Vérification formelle de propriétés : Application de l'outil OBP au cas d'étude CCS

Philippe Dhaussy, Luka Le Roux, Ciprian Teodorov

Univ. Européenne de Bretagne
Lab-STICC
UMR CNRS 6285
ENSTA-Bretagne, Brest.

philippe.dhaussy@ensta-bretagne.fr

Motivation : Intégration des méthodes formelles dans les processus d'ingéniérie

Notations

Exigences

Cas d'utilisation

Modèle Applicatif

Processus

Formalisation

Modèle formel à valider

Outils

Model-Checkers

Objectif : *Etudier les hypothèses et les conditions opérationnelles pour rendre possible l'ntégration des méthodes dans les processus.*

Disposer de modèles formels pour les outils de vérification

Plan

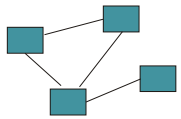
- Principe de vérification de propriétés
- L'outillage OBP et langage CDL
  - Contextes
  - Propriétés
- Application au cas CCS

Principe de vérification de propriétés

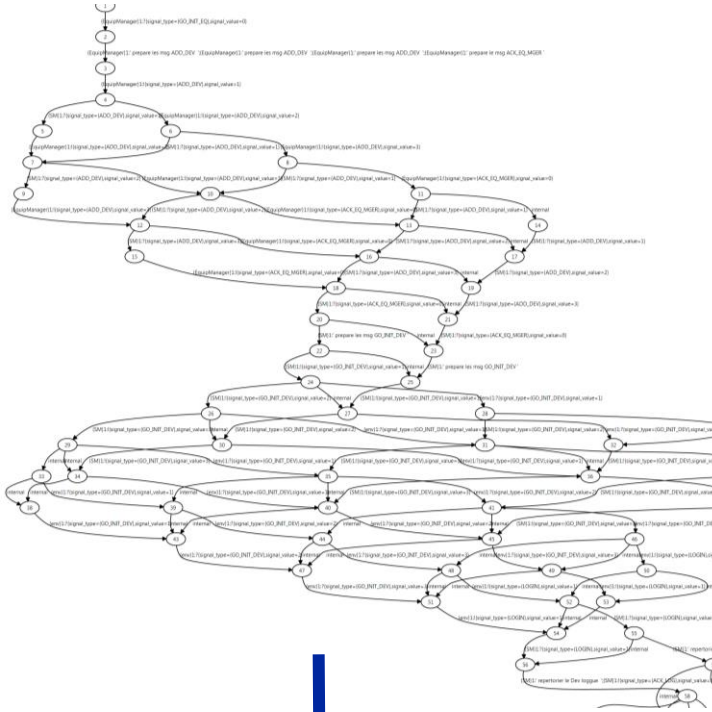
Système de transitions étiquetées

Propriétés formelles

Modèle formel à valider



Simulateur, Model-checker



trace

Verdict

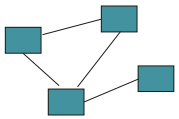
Analyse d'accessibilité

Principe de vérification de propriétés

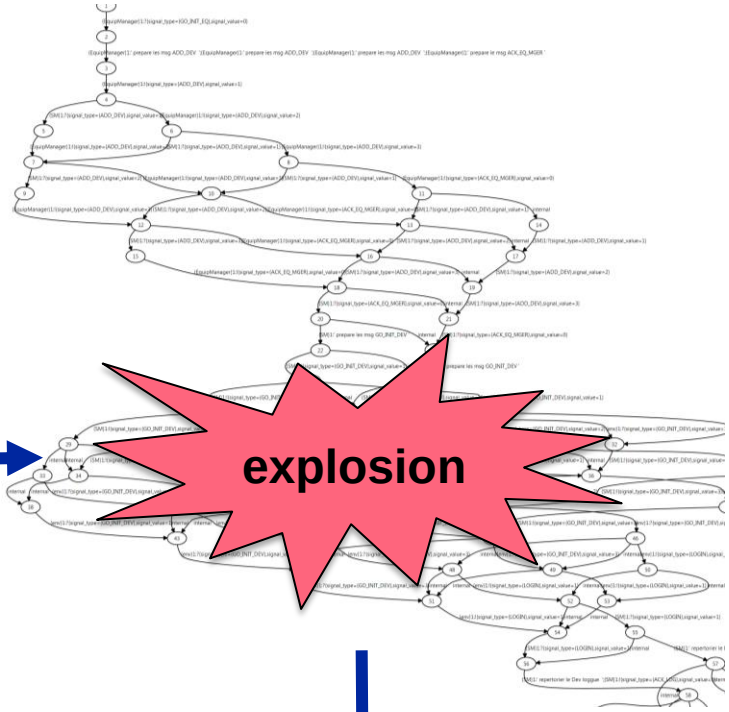
Système de transitions étiquetées

Propriétés formelles

Modèle formel à valider



Simulateur, Model-checker



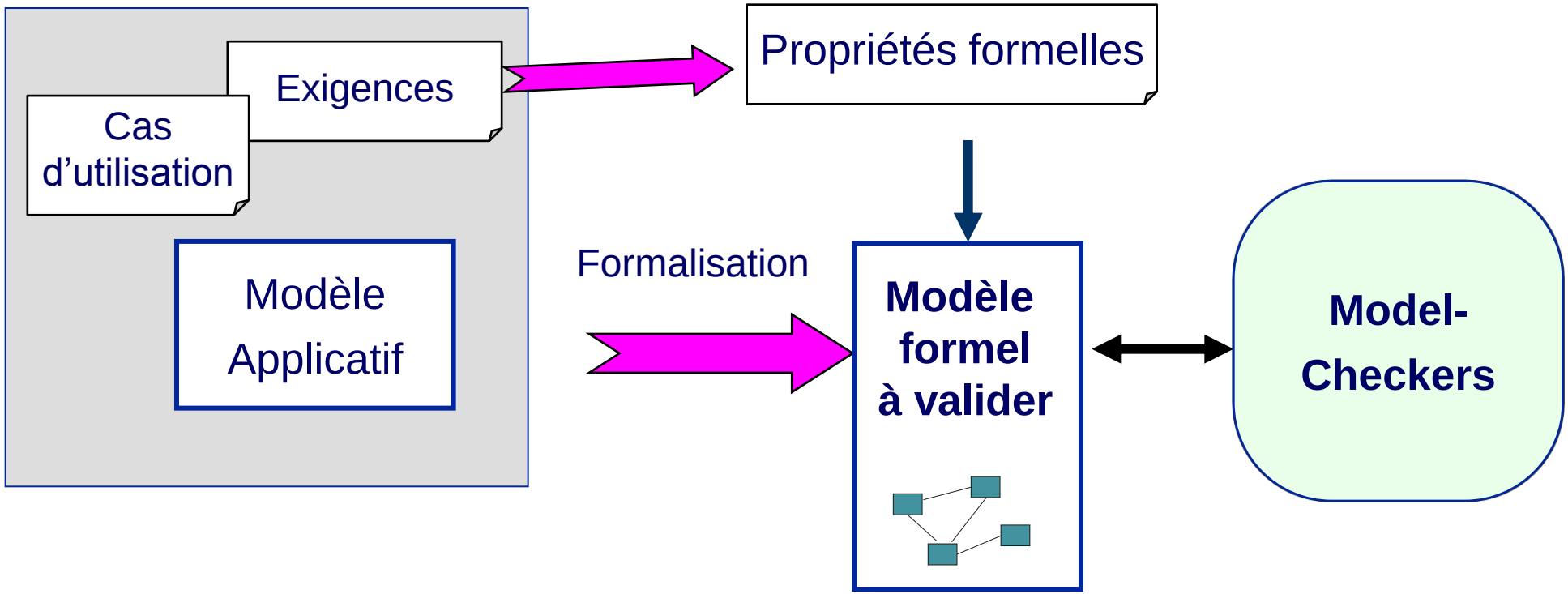
explosion

trace

Verdict

Analyse d'accessibilité

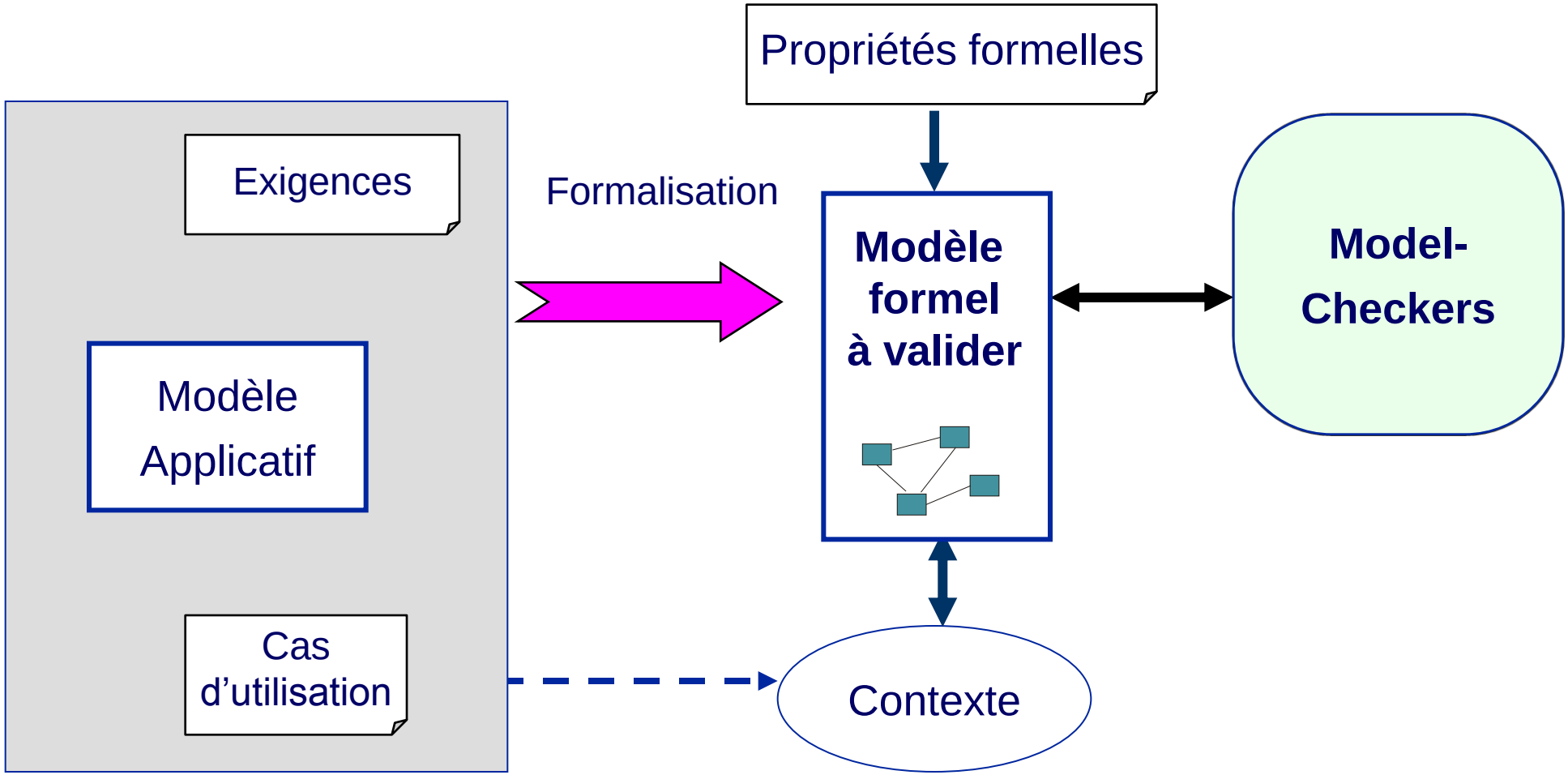
Difficulté: formalisation des propriétés



Gap entre le modèle à valider et le modèle formel

Logique temporelle : non adéquat

Lien : Contexte – propriété



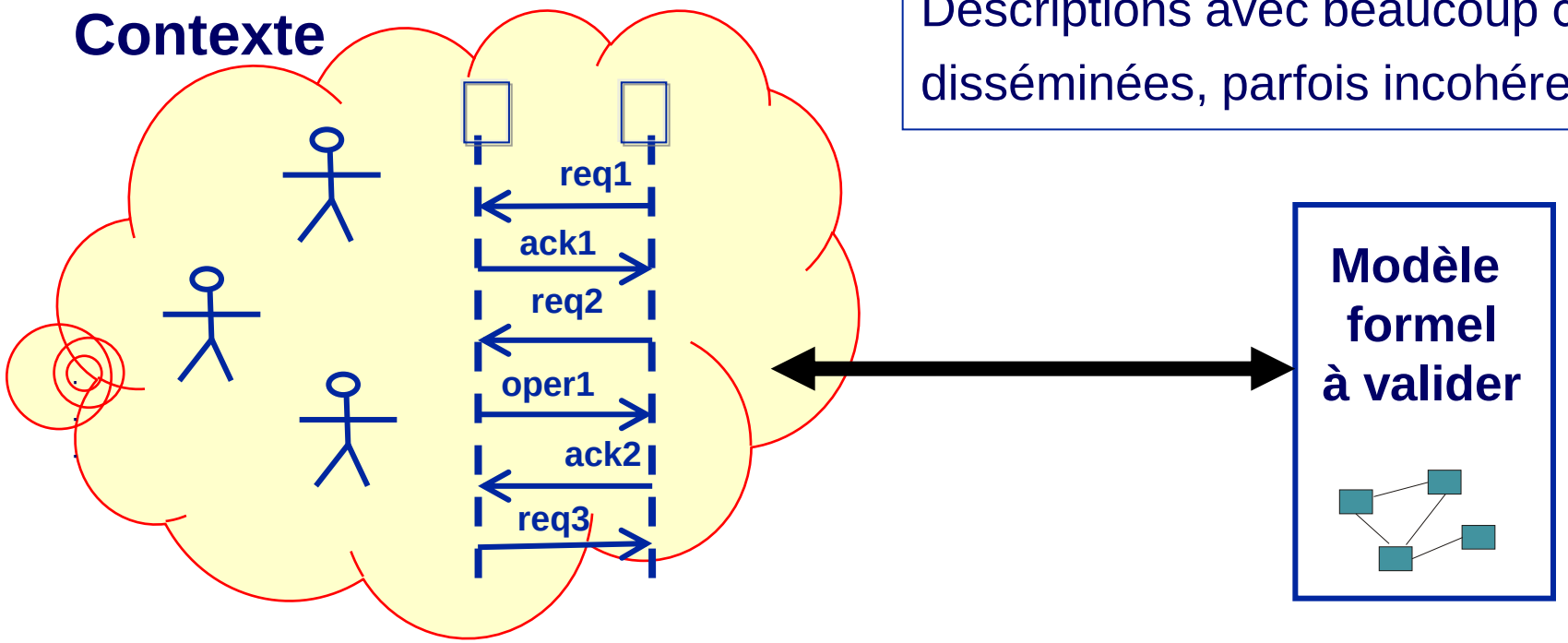
Vérifier les propriétés dans leur contexte

Expression des contextes

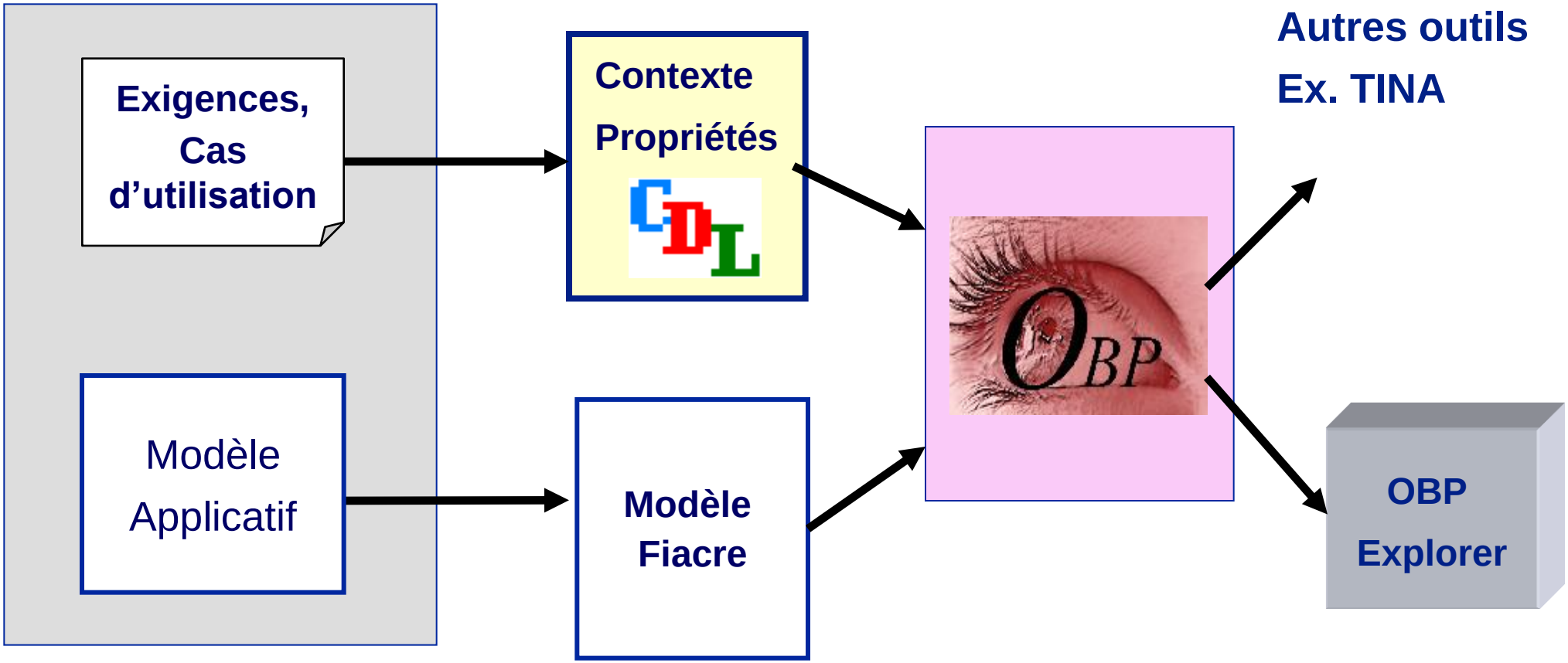
Représentent le comportement de l'environnement (Phases opérationnelles)

Initialisation, reconfiguration, modes dégradés, scénarios d'erreurs, etc.

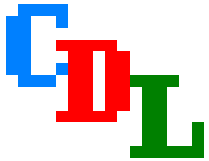
En réalité dans les spécifications :
Descriptions avec beaucoup d'implicite, disséminées, parfois incohérentes



L'outillage OBP et le langage CDL

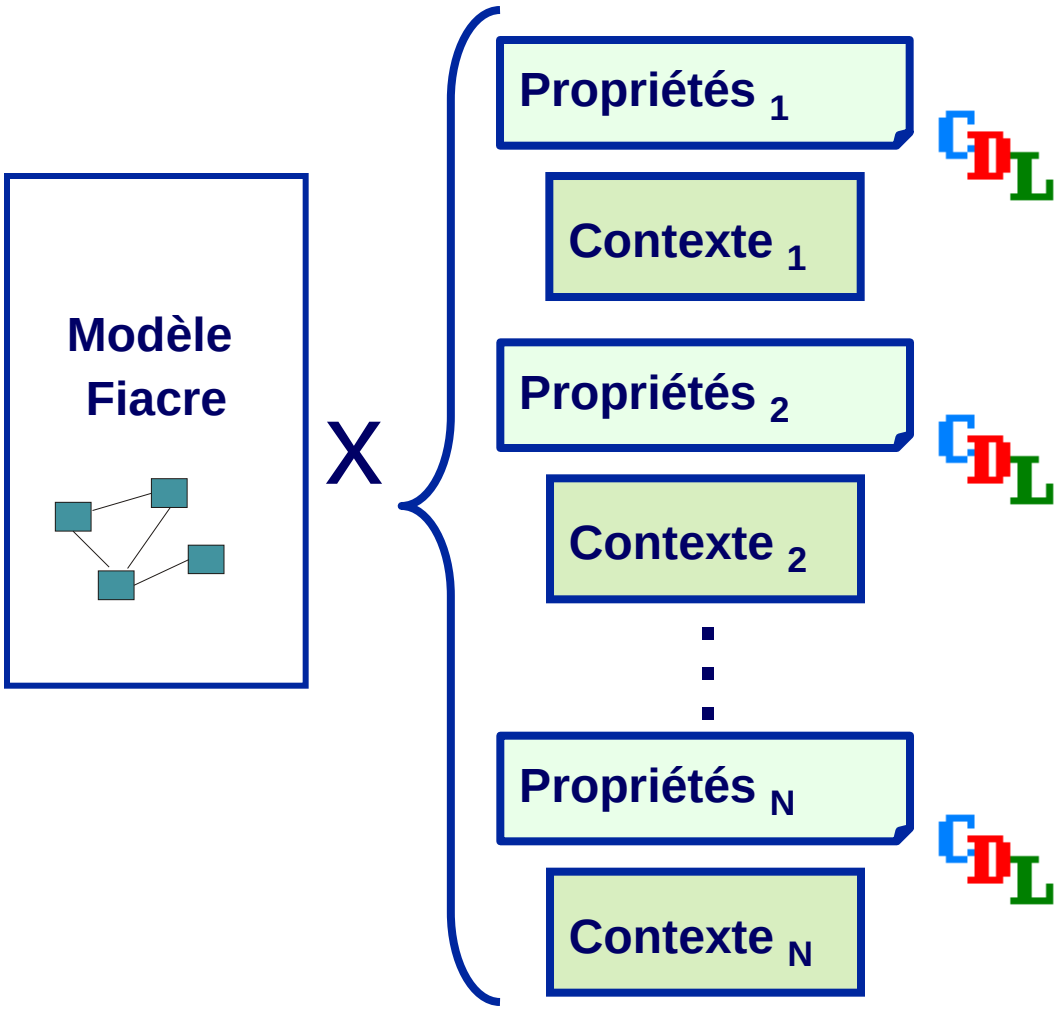


Diffusion : www.obpcdl.org

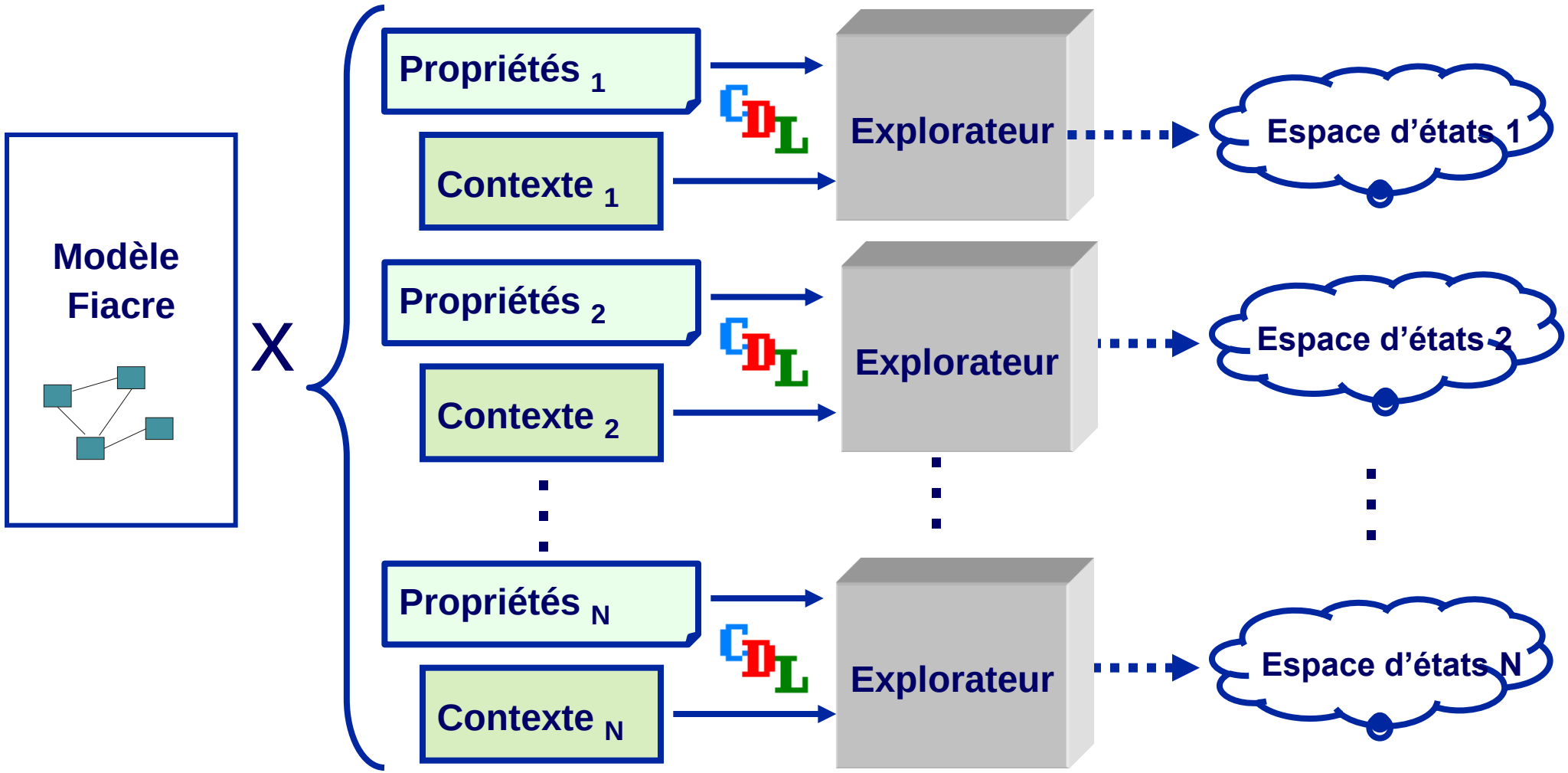
Langage 

- ***Spécification des contextes***
- ***Spécification des propriétés***

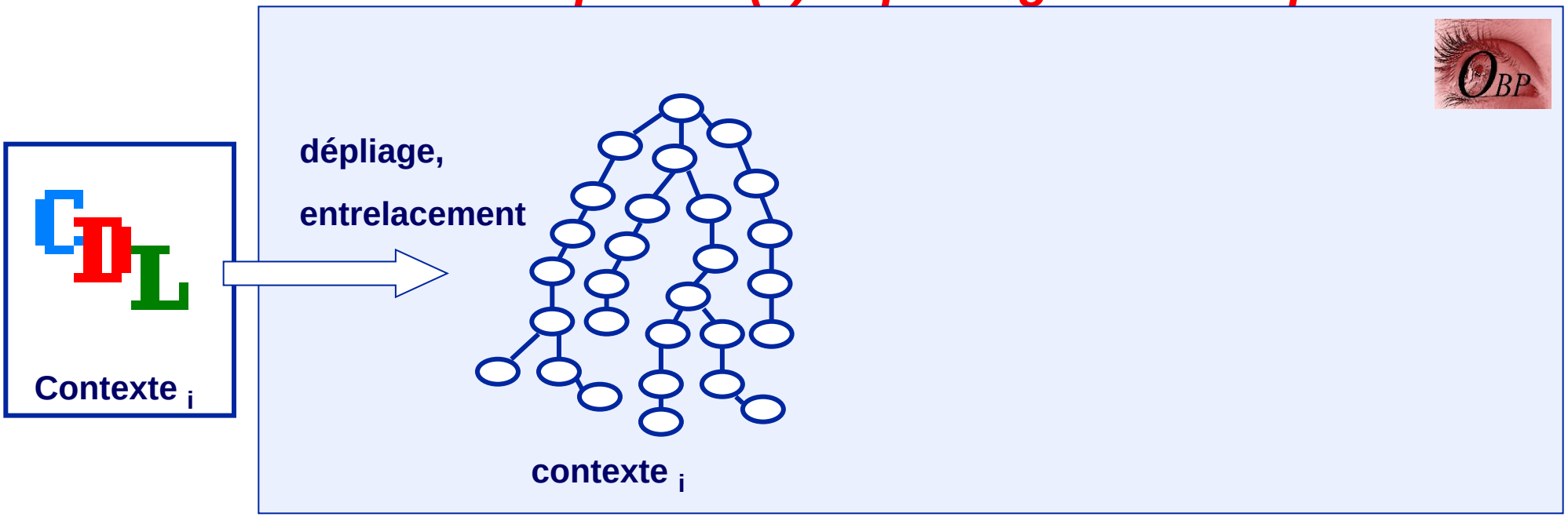
Réduction de la complexité (1) : Identification des contextes



Réduction de la complexité (1) : Identification des contextes

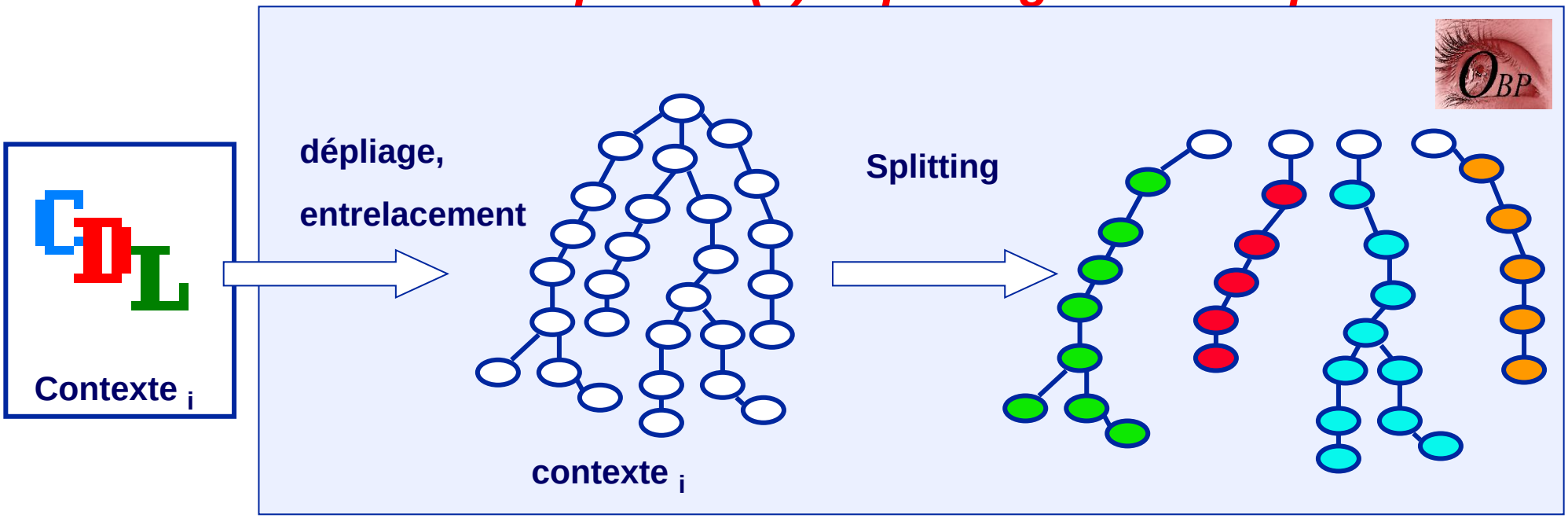


Réduction de la complexité (2) : splitting automatique



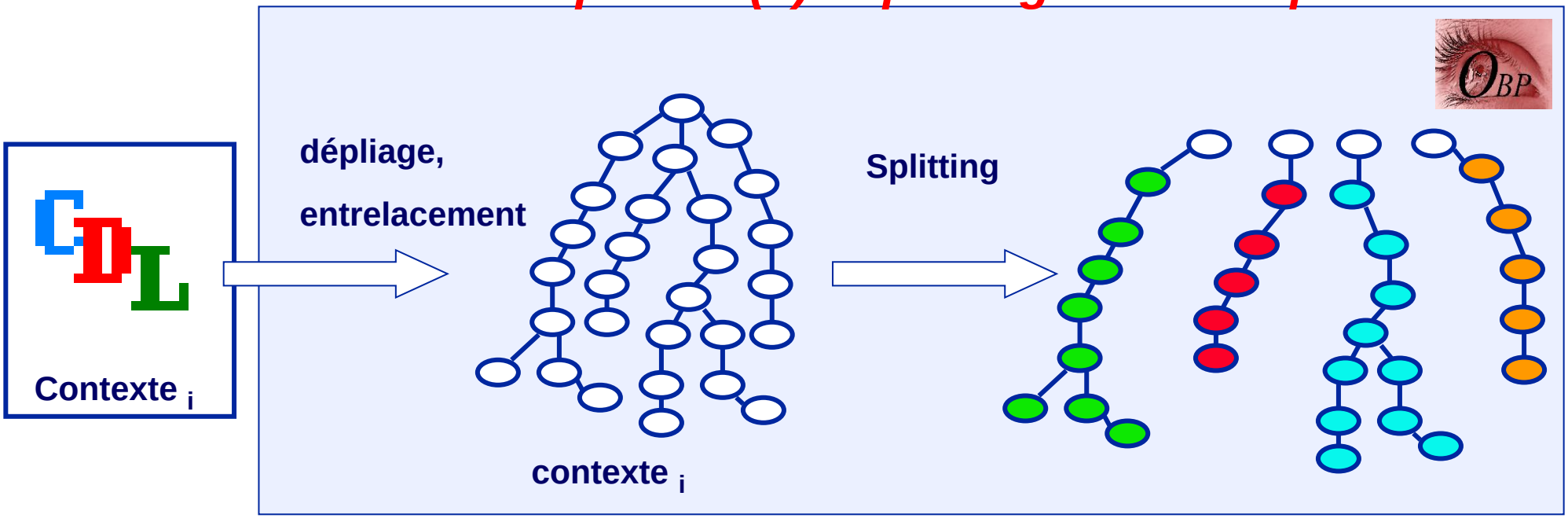
Contextes : Finis et acycliques

Réduction de la complexité (2) : splitting automatique



Contextes : Finis et acycliques

Réduction de la complexité (2) : splitting automatique



Ensemble de K_i sous-contextes

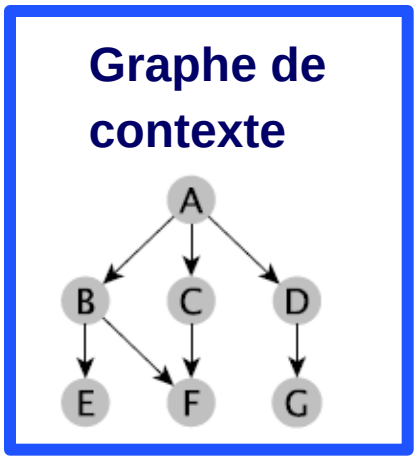
Modèle à valider (Fiacre)

K_i explorations et vérifications

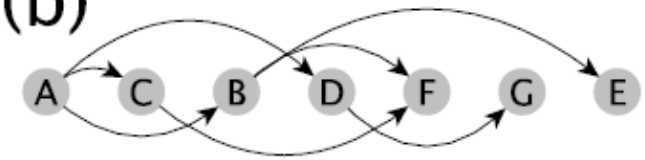
Explorateur

Optimisation des explorations
[PastFree Reachability Algorithm]

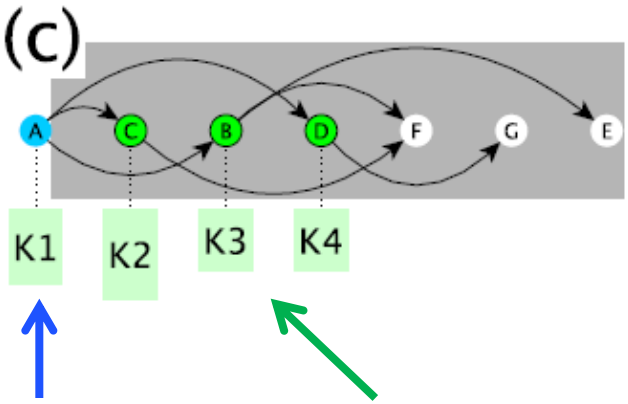
(a)



(b)



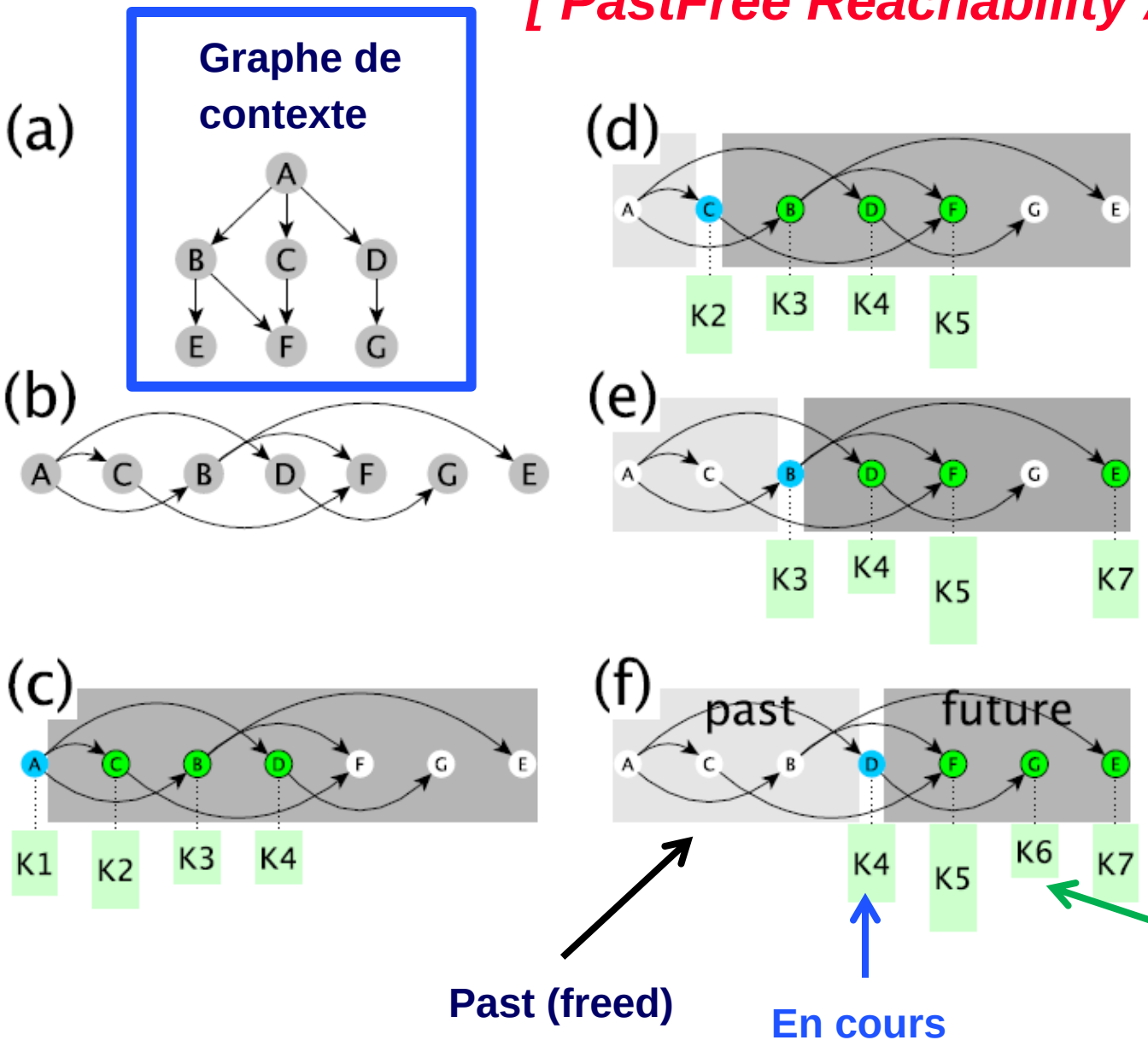
(c)



En cours

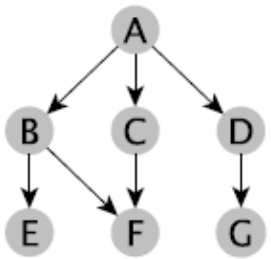
Atteignables

Optimisation des explorations
[PastFree Reachability Algorithm]

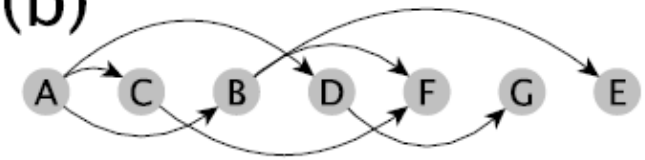


Optimisation des explorations
[PastFree Reachability Algorithm]

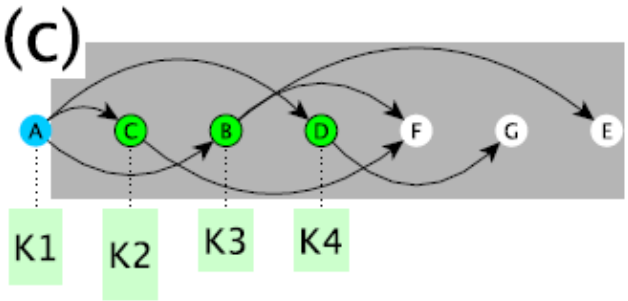
Graphe de contexte



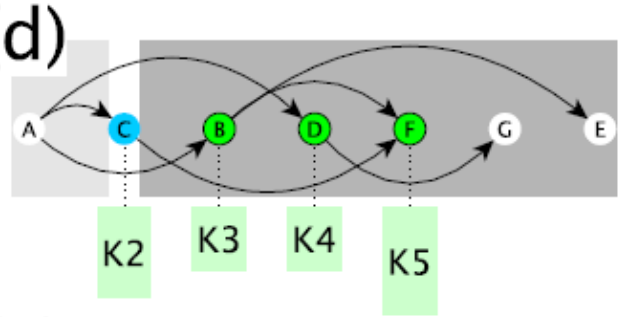
(a)



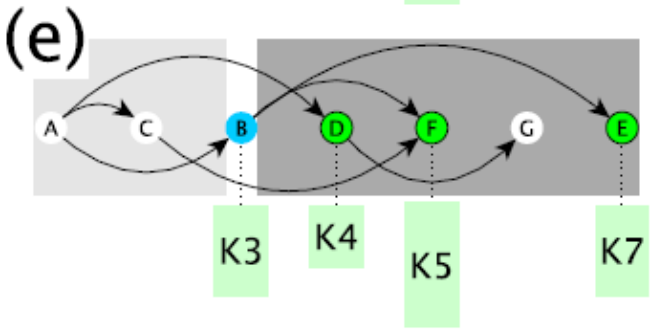
(b)



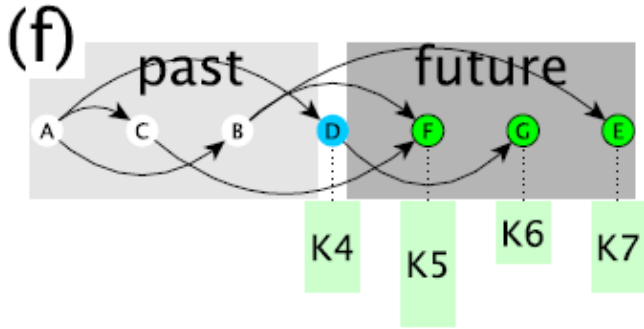
(c)



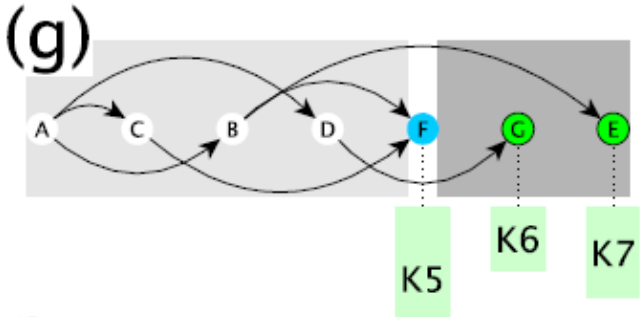
(d)



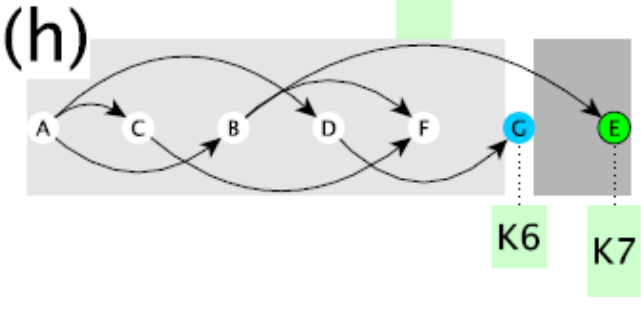
(e)



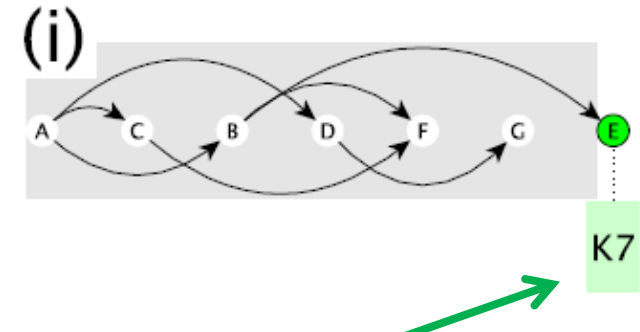
(f)



(g)

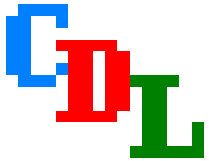


(h)



(i)

Atteignables

Langage 

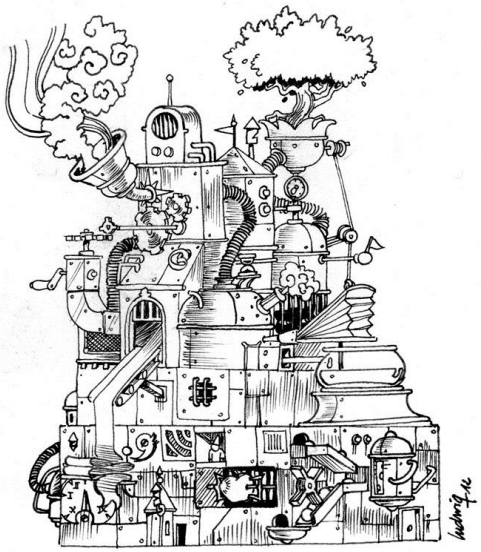
- ***Spécification des contextes***
- ***Spécification des propriétés***

Type et expression des propriétés

Disposer de primitives élémentaires d'observation, à grain fin

- **Invariants :** expression basée sur des prédicats

  Valeur d'une variable, Etat d'un processus, Etat d'une fifo



Type et expression des propriétés

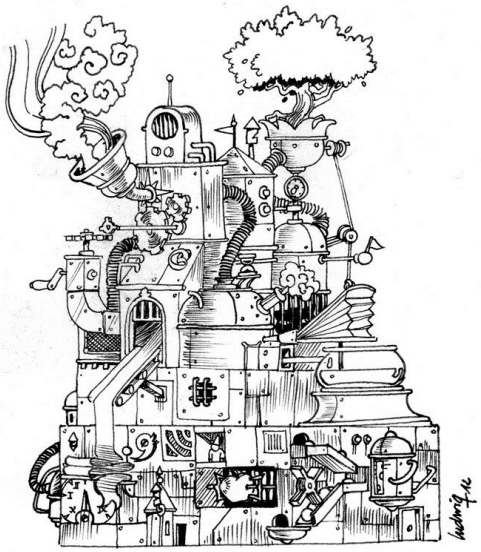
Disposer de primitives élémentaires d'observation, à grain fin

- **Invariants :** expression basée sur des prédicats

  Valeur d'une variable, Etat d'un processus, Etat d'une fifo

- **Observateur :** expression basée sur des patrons de définition de propriétés :

  Réponse, Précédence, Absence, Existence



Extension des patrons de Dwyer…

Type et expression des propriétés

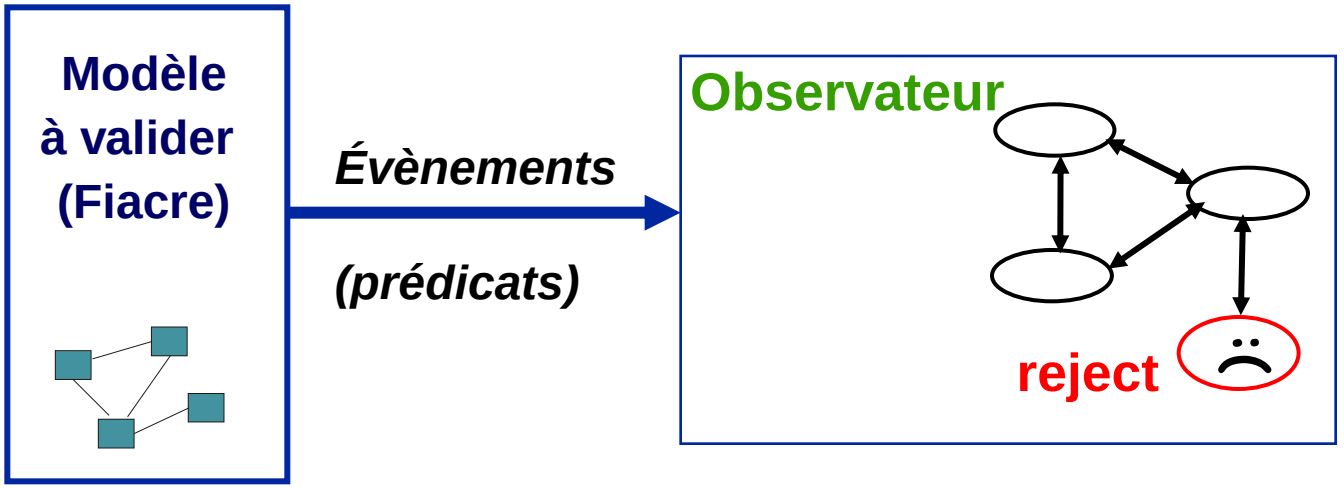
Disposer de primitives élémentaires d'observation, à grain fin

- **Invariants :** expression basée sur des prédicats

  Valeur d'une variable, Etat d'un processus, Etat d'une fifo

- **Observateur :** expression basée sur des patrons de définition de propriétés :

  Réponse, Précédence, Absence, Existence



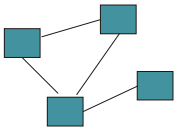
OBP : noyau d'observation

Explorer

1

exploration

Modèle à valider (Fiacre)



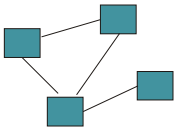
OBP : noyau d'observation

Explorer

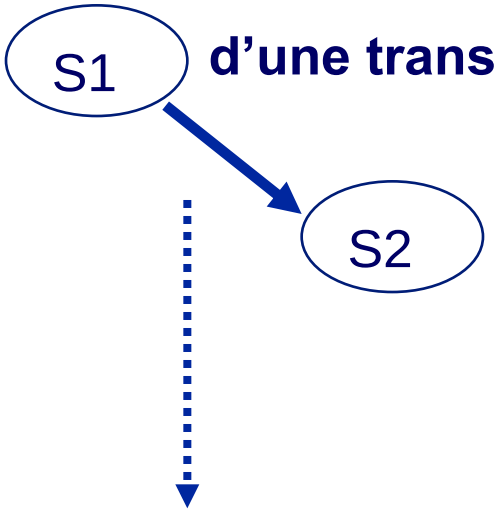
1

exploration

Modèle à valider (Fiacre)



Exécution d'une transition



Liste :

- **Valeurs de prédicats**
- **Évènements**

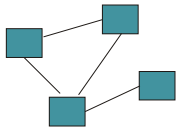
OBP : noyau d'observation

Explorer

1

exploration

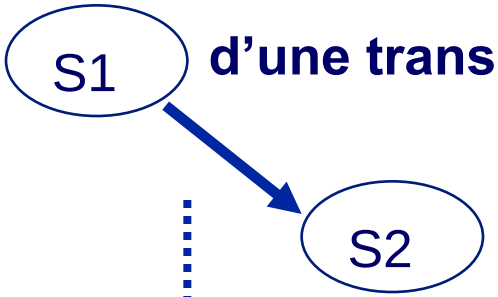
Modèle à valider (Fiacre)



2

observation

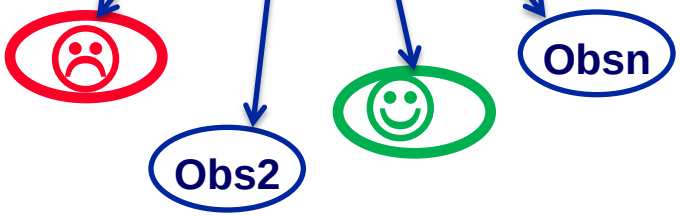
Exécution d'une transition



Liste :

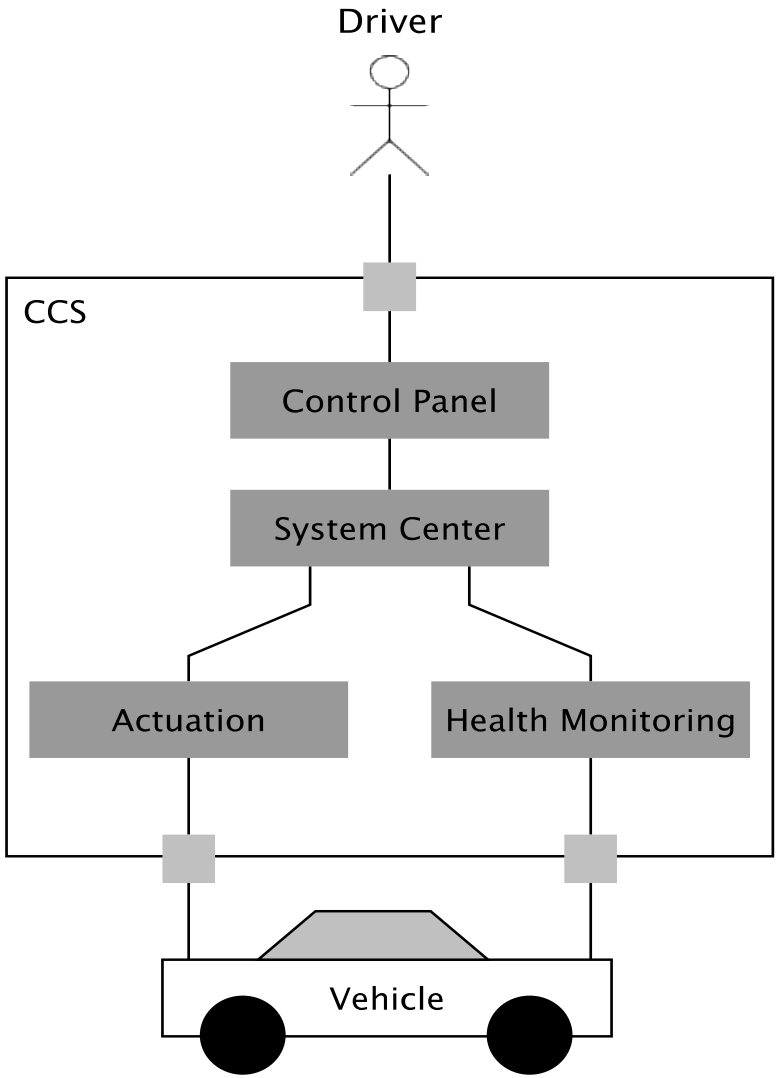
- Valeurs de prédicats
- Évènements

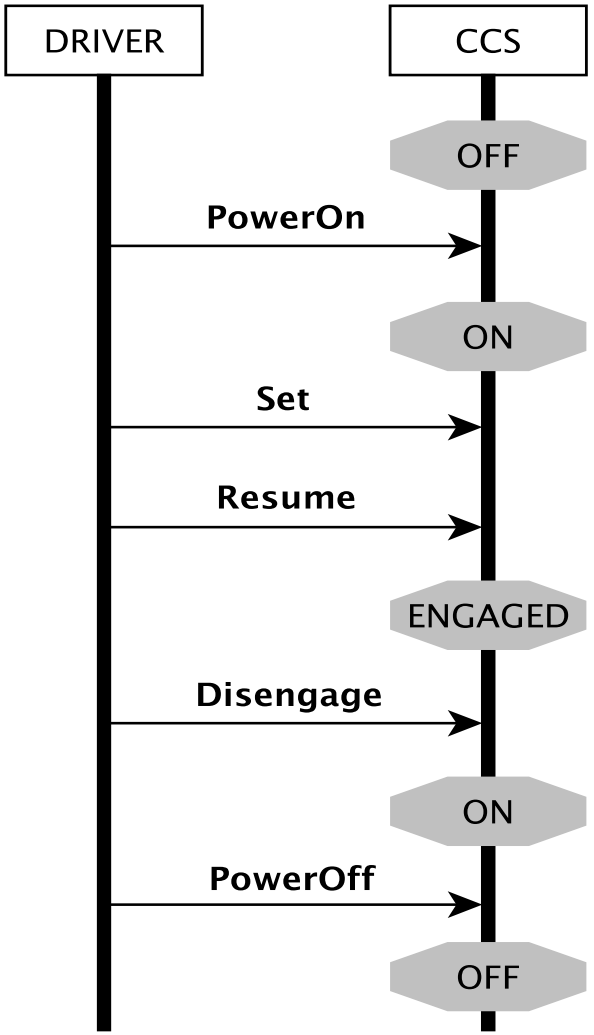
Machine d'observation



Ensemble d'observateurs

Cruise Control System (CCS)





Exigences

- ***Req1***: ***Après*** *la détection d'un événement induisant un désengagement et* ***tant que*** *le système n'est pas réengagé, le CCS* ***ne doit pas*** *tenter d'ajuster la vitesse du véhicule.*

- ***Req2***: *La vitesse de croisière* ***ne doit pas*** *être inférieure à 40km/h ou supérieure à 180km/h.*

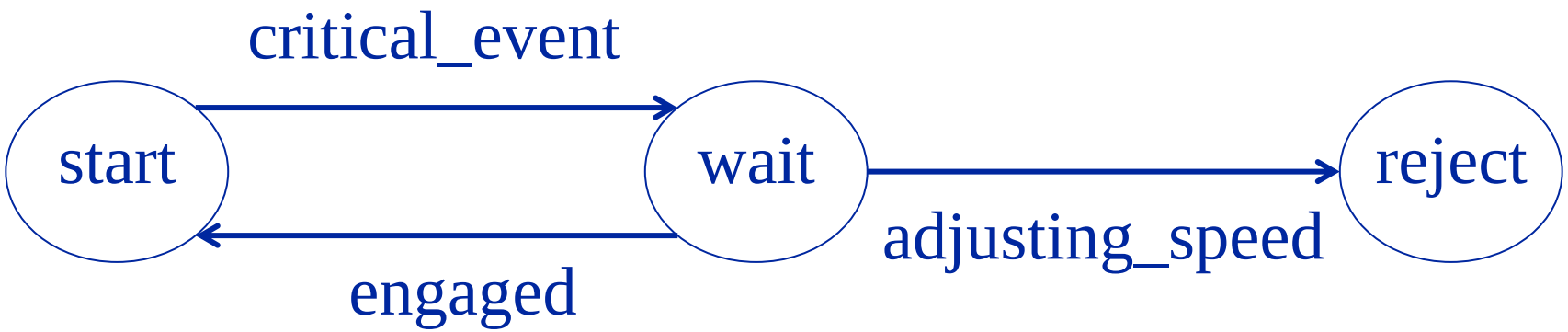
- ***Req3***: ***Tant que*** *le système est engagé, la vitesse cible* ***doit être*** *considérée comme définie.*

Exigence Req1 (CDL)

- ***Req1***: ***Après*** *la détection d'un événement induisant un désengagement et* ***tant que*** *le système n'est pas réengagé, le CCS* ***ne doit pas*** *tenter d'ajuster la vitesse du véhicule.*

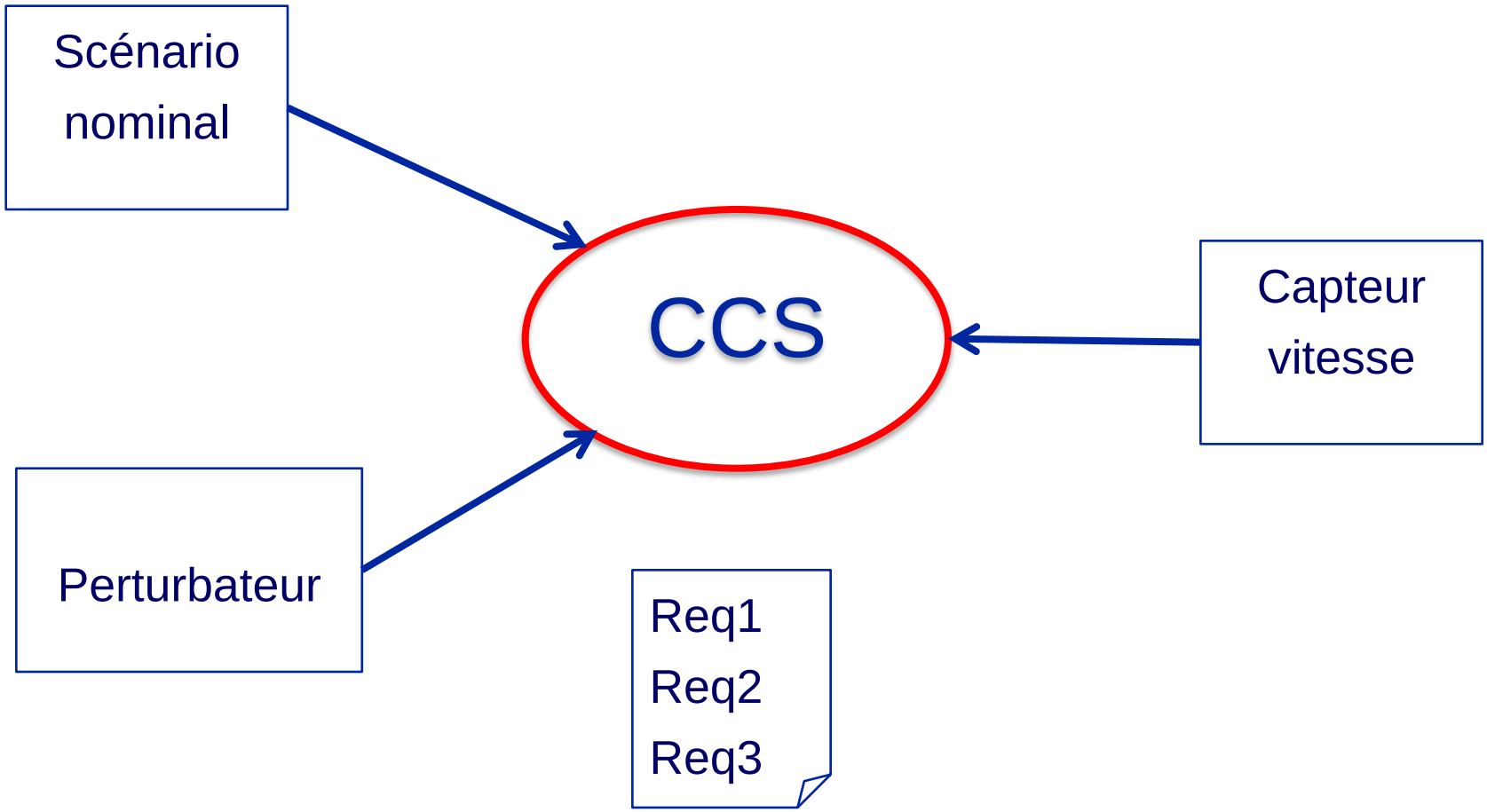
```
predicate disengageIsRequested is
                { HealthMonitoring@DisengageRequested }
event critical_event is
                { disengageIsRequested becomes true }

event adjusting_speed is
                { send any from Actuation to Car }

event engaged is
                { SystemCenter@Engaged becomes true }
```

Propriété Req1 (CDL)




```
property Req1 is
{
   start    -- / / critical_event / --> wait;
   wait     -- / / adjusting_speed / --> reject;
   wait     -- / / engaged / --> start
}
```

Spécification du contexte

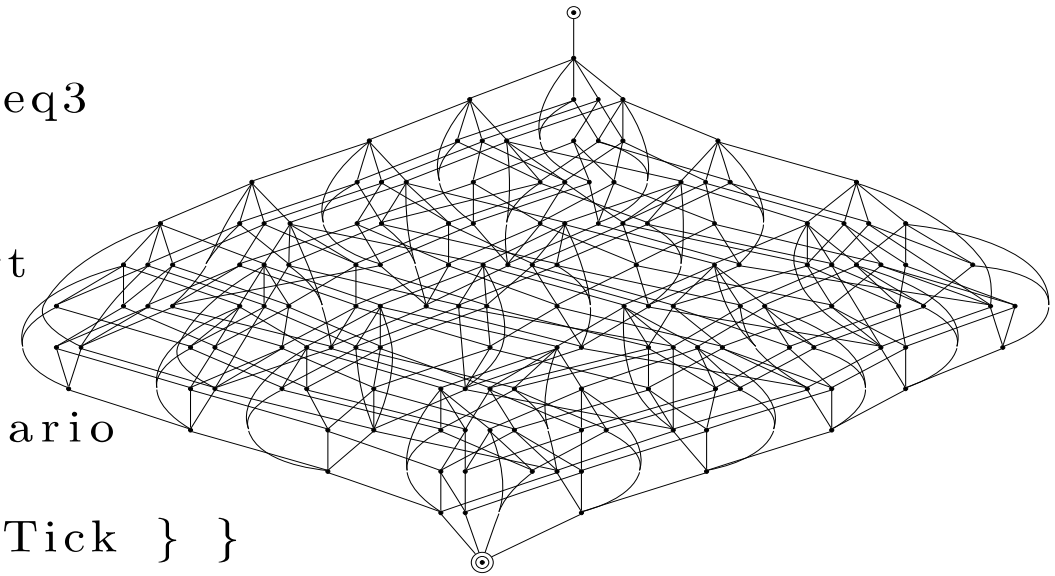


Spécification du contexte (CDL)

```
activity basic_scenario is {
        regular_speed_x2;
        evtBtnSet;
        evtBtnResume;
        regular_speed_x2;
        evtBtnDisengage }
```

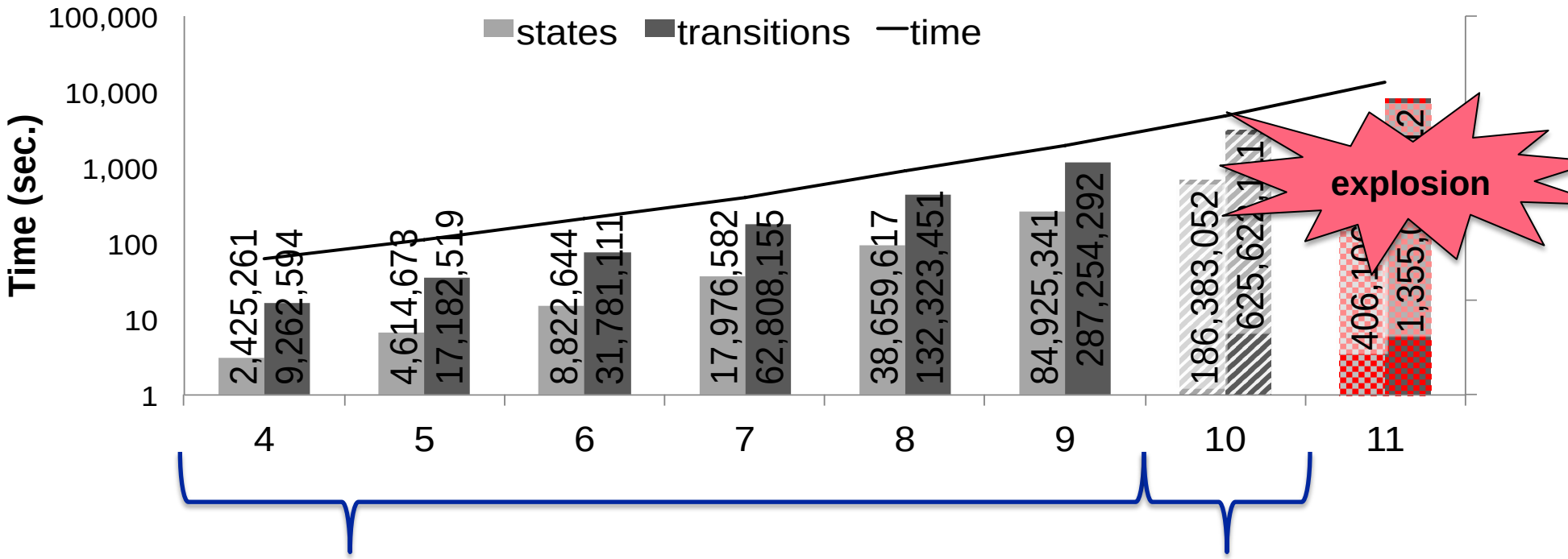
```
activity pertubator is {
        regular_speed_x2
[]      dangerous_speed_x1
[]      pedals_x2
[]      buttons_x3 }
```

```
cdl myContext is {
        properties req1, req3
        assert req2
        init is {
                evtBtnStart
        }
        main is {
                basic_scenario
        ||      pertubator
        ||      loop 4 evtTick } }
```



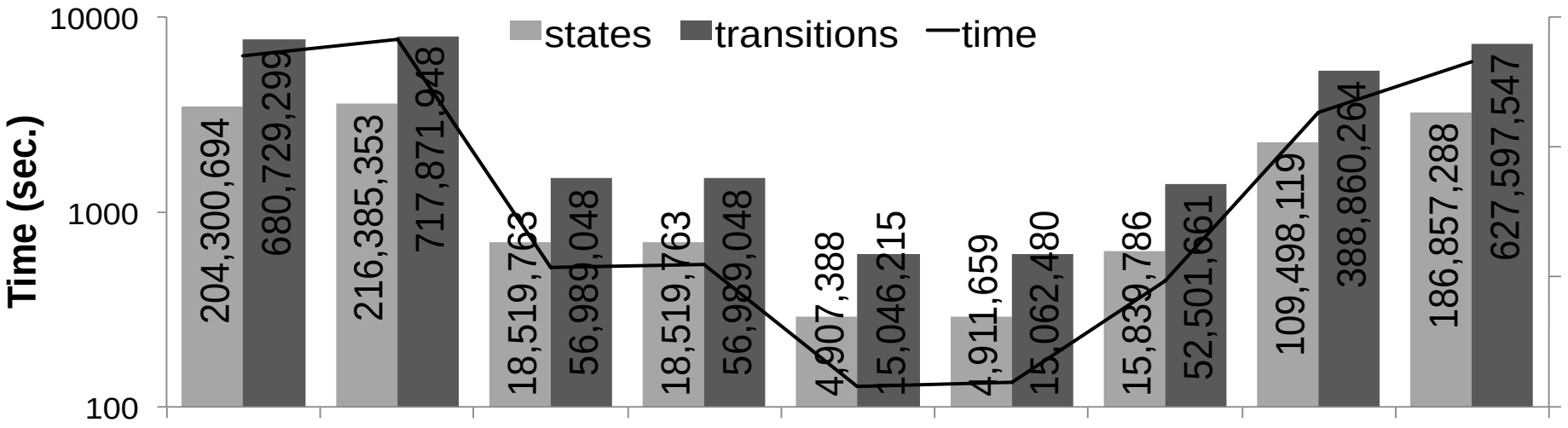
Résultats des explorations

Résultat des explorations pour différents contextes (de 4 à 11 *ticks*).



Résultats avec splitting

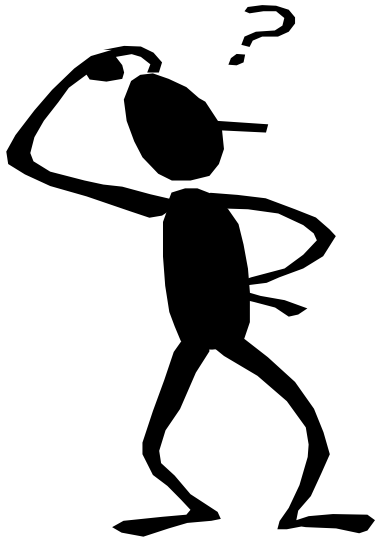
Génération automatique de 9 partitions pour un contexte dans le cas de 11 *ticks*.



Conclusion

- Vérification des 3 propriétés
- 2 stratégies d'optimisation efficaces (PFR et splitting)
- Méthodologie à définir
- Analyse des retours de preuve : aide au diagnostic

Merci de vos questions



www.obpcdl.org