Analyse de causalité pour l'attribution de responsabilité dans les systèmes à composants

Gregor Goessler

Equipe-projet SPADES

INRIA

Séminaire In'tech

26 Novembre 2015

Motivation

Besoin de techniques pour analyser les causes de dysfonctionnements :

- complexité croissante des systèmes critiques, multitude de contributeurs de composants (sous traitance, ...)
- l'analyse manuelle est longue et coûteuse
- le manque de techniques pour analyser la **responsabilité** est un obstacle au déploiement des systèmes critiques complexes (voitures autonomes, appareils médicaux interconnectés)

Motivation

Besoin de techniques pour analyser les causes de dysfonctionnements :

- complexité croissante des systèmes critiques, multitude de contributeurs de composants (sous traitance, ...)
- l'analyse manuelle est longue et coûteuse
- le manque de techniques pour analyser la **responsabilité** est un obstacle au déploiement des systèmes critiques complexes (voitures autonomes, appareils médicaux interconnectés)

Objectif : développer une analyse de causalité pour déterminer automatiquement les composants responsables d'un dysfonctionnement.

Projet LISE : Liability Issues in Software Engineering

2008-2011

Objectif : fournir un ensemble de méthodes et d'outils informatiques et juridiques permettant aux parties

- de définir précisément leurs **responsabilités** respectives pour des dysfonctionnements de systèmes logiciels
- de préciser les **éléments de preuve** qui pourront être utilisés pour déterminer ces responsabilités
- d'**établir ces responsabilités** en cas de dysfonctionnement

Périmètre :

- cadre contractuel
- responsabilités liées aux dysfonctionnements des logiciels

Projet LISE : Liability Issues in Software Engineering

2008-2011

Approche proposée :

1. définition précise et non ambigüe des responsabilités
2. implémentation d'une infrastructure de log garantissant la disponibilité de toutes les informations nécessaires pour établir les responsabilités
3. implémentation d'un analyseur de logs

Difficultés à définir les responsabilités en cas de dysfonctionnement :

- Juridico-techniques : liens entre fautes, dommage et responsabilités des acteurs

Projet LISE : Liability Issues in Software Engineering

2008-2011

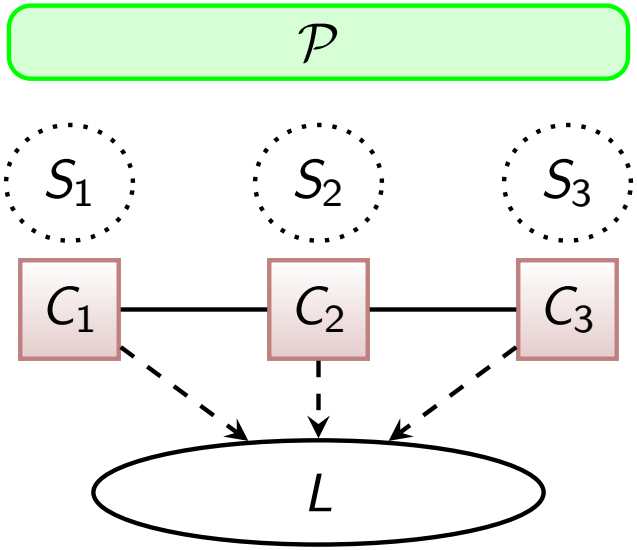
Approche proposée :

1. définition précise et non ambigüe des responsabilités
2. implémentation d'une infrastructure de log garantissant la disponibilité de toutes les informations nécessaires pour établir les responsabilités
3. implémentation d'un analyseur de logs

Difficultés à définir les responsabilités en cas de dysfonctionnement :

- Juridico-techniques : liens entre fautes, dommage et responsabilités des acteurs

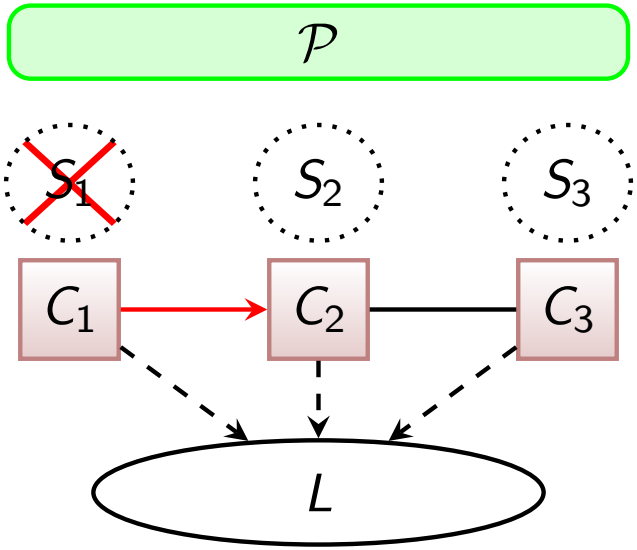
Enoncé du problème



Donné :

- un système de n composants avec spécifications $S_1, ..., S_n$
- un log L
- une propriété de sûreté $\mathcal{P}$ requise

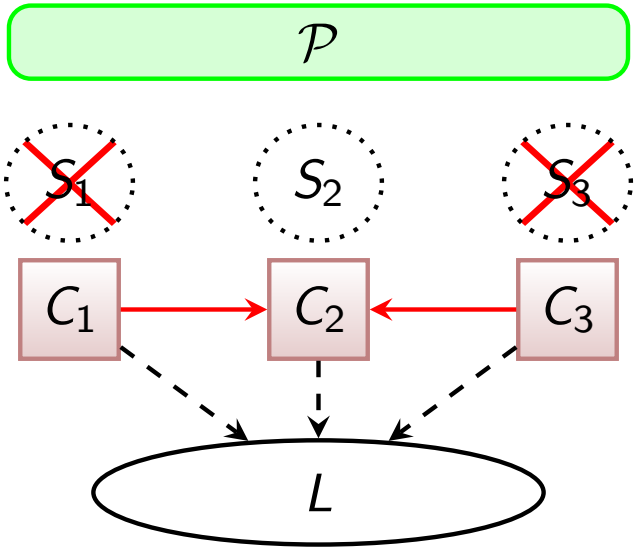
Enoncé du problème



Donné :

- un système de n composants avec spécifications $S_1, ..., S_n$
- un log L
- une propriété de sûreté $\mathcal{P}$ requise

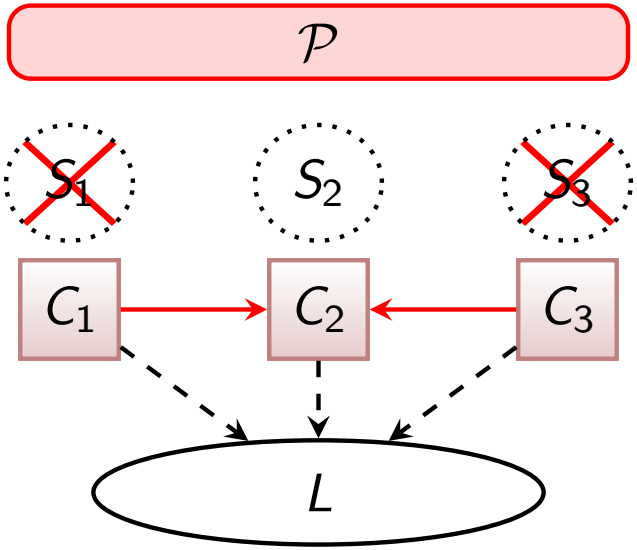
Enoncé du problème



Donné :

- un système de n composants avec spécifications $S_1, ..., S_n$
- un log L
- une propriété de sûreté $\mathcal{P}$ requise

Enoncé du problème

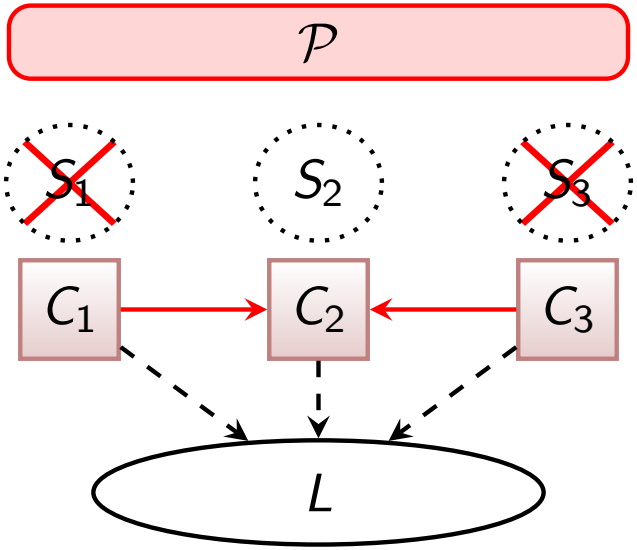


Donné :

- un système de n composants avec spécifications $S_1, ..., S_n$
- un log L
- une propriété de sûreté $\mathcal{P}$ requise qui est violée par L
- un sous ensemble X des fautes observées

Les fautes X sont-elles une cause pour la violation de $\mathcal{P}$ par L ?

Enoncé du problème



Donné :

- un système de n composants avec spécifications $S_1, ..., S_n$
- un log L
- une propriété de sûreté $\mathcal{P}$ requise qui est violée par L
- un sous ensemble X des fautes observées

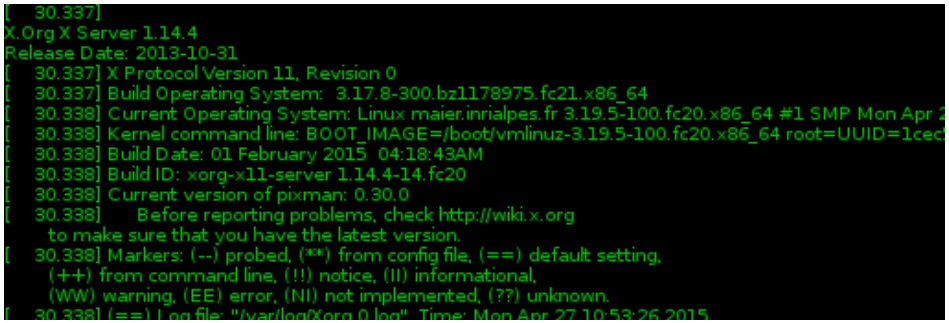
Les fautes X sont-elles une cause pour la violation de $\mathcal{P}$ par L ?

Hypothèse

Si tous les composants satisfont leur spécification alors $\mathcal{P}$ est satisfaite.

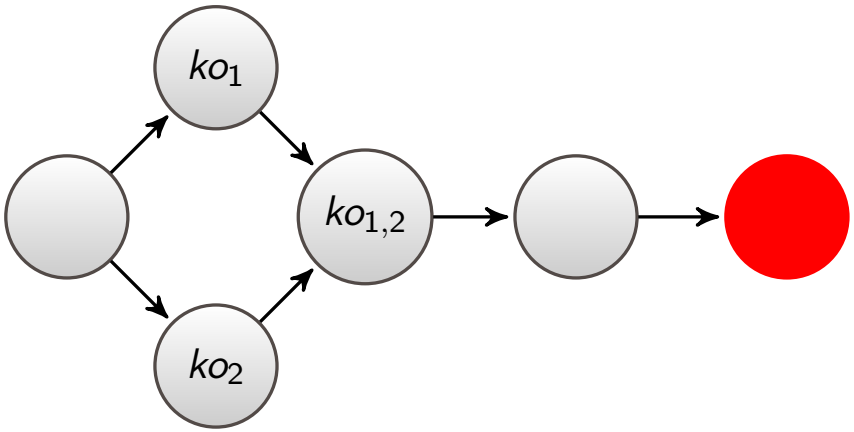
Log et trace d'exécution

Log :



Trace d'exécution = ensemble d'observations = abstraction d'un log.

Exemple d'une trace d'exécution d'un système réparti :



- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

Analyse de causalité par raisonnement contrefactuel

Trace **contrefactuelle** par rapport à un ensemble X de fautes observées $=$ trace où X ne se produit pas, toutes choses étant égales par ailleurs.

X est une **cause nécessaire** pour la violation de la propriété $\mathcal{P}$ si dans les traces contrefactuelles par rapport à X, $\mathcal{P}$ aurait été satisfaite.

Définition **formelle** des traces contrefactuelles ?

Analyse de causalité par raisonnement contrefactuel

Trace **contrefactuelle** par rapport à un ensemble X de fautes observées = trace où X ne se produit pas, toutes choses étant égales par ailleurs.

X est une **cause nécessaire** pour la violation de la propriété $\mathcal{P}$ si dans les traces contrefactuelles par rapport à X, $\mathcal{P}$ aurait été satisfaite.

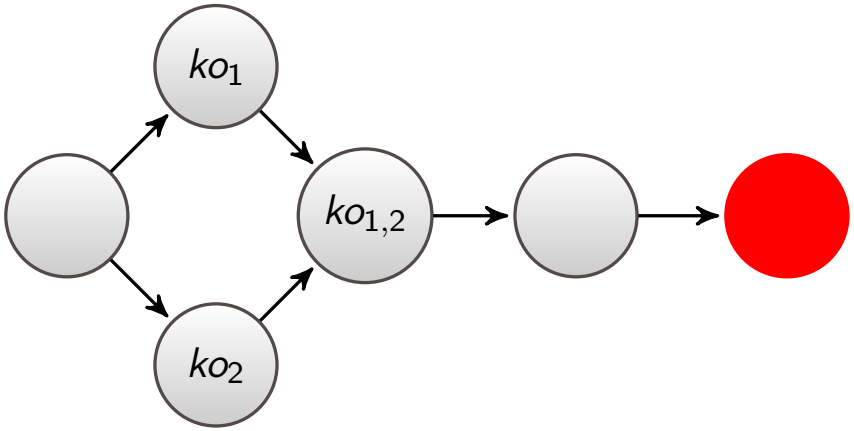
Définition **formelle** des traces contrefactuelles ?

Difficultés :

- construire des traces **proches** de la trace observée
- la correction des fautes X peut impacter le comportement des autres composants : éviter les **incohérences**
- la chaîne de causalité des fautes à une violation de $\mathcal{P}$ peut être d'une longueur importante, inconnue
- nombre important (voire infini) de scénarios, en général

Analyse de causalité par raisonnement contrefactuel

Principe

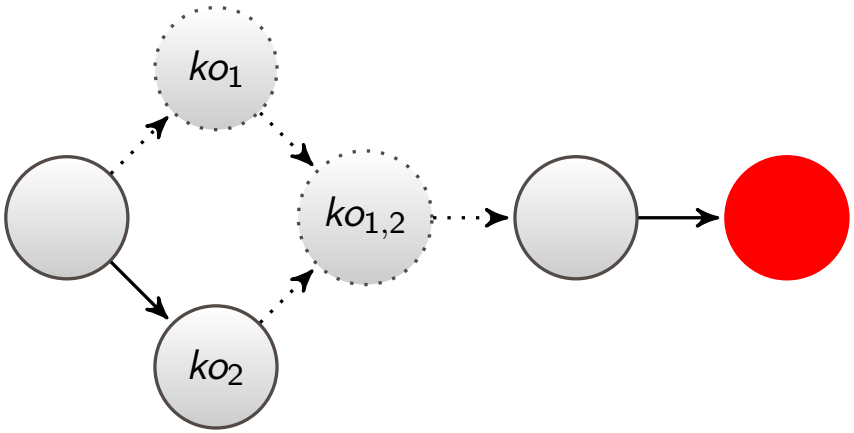


- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

Analyse de **causalité des fautes** X (ici $X = ko_1$) en les "corrigeant" :

Analyse de causalité par raisonnement contrefactuel

Principe



- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

Analyse de **causalité des fautes** X (ici $X = ko_1$) en les "corrigeant" :

1. supprimer X

Analyse de causalité par raisonnement contrefactuel

Principe

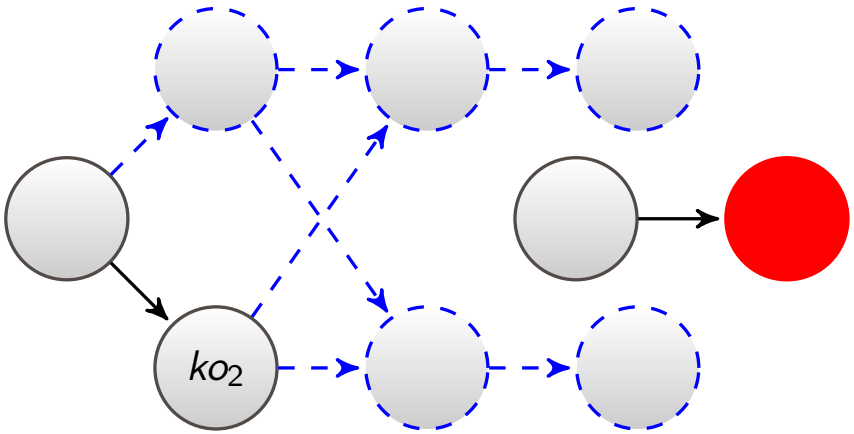
- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

Analyse de **causalité des fautes** X (ici $X = ko_1$) en les "corrigeant" :

1. supprimer X

Analyse de causalité par raisonnement contrefactuel

Principe



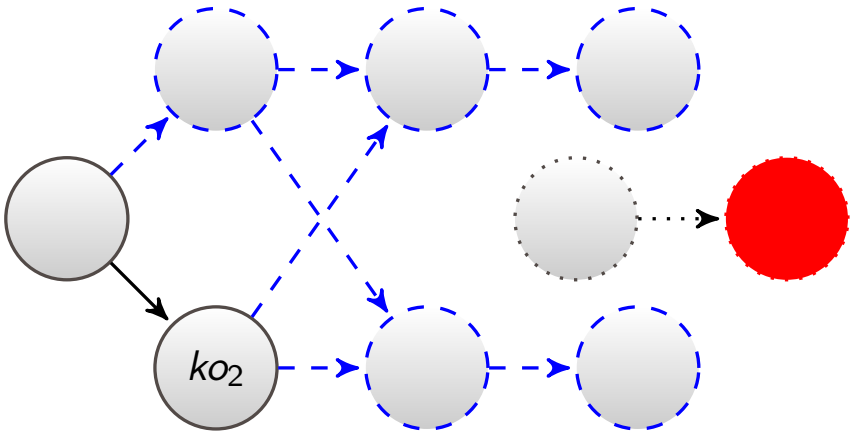
- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

Analyse de **causalité des fautes** X (ici $X = ko_1$) en les "corrigeant" :

1. supprimer X
2. substituer par les comportements attendus (ici : donnés par $\mathcal{S}_1$)

Analyse de causalité par raisonnement contrefactuel

Principe



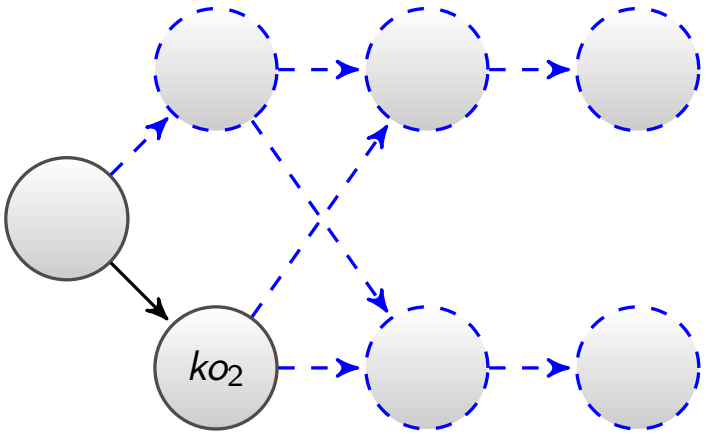
- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

Analyse de **causalité des fautes** X (ici $X = ko_1$) en les "corrigeant" :

1. supprimer X
2. substituer par les comportements attendus (ici : donnés par $\mathcal{S}_1$)
3. supprimer les observations qui n'ont plus d'explication

Analyse de causalité par raisonnement contrefactuel

Principe



- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

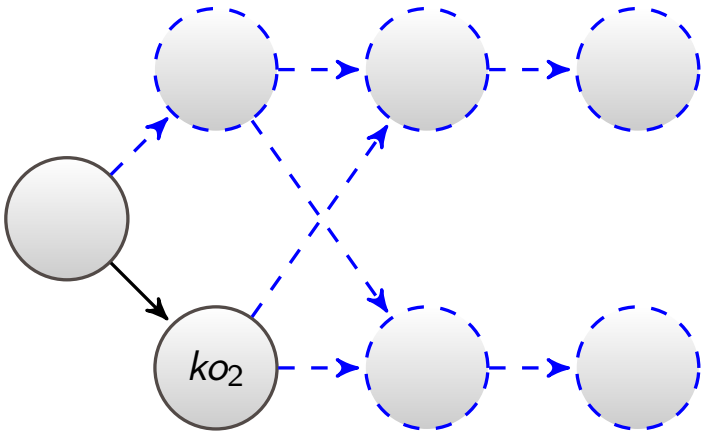
Analyse de **causalité des fautes** X (ici $X = ko_1$) en les "corrigeant" :

1. supprimer X
2. substituer par les comportements attendus (ici : donnés par $\mathcal{S}_1$)
3. supprimer les observations qui n'ont plus d'explication

X est une **cause nécessaire** pour la violation de $\mathcal{P}$ si sur les traces contrefactuelles, $\mathcal{P}$ est satisfaite.

Analyse de causalité par raisonnement contrefactuel

Principe



- $\rightarrow$: ordre partiel entre observations
- ko_i : faute du composant i
- ● : violation de $\mathcal{P}$ = dysfonctionnement du système
- explication d'une observation q = exécution possible menant à q

Analyse de **causalité des fautes** X (ici $X = ko_1$) en les "corrigeant" :

1. supprimer X
2. substituer par les comportements attendus (ici : donnés par $\mathcal{S}_1$)
3. supprimer les observations qui n'ont plus d'explication

X est une **cause nécessaire** pour la violation de $\mathcal{P}$ si sur les traces contrefactuelles, $\mathcal{P}$ est satisfaite.

X est une **cause suffisante** pour la violation de $\mathcal{P}$ si après correction de toutes les fautes sauf X, $\mathcal{P}$ est toujours violée.

Analyse de causalité par raisonnement contrefactuel

Difficultés (revisitées) :

- construire des traces **proches** de la trace observée
- la correction des fautes X peut impacter le comportement des autres composants : éviter les **incohérences**
- la chaîne de causalité des fautes à une violation de $\mathcal{P}$ peut être d'une longueur importante, inconnue
- nombre important (voire infini) de scénarios, en général

Analyse de causalité par raisonnement contrefactuel

Difficultés (revisitées) :

- construire des traces **proches** de la trace observée
  $\rightarrow$ préservation de toutes les observations ayant une explication
- la correction des fautes X peut impacter le comportement des autres composants : éviter les **incohérences**
- la chaîne de causalité des fautes à une violation de $\mathcal{P}$ peut être d'une longueur importante, inconnue
- nombre important (voire infini) de scénarios, en général

Analyse de causalité par raisonnement contrefactuel

Difficultés (revisitées) :

- construire des traces **proches** de la trace observée
  $\rightarrow$ préservation de toutes les observations ayant une explication
- la correction des fautes X peut impacter le comportement des autres composants : éviter les **incohérences**
  $\rightarrow$ suppression des observations sans explication
- la chaîne de causalité des fautes à une violation de $\mathcal{P}$ peut être d'une longueur importante, inconnue
- nombre important (voire infini) de scénarios, en général

Analyse de causalité par raisonnement contrefactuel

Difficultés (revisitées) :

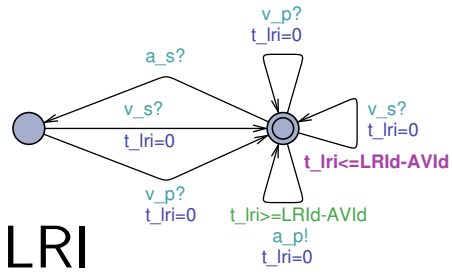
- construire des traces **proches** de la trace observée
  $\rightarrow$ préservation de toutes les observations ayant une explication
- la correction des fautes X peut impacter le comportement des autres composants : éviter les **incohérences**
  $\rightarrow$ suppression des observations sans explication
- la chaîne de causalité des fautes à une violation de $\mathcal{P}$ peut être d'une longueur importante, inconnue
  $\rightarrow$ exploration de toutes les continuations possibles selon les spécifications
- nombre important (voire infini) de scénarios, en général

Analyse de causalité par raisonnement contrefactuel

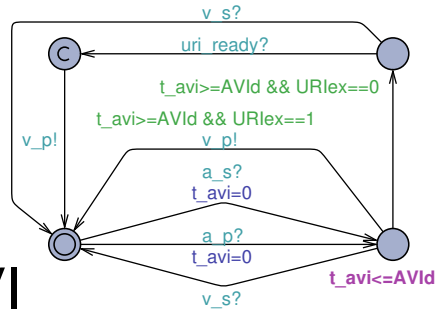
Difficultés (revisitées) :

- construire des traces **proches** de la trace observée
  $\rightarrow$ préservation de toutes les observations ayant une explication
- la correction des fautes X peut impacter le comportement des autres composants : éviter les **incohérences**
  $\rightarrow$ suppression des observations sans explication
- la chaîne de causalité des fautes à une violation de $\mathcal{P}$ peut être d'une longueur importante, inconnue
  $\rightarrow$ exploration de toutes les continuations possibles selon les spécifications
- nombre important (voire infini) de scénarios, en général
  $\rightarrow$ raisonnement sur une représentation symbolique finie

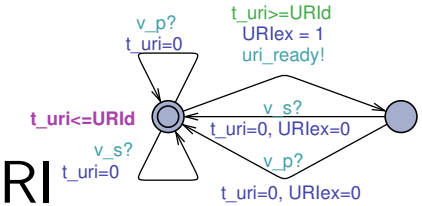
Cas d'étude : Pacemaker [Pajic et al., 2012]



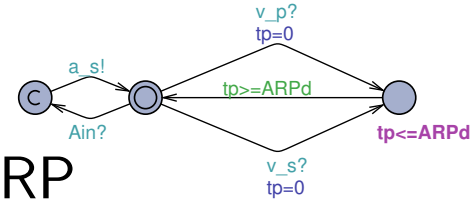
LRI



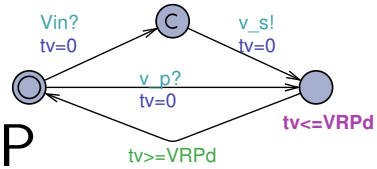
AVI



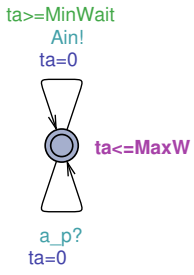
URI



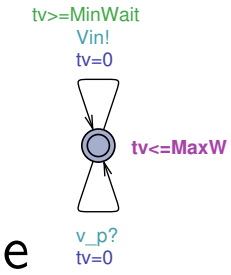
PVARP



VRP

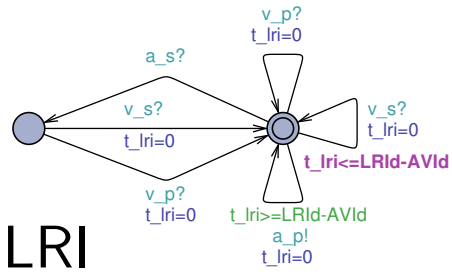


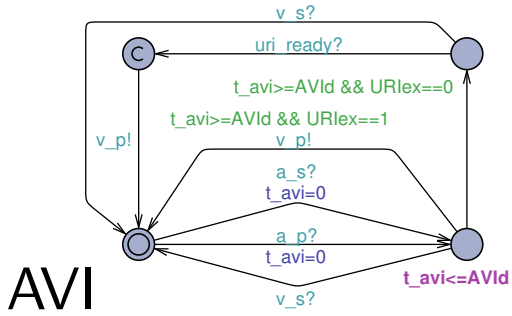
atrium

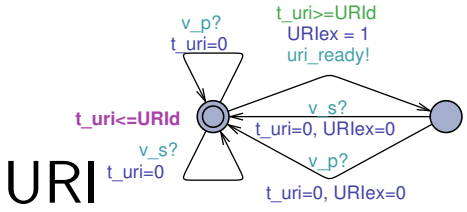


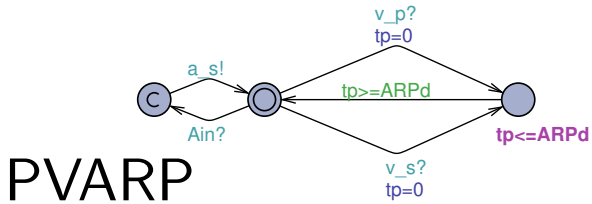
ventricle

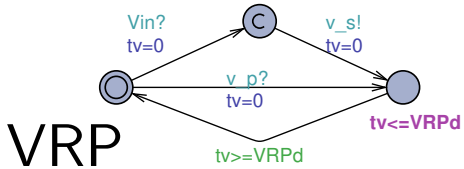
Cas d'étude : Pacemaker [Pajic et al., 2012]



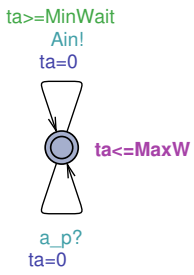


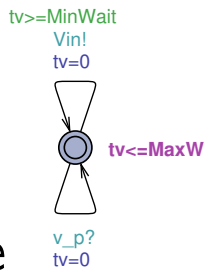




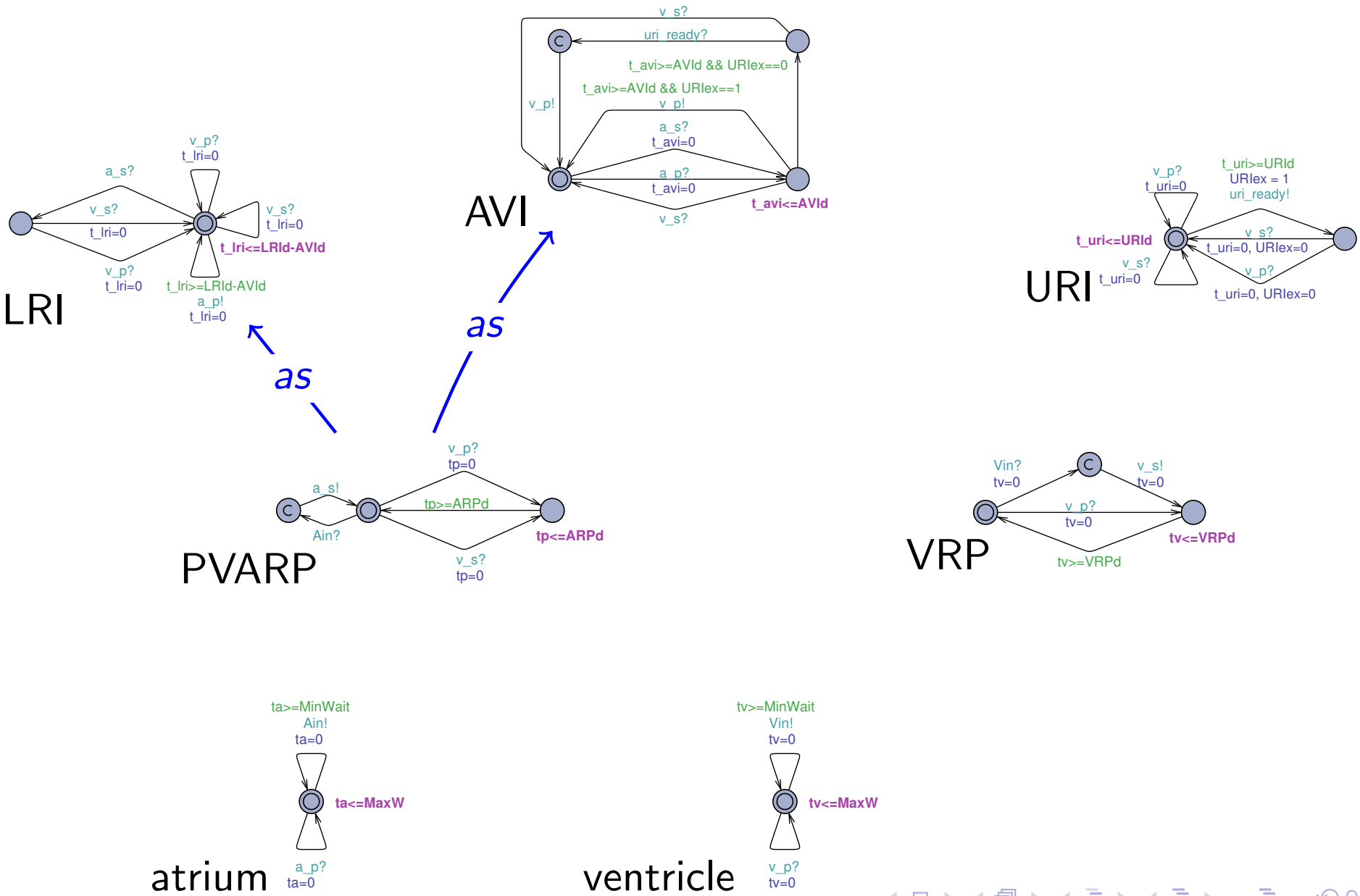


Ain

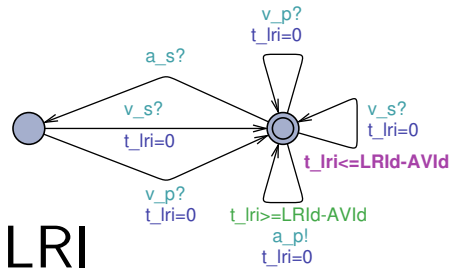


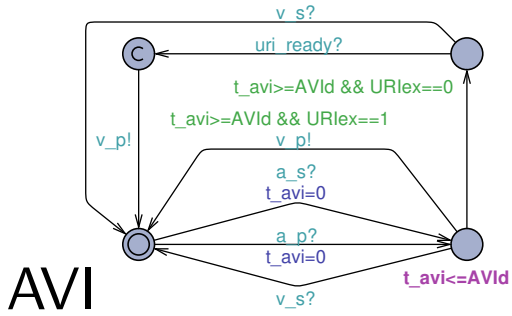


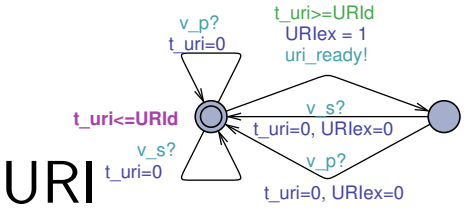
Cas d'étude : Pacemaker [Pajic et al., 2012]

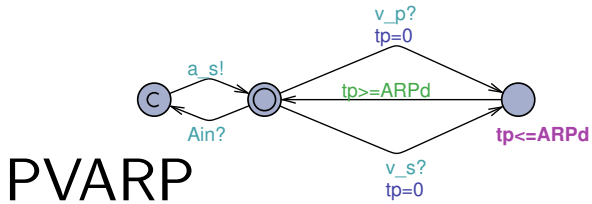


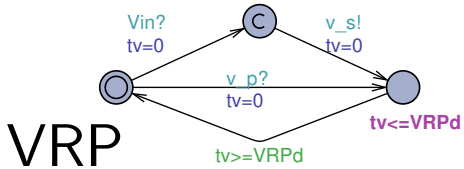
Cas d'étude : Pacemaker [Pajic et al., 2012]



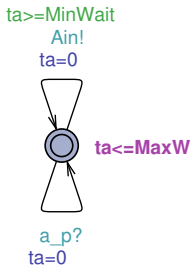


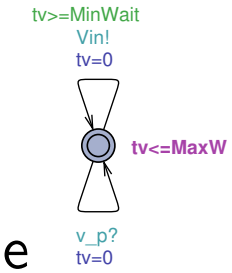




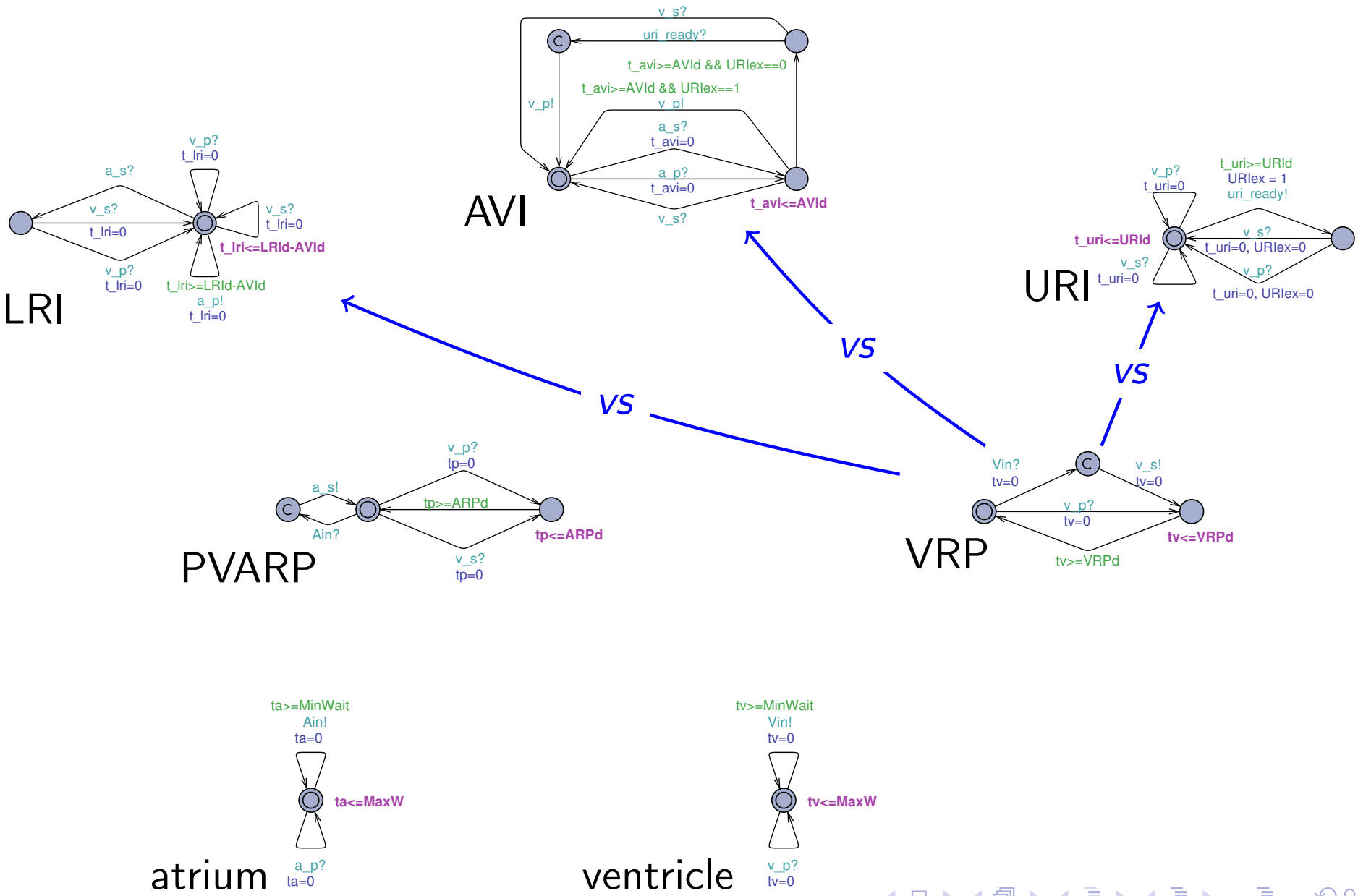


Vin

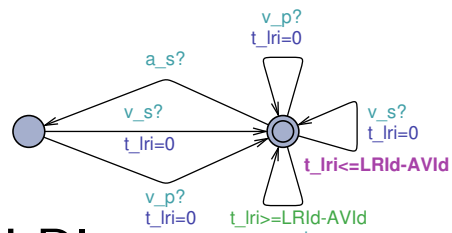




Cas d'étude : Pacemaker [Pajic et al., 2012]

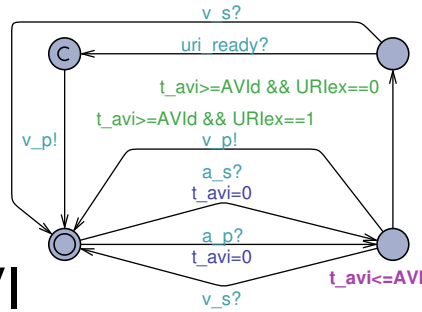


Cas d'étude : Pacemaker [Pajic et al., 2012]

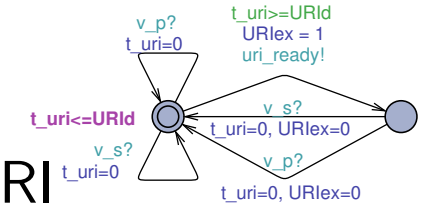


LRI

ap →

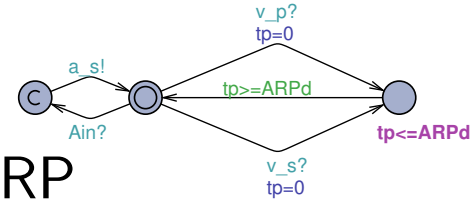


AVI

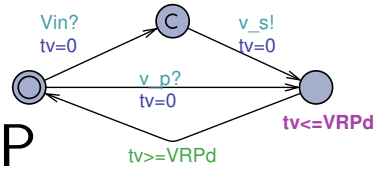


URI

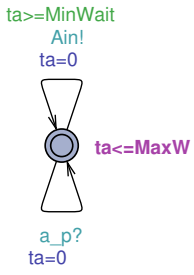
ap



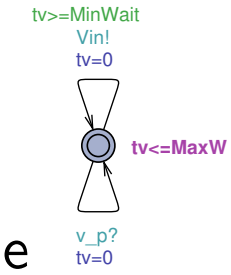
PVARP



VRP

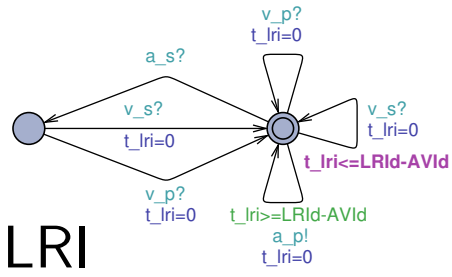


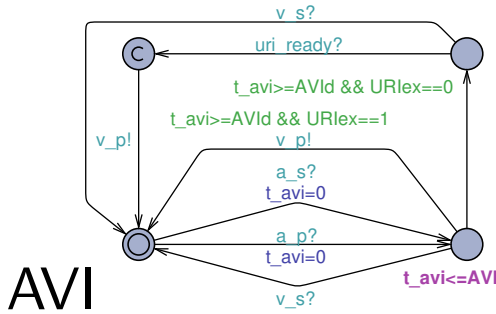
atrium



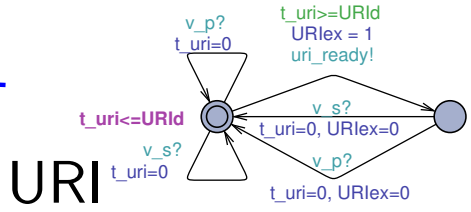
ventricle

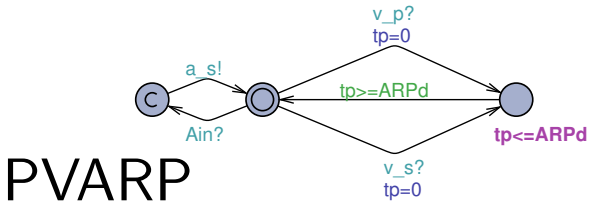
Cas d'étude : Pacemaker [Pajic et al., 2012]

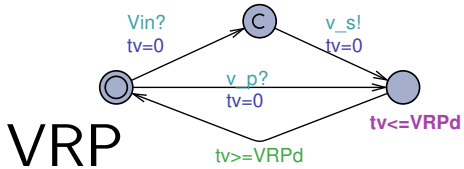


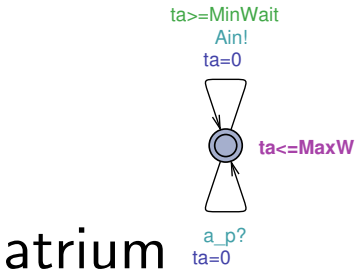


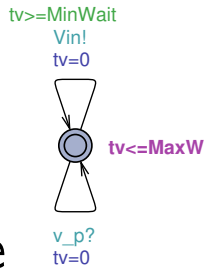
urirdy



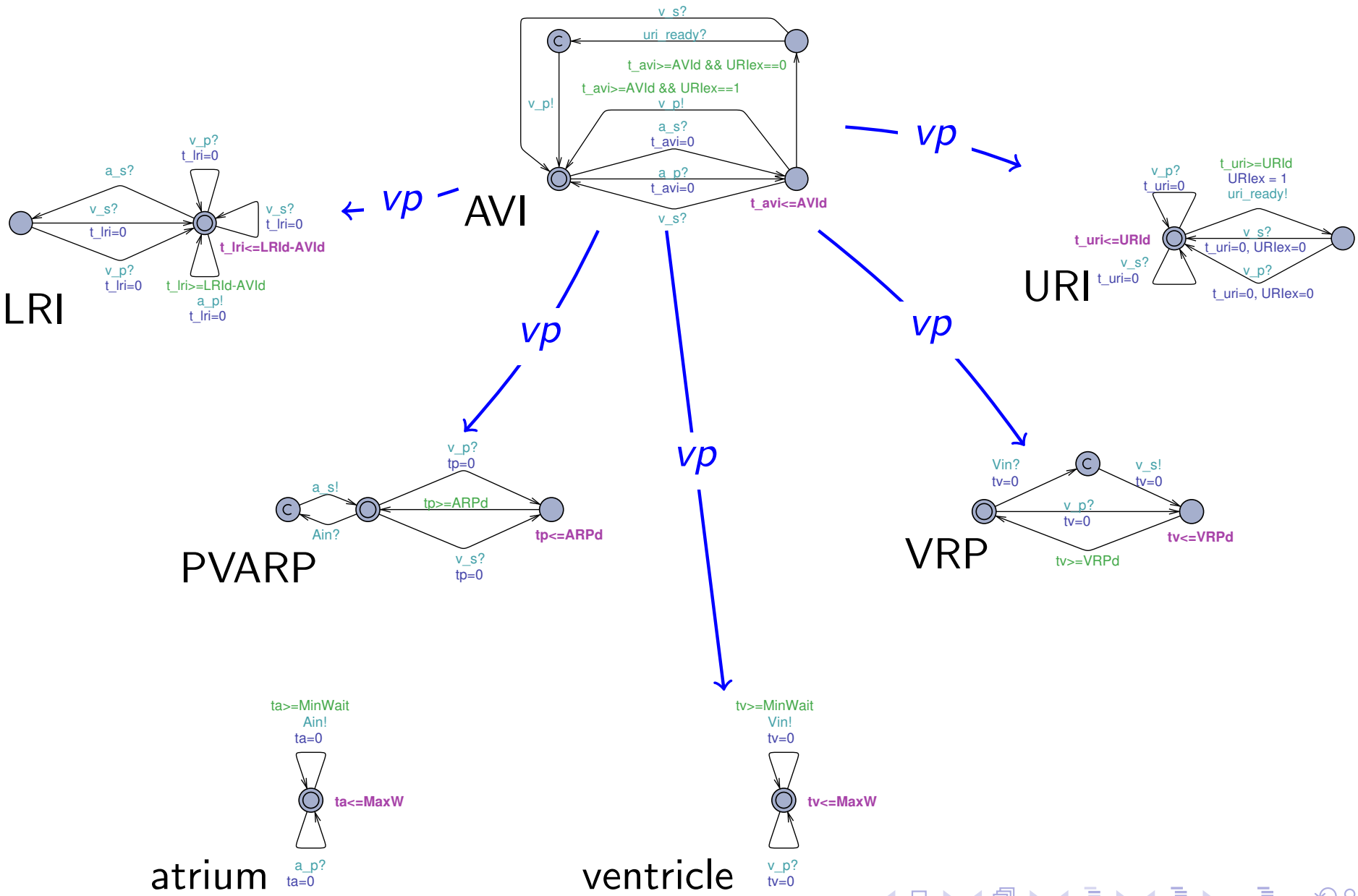




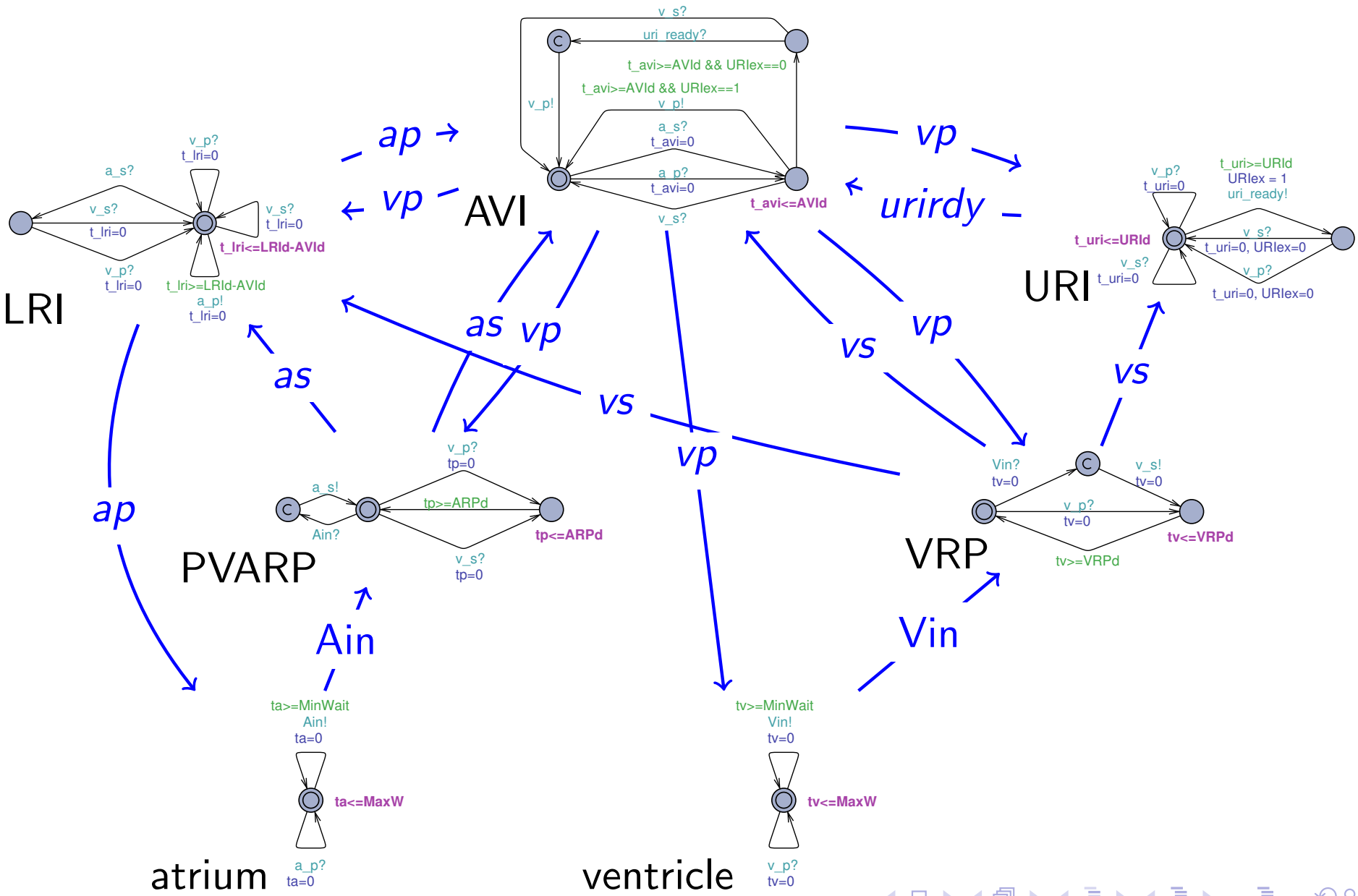




Cas d'étude : Pacemaker [Pajic et al., 2012]



Cas d'étude : Pacemaker [Pajic et al., 2012]



Cas d'étude : Pacemaker

Propriété de sûreté $\mathcal{P}$:

1. pas de stimulation auriculaire pendant $850ms$ après $\{\text{Vin}!, \text{vp}!\}$;
2. pas de stimulation ventriculaire pendant $400ms$ après $\{\text{Vin}!, \text{vp}!\}$.

atrium	500	Ain !		100			200	ap ?	150		
ventricle	500			100	Vin !		200		150		vp ?
AVI	500		as ?	100			200		150	urirdy ?	vp !
LRI	500		as ?	100		vs ?	200	**ap !**	150		
URI	**500**			100		vs ?	200		150	urirdy !	
PVARP	500	Ain ?	as !	100		vs ?	200		150		vp ?
VRP	500			100	Vin ?	vs !	200		150		vp ?

2 fautes de composants :

- LRI : *ap*! précoce ;
- URI : *urirdy*! manquant à $400ms$.

Cas d'étude : Pacemaker

Propriété de sûreté $\mathcal{P}$:

1. pas de stimulation auriculaire pendant $850ms$ après $\{\text{Vin}!, \text{vp}!\}$;
2. pas de stimulation ventriculaire pendant $400ms$ après $\{\text{Vin}!, \text{vp}!\}$.

atrium	500	Ain !		100			200	ap ?	150		
ventricle	500			100	Vin !		200		150		vp ?
AVI	500		as ?	100			200		150	urirdy ?	vp !
LRI	500		as ?	100		vs ?	200	**ap !**	150		
URI	**500**			100		vs ?	200		150	urirdy !	
PVARP	500	Ain ?	as !	100		vs ?	200		150		vp ?
VRP	500			100	Vin ?	vs !	200		150		vp ?

2 fautes de composants :

- LRI : *ap*! précoce ;
- URI : *urirdy*! manquant à $400ms$.

Cas d'étude : Pacemaker

Propriété de sûreté $\mathcal{P}$:

1. pas de stimulation auriculaire pendant $850ms$ après $\{\text{Vin}!, \text{vp}!\}$;
2. pas de stimulation ventriculaire pendant $400ms$ après $\{\text{Vin}!, \text{vp}!\}$.

atrium	500	Ain !		100			200	ap ?	150
ventricle	500			100	Vin !		200		150
AVI	500		as ?	100			200		150
LRI	500		as ?	100		vs ?	200	**ap !**	150
URI	**400**								
PVARP	500	Ain ?	as !	100		vs ?	200		150
VRP	500			100	Vin ?	vs !	200		150

2 fautes de composants :

- LRI : *ap*! précoce ;
- URI : *urirdy*! manquant à $400ms$.

Cas d'étude : Pacemaker

Propriété de sûreté $\mathcal{P}$:

1. pas de stimulation auriculaire pendant $850ms$ après $\{\text{Vin}!, \text{vp}!\}$;
2. pas de stimulation ventriculaire pendant $400ms$ après $\{\text{Vin}!, \text{vp}!\}$.

atrium	500	Ain !		100			200	ap ?	150	...
ventricle	500			100	Vin !		200		150	...
AVI	500		as ?	100			200		150	
LRI	500		as ?	100		vs ?	200	**ap !**	150	...
URI	**400**	urirdy !	...							
PVARP	500	Ain ?	as !	100		vs ?	200		150	...
VRP	500			100	Vin ?	vs !	200		150	...

2 fautes de composants :

- LRI : *ap*! précoce ;
- URI : *urirdy*! manquant à $400ms$.

$\Rightarrow$ Seule la faute de LRI est une cause pour la violation de $\mathcal{P}$.

Conclusion

Contributions :

- Détermination des causes de dysfonctionnements
  - basée sur **une seule trace d'exécution**
  - **aucun accès aux composants** n'est requis
- Implémentation POC pour les systèmes
  - temps-réel
  - flot de données synchrone (Lustre)

Applications :

- aider à établir la **responsabilité** des fournisseurs de composants
- expliquer la violation d'une propriété par un contre-exemple (test, model-checking) lors de la **conception**

Perspectives :

- boîtes blanches (comportement des composants connu, pas de spécifications)
- analyse quantitative (p.ex. basée sur probabilités)

1st Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies

Dates importantes :

résumés : 10 janvier
papiers : 17 janvier
workshop : 8 avril, Eindhoven