# Vers une Approche de Modélisation du Domaine d'un Système Dans le Cadre de la Méthodologie SysML/KAOS

Steve Tueno[1,3]

**Sous l'encadrement de**

Régine Laleau[1], Amel Mammar[2], Marc Frappier[3]

[1]LACL – Université Paris Est Créteil Val de Marne, France
[2]SAMOVAR-CNRS – Télécom SudParis, France
[3]GRIL – Université de Sherbrooke, Canada

MFDL, GDR GPL, Paris *(07/12/2017)*

# Sommaire

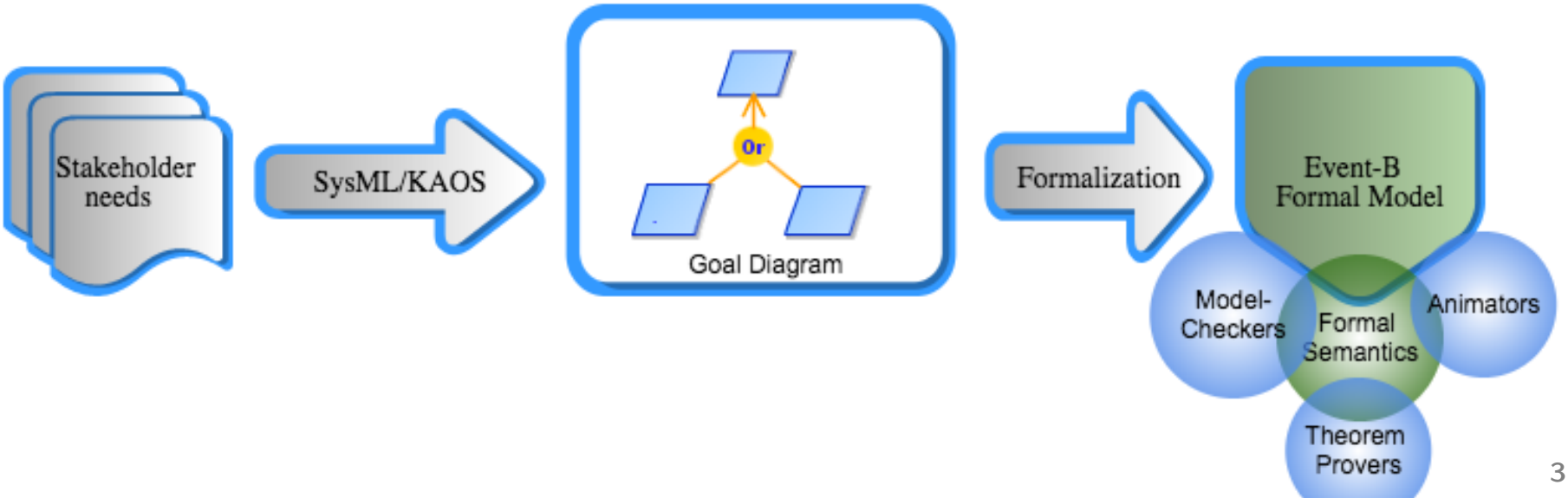
# L'approche *FORMOSE*

## projet FORMOSE (ANR-14-CE28-0009)

**FORMOSE:** Formal Requirements Modeling for Critical Complex Systems, Method and Toolkit

### objectif

Mettre en place une méthode et un ensemble d'outils pour la modélisation formelle des exigences de systèmes critiques et complexes



# La méthode *SysML/KAOS*

**SysML/KAOS :** Méthode permettant la modélisation des exigences d'un système au travers d'une hiérarchie de buts.

Plusieurs opérateurs servent à la hiérarchisation des buts. En ce qui concerne les exigences fonctionnelles, on distingue :

- l'opérateur ***AND*** pour la décomposition d'un but en une ***conjonction*** de sous-buts
- l'opérateur ***OR*** pour la décomposition d'un but en une ***disjonction*** de sous-buts
- l'opérateur ***MILESTONE*** pour la décomposition d'un but en un ***séquencement*** de sous-buts

# Formalisation d'un diagramme de buts *SysML/KAOS* (1/2)

- Chaque niveau de raffinement donne lieu à une machine Event-B
- Chaque but donne lieu à un évènement
- Les **liens de raffinement entre buts** se traduisent par des **obligations de preuve spécifiques**

# Formalisation d'un diagramme de buts *SysML/KAOS* (2/2)

## Extrait du cas d'étude *"Landing Gear System"* (ABZ 2014)

extraction/retraction des trains d'atterrissage d'un avion

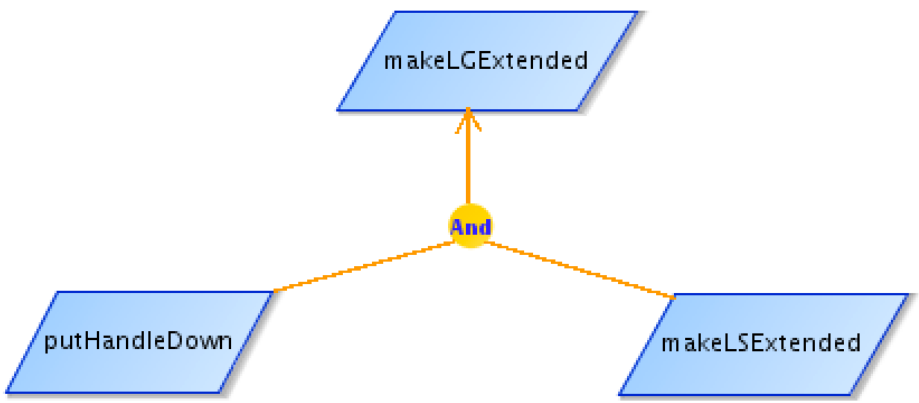



***Figure 1 :*** Extrait du diagramme de buts

*SYSTEM*
*lg_system_ref_0*
*EVENTS*
*makeLGExtended*=
**BEGIN**
// extension of the landing gear
**END**
**END**

***Figure 2 :*** Formalisation du niveau racine

*SYSTEM*
*lg_system_ref_0*
**SETS**
●●●
**CONSTANTS**
●●●
**PROPERTIES**
●●●
**VARIABLES**
●●●
**INVARIANT**
●●●

**INITIALISATION**
●●●
*EVENTS*
*makeLGExtended*=
**BEGIN**
// extension of the landing gear
●●●
**END**
**END**

**Le problème :**

Nécessité de compléter manuellement les éléments caractérisant **l'état du système** et **le corps des évènements**

# Objectifs

Sachant que la caractérisation de l'état du système passe par une interprétation des propriétés du domaine,

**Approche de modélisation des propriétés du domaine**

Proposer une approche de modélisation du domaine d'application du système

**Mise en correspondance de la modélisation du domaine avec la spécification formelle**

Proposer des mécanismes de génération de **la partie structurelle** du modèle formel à partir de la **représentation du domaine**

# Sommaire

# Présentation

**Définition**

**Modéliser le domaine** d'un système consiste à définir une représentation des **entités** qu'il manipule et des **propriétés** et **contraintes** associées.

Cette modélisation est effectuée dans la littérature à travers:

- **les Diagrammes UML (classes ou objets)** : semi-formels et faiblement expressifs (dynamique, raffinement, prédicats, ...)
- **les Ontologies**

# Ontologies

## Définition

Une ontologie est une **représentation formelle** d'un **ensemble** cohérent de **connaissances**.

## Formalismes existants de spécification d'ontologies

- **OWL** (Ontology Web Language)
- **RDFS** (Resource Description Framework Schema)
- **PLIB** (Part LIBrary)
- **F-Logic** (Frame Logic)
- **DAML+OIL** ( DARPA Agent Markup Language+Ontology Inference Layer)
- **DL** (Description Logic)

# Tableau comparatif des 3 principaux formalismes (1/2)

| Caractéristiques | OWL | PLIB | F-Logic |
|---|---|---|---|
| CWA vs OWA | OWA | CWA | CWA |
| Héritage | multiple | simple | multiple |
| Typage | faible | fort (chaque élément a un et un seul type) | faible |
| Expressivité | forte | faible | faible |
| Résonneurs | ++ | – | +- |
| Représentation graphique | ++ | – | – |
| Caractérisation de la variabilité d'un élément (Relation, Attribut ou Concept) | statique | statique | statique |

# Tableau comparatif des 3 principaux formalismes (2/2)

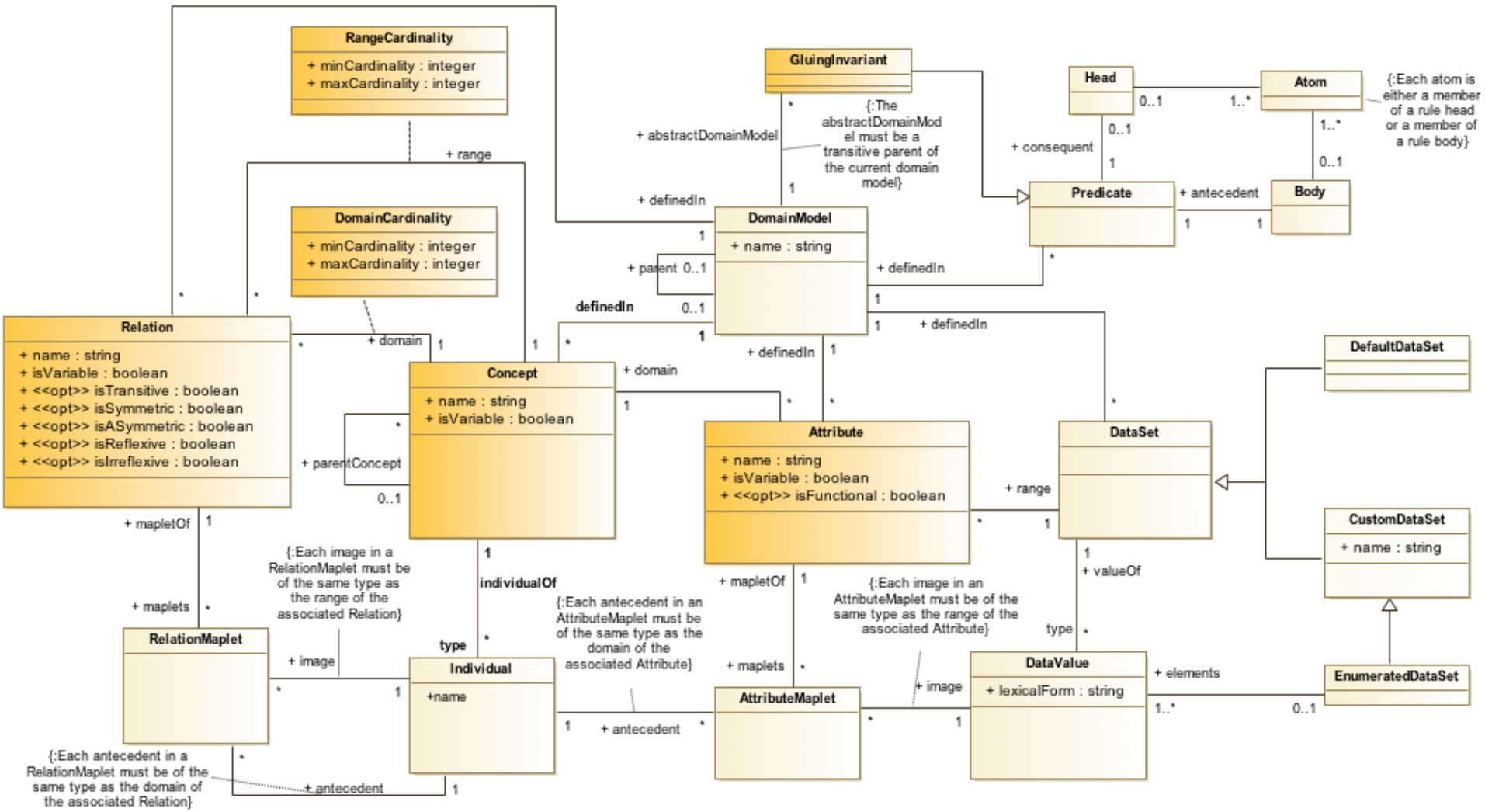
## Conclusion

Aucun des formalismes ne satisfait totalement nos exigences, *notamment la possibilité de* ***modéliser des entités dynamiques*** *: d'où peuvent provenir les variables de la spécification formelle?*

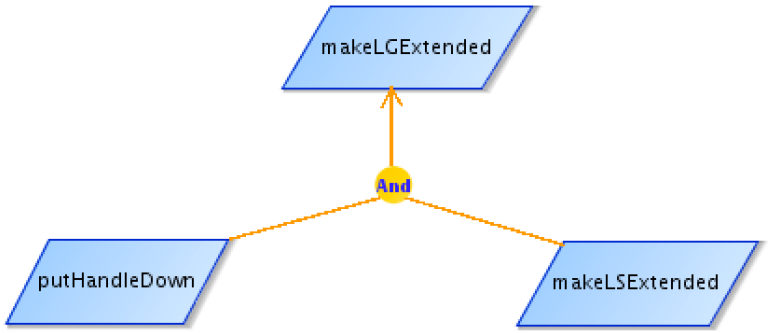
$\Rightarrow$

Mise en place d'un **nouveau formalisme** de modélisation d'ontologies basé sur *OWL* et *PLIB* qui apparaissent comme étant les mieux allignés avec nos objectifs.

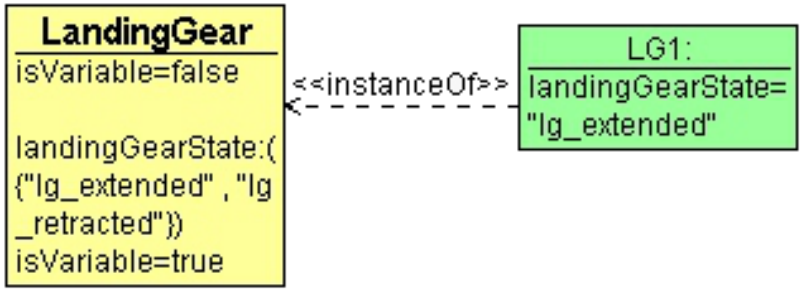
# Métamodèle du nouveau formalisme [MoDRE 2017]



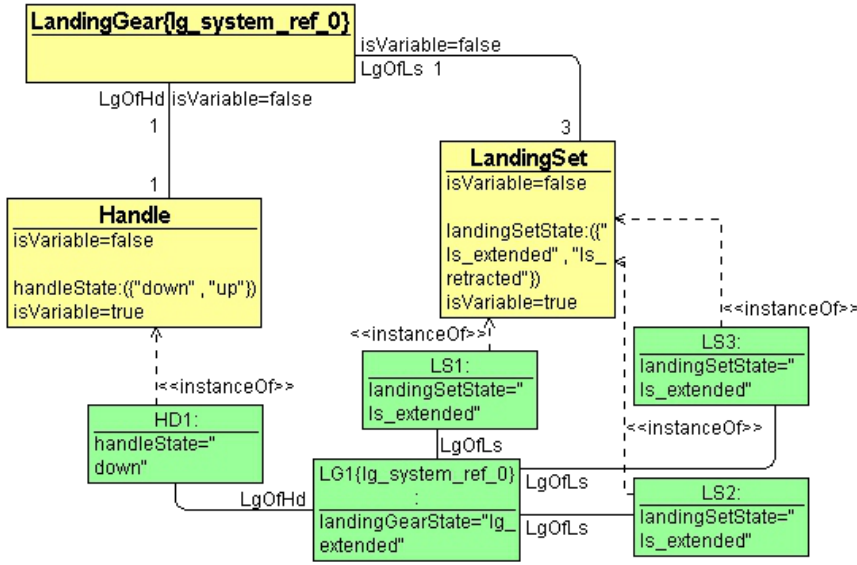
# Illustration sur le cas d'étude




Extrait du diagramme de buts



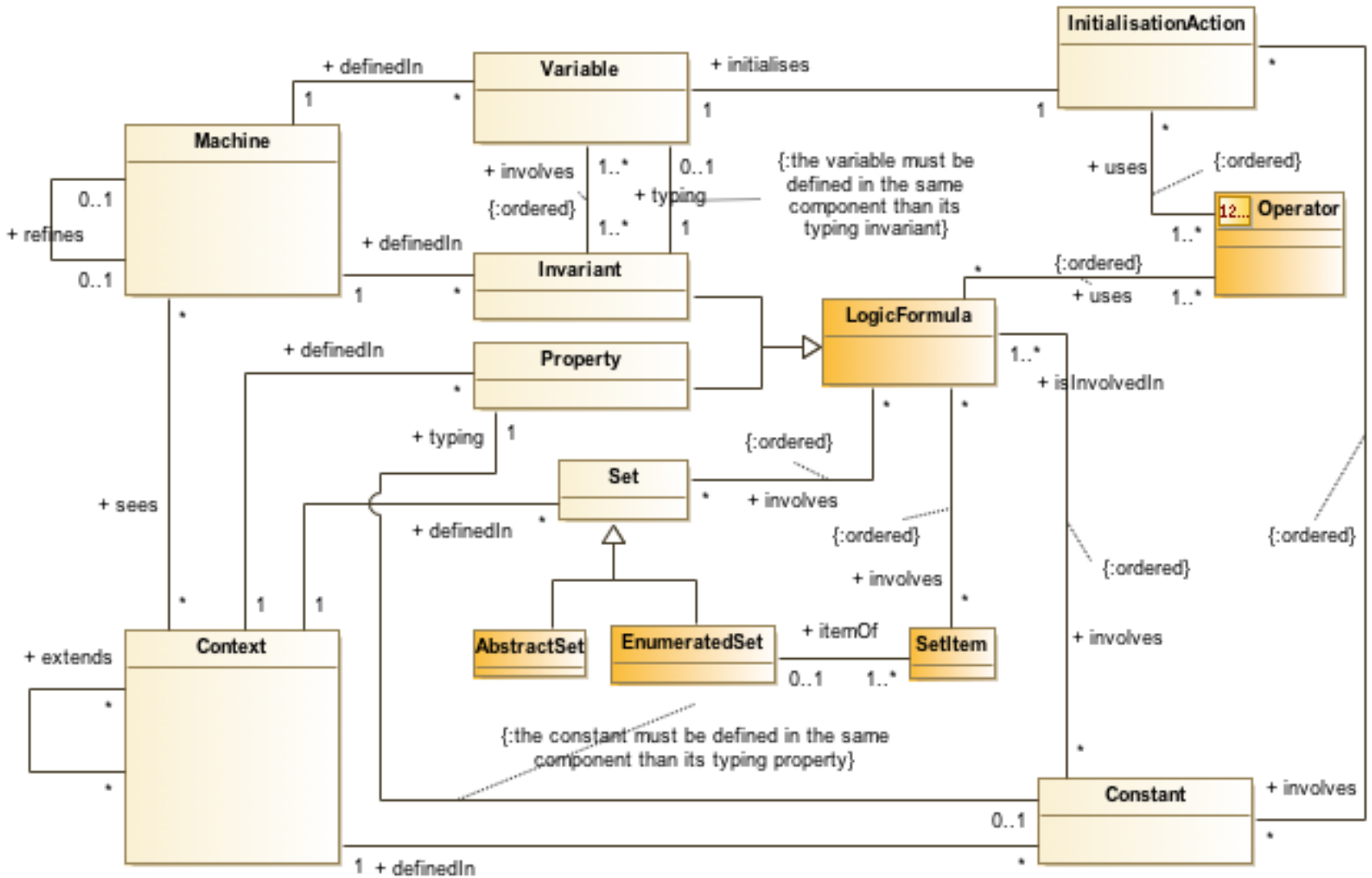

***lg_system_ref_0***: ontologie du niveau racine




***lg_system_ref_1***: ontologie du premier niveau de raffinement

# Sommaire

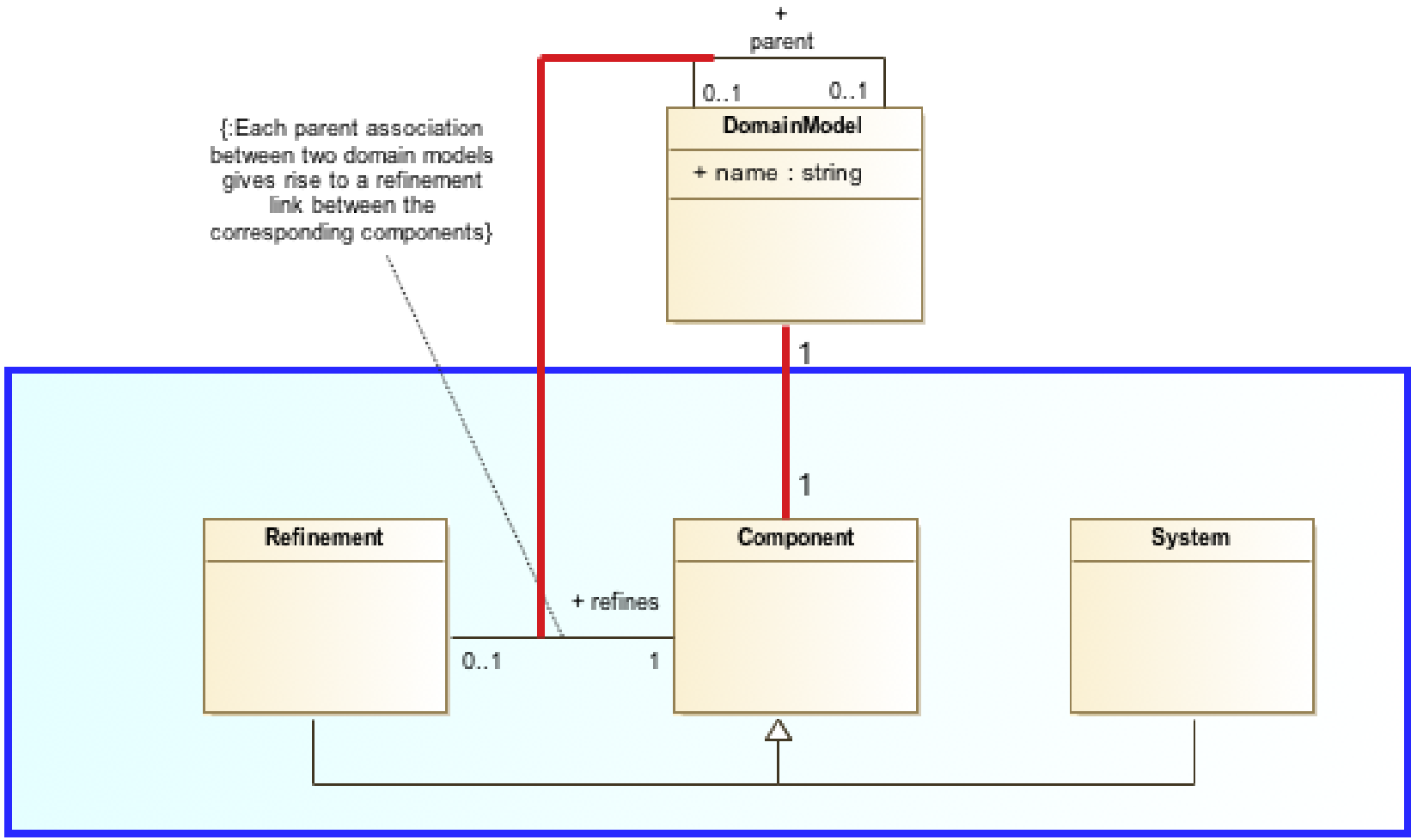
# Métamodèle des Éléments d'Event-B pour Formose



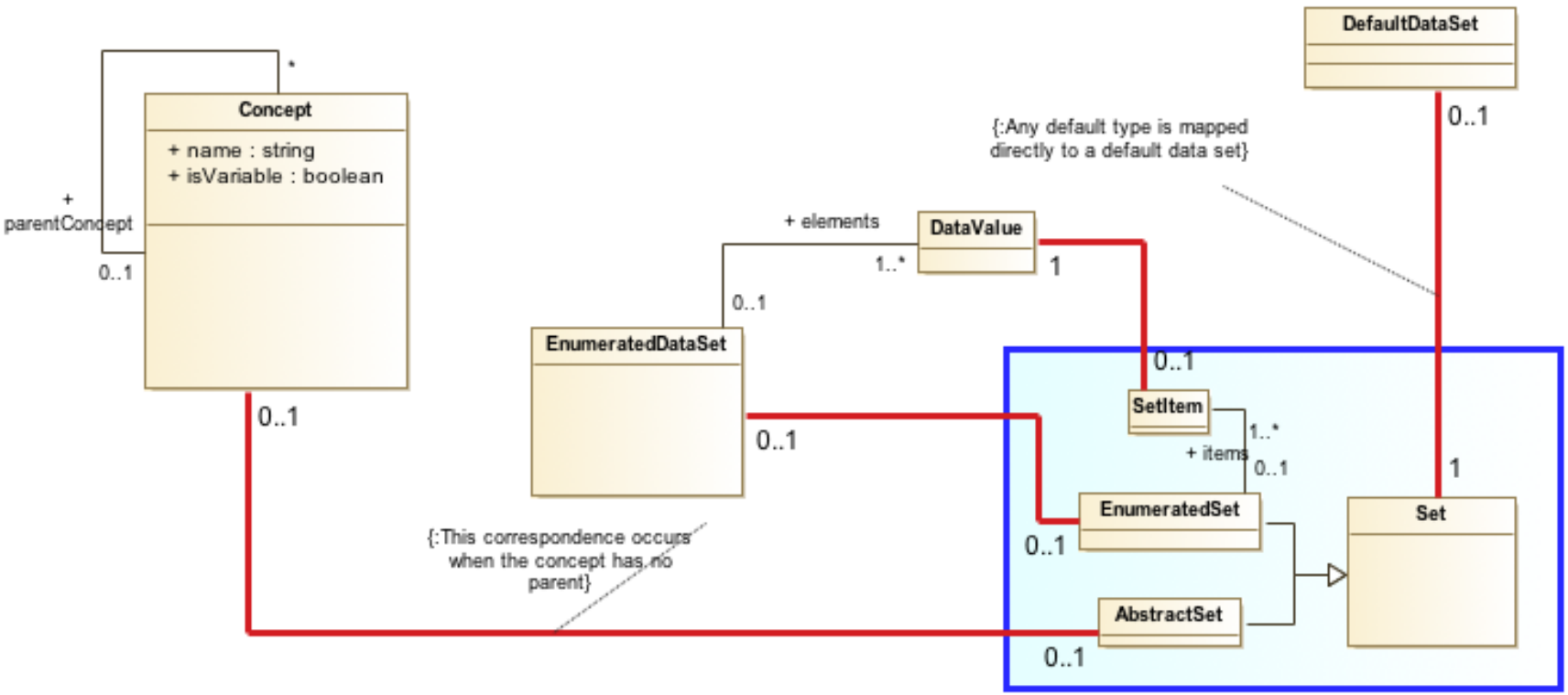
# Quelques Opérateurs

- $(Inclusion\_OP, [op_1, op_2]) \Leftrightarrow op_1 \subseteq op_2$
- $(Belonging\_OP, [op_1, op_2]) \Leftrightarrow op_1 \in op_2$
- $(BecomeEqual2SetOf\_OP, [va, op_2, ..., op_n]) \Leftrightarrow va := \{op_2, ..., op_n\}$

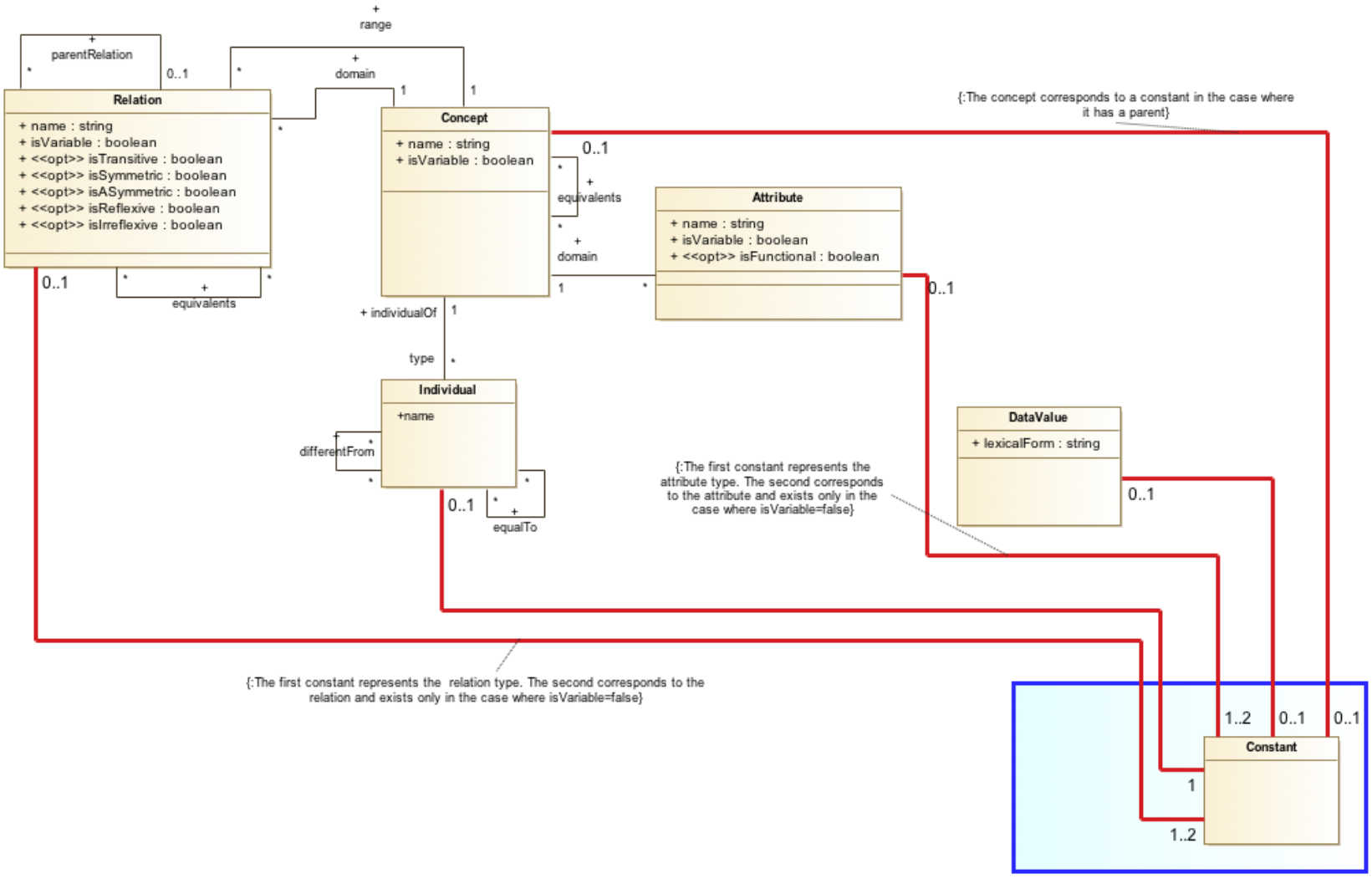
# Traduction vers des instances de *Component*



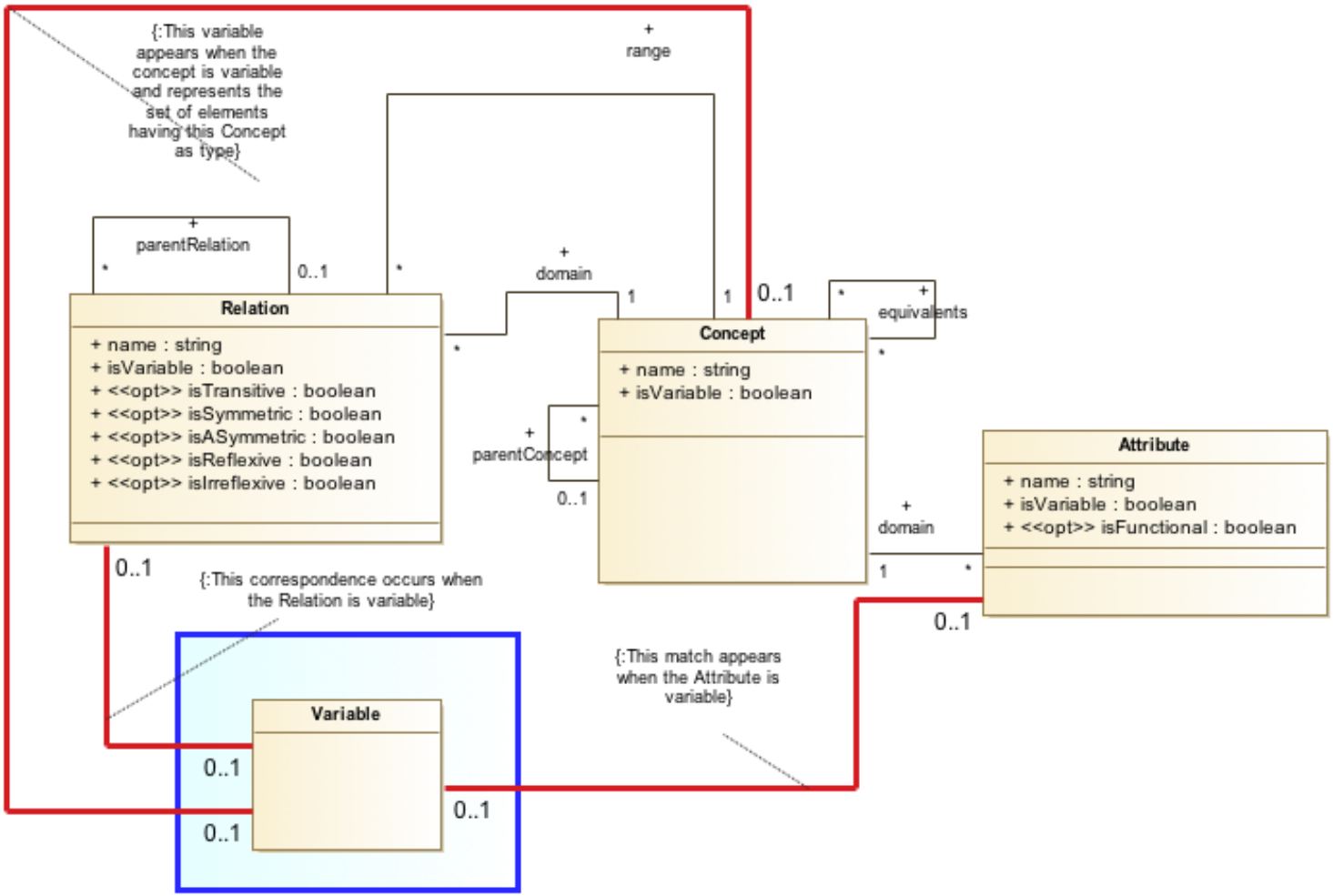
# Traduction vers des instances de *Set*



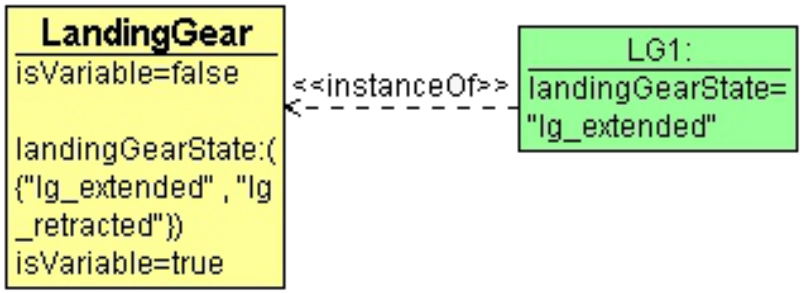
# Traduction vers des instances de *Constant*



# Traduction vers des instances de *Variable*



# Illustration sur le cas d'étude




***lg_system_ref_0***: ontologie du niveau racine

*SYSTEM* *lg_system_ref_0*
***SETS*** *LandingGear; DataSet_1={lg_extended, lg_retracted}*

**CONSTANTS**
T_landingGearState, LG1
**PROPERTIES**
LG1 ∈ LandingGear
∧ LandingGear={LG1}
∧ T_landingGearState = LandingGear ⟶ DataSet_1
**VARIABLES** landingGearState
**INVARIANT**
landingGearState ∈ T_landingGearState
**INITIALISATION**
landingGearState := {LG1 ↦ lg_extended }
END

# Sommaire

# Conclusion

- Modélisation du domaine sous forme **d'ontologies**
- Proposition d'un **formalisme pour l'expression des ontologies** basé sur *OWL* et *PLIB* qui sont les formalismes les mieux alignés avec nos objectifs
- **Liens de correspondance entre ontologies et spécifications formelles**

# Axes de recherche

- **Propagation des mises à jour de la spécification Event-B au sein de l'ontologie et réciproquement**
- **Implémentation de l'approche sous Openflexo**