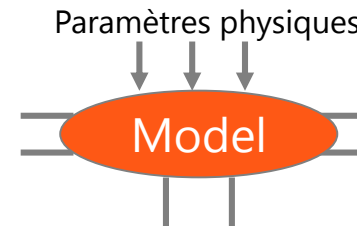
PROGRAMMATION ACAUSALE



Modélisation causale

Modélisation guidée par l'informatique

- Entrées = données importées de l'environnement
 - Sorties = données exportées vers l'environnement calculées à partir des entrées et des paramètres du modèle
- Avec BuildSysPro, on utilise ce type de modélisation lorsque l'on crée des fonctions



Modélisation acausale

Modélisation guidé par la physique

- Variables acausales (grandeurs physiques) interdépendantes que l'on observe dans le temps
 - Entrées/sorties causales possibles
- Avec BuildSysPro, on utilise ce type de modélisation lorsque l'on crée des modèles