PARDI
Verification of PARameterized DIstributed Systems
Revue de projet T0 + 20 mois

Philippe Quéinnec *et al.*

04/10/2018


Plan

PARDI – Verification of PARameterized DIstributed Systems

Partenaires

- Toulouse INP / IRIT (Philippe Quéinnec, Aurélie Hurault, Philippe Mauran, Meriem Ouederni)
- Univ. Paris Sud / LRI (Sylvain Conchon, Fatiha Zaïdi)
- INRIA Nancy (Stephan Merz, Marie Duflot-Kremer)
- Univ. Pierre et Marie Curie / LIP6 (Pascal Poizat, Souheib Baarir)

PARDI – Objectifs

Objectifs

Vérification mécanisée de systèmes distribués paramétrés

- Systèmes distribués asynchrones communiquant par messages
- Paramètres $\Rightarrow$ famille de systèmes
- Vérification formelle, automatique et assistée

Points durs :

- Multiplicité des paramètres : nombre de sites, nombre de fautes, relation entre paramètres ($\mathit{fautes} < \mathit{sites}/2$), modèles de communication (ordre de délivrance, structure du réseau...), modèles de fautes
- Asynchronisme
- Décidabilité vs expressivité

PARDI – résultats attendus

Résultats attendus (proposition)

- Vérificateur de modèles pour systèmes paramétrés
- Bibliothèque de résultats / théorèmes sur les modèles de communication et de fautes
- Assistant de preuve enrichi de théorèmes sur les paramètres
- Interaction vérificateur de modèles $\leftrightarrow$ assistant de preuve :
  - L'assistant de preuve décharge des propriétés au vérif. de modèles
  - Le vérif. de modèles nourrit l'assistant de preuve en invariants
- Sémantique formelle d'un langage dédié pour la description de systèmes paramétrés basés workflow

Rencontres et échanges

- Réunions plénières :
  - 30/01 - 31/01 - 01/02/17 (lancement)
  - 11/07/17
  - 20/12/17
  - *8-9/11/18*
- Échanges :
  - David Declerck : LRI $\rightarrow$ IRIT, 12/03/18 $\rightarrow$ 14/09/18
  - Sara Houhou : LIP6 $\rightarrow$ IRIT, 15/10/18 $\rightarrow$ 30/10/18
- Embauches :
  - Adam Shimi, doctorant Toulouse INP
  - Sara Houhou, doctorante co-totutelle UPMC
  - Poonam Kumari, master université de Lorraine

Task 1 : Parameters and systems

The overall objective of this task is to formulate application needs.

Deliverables

- ✓ D1.1 T0+5 **Case studies**
- ✓ D1.1 T0+10 **Case studies**
- ✓ D1.2 T0+6 **Report on the considered parameters**
- ✓ D1.2 T0+12 **Report on the considered parameters**
- ✓ D1.3 T0+18 Collection of generic results on parameters
- → D1.4 T0+40 Algorithm for minimality
- → D1.5 T0+48 Evaluation of the project results on the case studies

(extrait du texte de la proposition)

Task 2 : Automatic Verification of Parameterized Systems

The objective of this task is the cooperation between Cubicle and TLA$^+$ and to improve the expressiveness of the input language of Cubicle.

Deliverables

- × D2.1 T0+15 ~~Cubicle that can take as input TLA$^+$ programs~~
- ✓ **Translator of a subet of TLA$^+$ to Cubicle**
- ✓ **Cubicle for asynchronous communication**
- → D2.2 T0+24 Document describing the extension of Cubicle with new data types
- → D2.3 T0+36 ~~Cubicle with extended input language~~
- → D2.3 T0+44 Cubicle with extended input language

Task 3 : Mechanized Proofs of Parameterized Systems

Task 3 is devoted to tool support for the mechanical verification of the class of systems addressed in PARDI.

Deliverables

- × D3.1 T0+12 Architecture for interaction
- → D3.2 T0+30 New releases of Cubicle and TLAPS
- ✓ D3.3 T0+36 **TLA$^+$ library for communication** and fault models
- → D3.3 T0+44 TLA$^+$ library for communication and fault models

Task 4 : Automatic Verification of Parameterized Workflow Systems

The first objective is to define a pivot language (DSL) for PWS. Secondly [...] supporting the parameters identified in Task 1. Lastly how [adaptation, composition, and repair] can be verified.

Deliverables

- × D4.1 T0+12 ~~Definition of a PWS DSL~~
- ✓ **Formalisation of non-parameterized BPMN**
- ✓ **Formalisation of parameterized BPMN** (communication)
- → D4.1 T0+36 Transformation into TLA^+
- → D4.2 T0+36 & T0+48 Definition of a PWS checker
- → D4.3 T0+36 & T0+48 Application to adaptation, composition and repair

Difficultés / adaptations

- Cubicle prenant en entrée du TLA$^+$: TLA$^+$ très (trop) expressif
  $\Rightarrow$ un traducteur sous-ensemble de TLA$^+$ $\rightarrow$ Cubicle (version beta)
- Interaction vérificateur de modèles $\leftrightarrow$ assistant de preuve
  Le vérificateur de modèles peut exhiber des invariants.
  $\Rightarrow$ Travaux en cours sur l'interaction Cubicle $\leftrightarrow$ Why3 (thèse Mattias Roux, LRI)
- Langage dédié pour les workflows paramétrés
  La sémantique de BPMN est informelle et ambiguë mais utilité d'inventer un nouveau langage ?
  $\Rightarrow$ sémantique formelle de BPMN paramétré (nombre d'entités et communication asynchrone)

Résultats – études de cas

Toulouse INP, UPMC, INRIA Nancy, Univ. Paris Sud

Études de cas

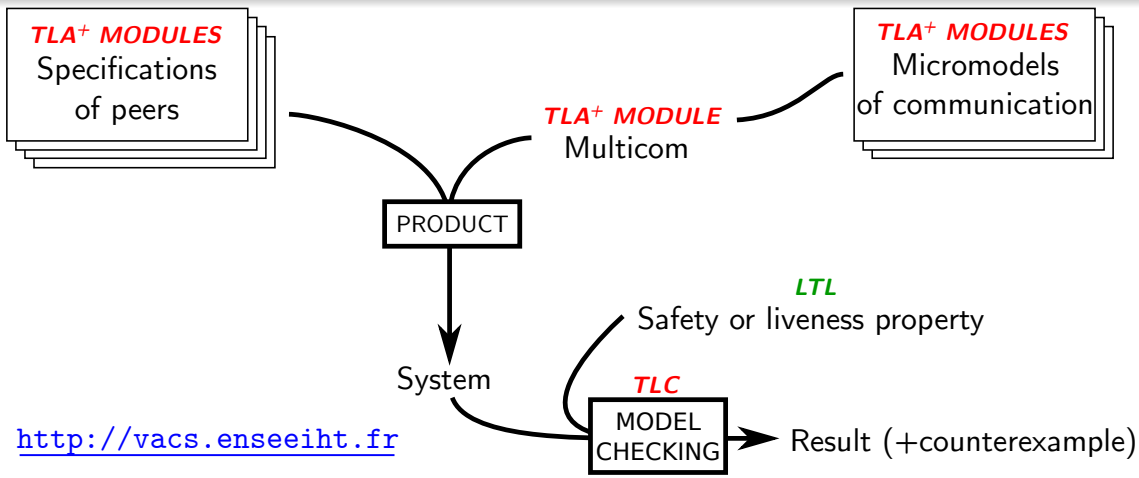
- 32 modèles en TLA$^+$, 20 en Cubicle, 3 en langage de workflow
- 6 algorithmes distribués (36 modèles), 3 workflows (8 modèles), 3 algorithmes concurrents (6 modèles)
- Paramètres numériques : nombre de processus, nombre de défaillances, nombre de valeurs échangées / décidées
- Relations entre les paramètres $nbproc > nbfault/3$, $nbfault < nbval < nbproc$
- Paramètres fonctionnels : structure, défaillances, **modèles de communication**

Résultats – framework & modèles de communication

Toulouse INP, INRIA Nancy

Framework

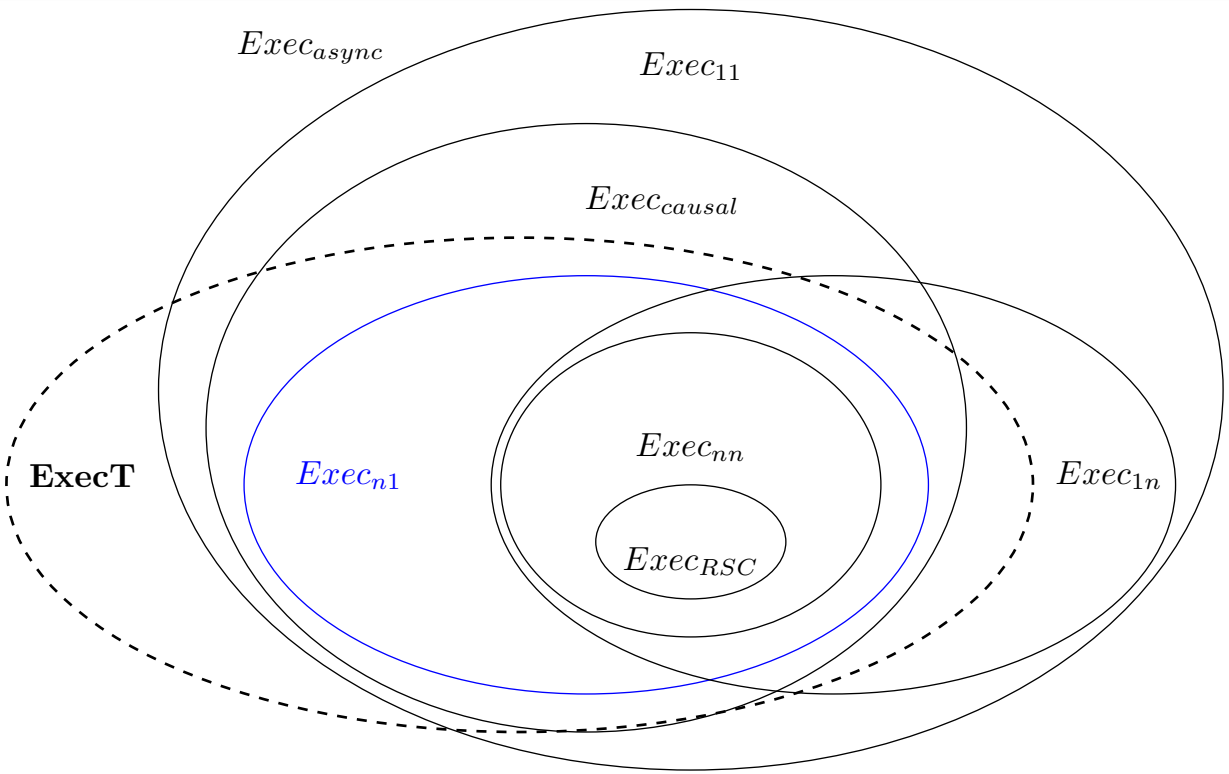
- Framework de vérification mécanisée en TLA$^+$
- Un modèle de communication est un assemblage de micro-modèles :
  - Multiplicité : point-à-point, diffusion, *convergecast*
  - Ordre de délivrance : génériques (FIFO 1-1, causal, etc), applicatifs (priorité)
  - Autres : limite sur le nombre de messages envoyés / en transit...
- Les spécifications opérationnelles TLA$^+$ sont prouvées corrects et complètes par rapport aux modèles axiomatiques



Résultats – framework & modèles de communication

Modèles de communication asynchrone

- Diversité de la communication asynchrone
- Résultats théoriques sur la hiérarchie des ordres de délivrance
- Différences point-à-point, diffusion (*broadcast*), *convergecast*



$Exec_{xy}$ = exécutions possibles avec xy

ExecT = diffusion totalement ordonnée

$Exec_{n1}$ = “boîte à lettres”

Résultats – Vérificateur de modèles

Univ. Paris Sud, Toulouse INP

Existant : Cubicle, vérificateur symbolique de modèles paramétrés

- État : variables partagées + tableaux indéxés par les processus
- Propriété : *cube* (prédicat avec quantification existentielle)
- Système étudié : système de transition avec transitions paramétrées par quantification existentielle (le ou les processus)
- Principe : atteignabilité arrière avec représentation symbolique des états par des formules logiques ($\rightarrow$ SMT Alt-Ergo)

$V := \emptyset$; push(Q, Unsafe)
while not_empty(Q) **do**
  φ := pop(Q)
  if $\varphi \wedge$ Init sat **then return** unsafe
  if $\neg(\varphi \vDash \bigvee_{\psi \in V} \psi)$ **then**
    $V := V \cup \{ \varphi \}$
    push(Q, $\mathrm{pre}_T(\varphi)$) *(* préimage *)*
return safe

Résultats – Vérificateur de modèles

Cubicle-W : modèle mémoire faible (thèse de David Declerck)

- Identification explicite du processus acteur
- Événements d'écriture et de lecture
- Ordres entre événements : *process order*, *read-from*, *coherency*...
- Analyse arrière en utilisant les relations (absence de cycle)
- Calcul de pré-image qui produit les événements et leur relations

Cubicle pour la communication asynchrone

- Canaux de communication avec perte
- Événements d'envoi et de réception de message
- Ordres des envois, des réceptions, lien émission - réception
  ($\rightarrow$ usuel ordre causal *happens-before*)
- 7 ordres de délivrance (fifo 1-1, causal...)

```
chan Req[1,1] : int
chan Perm[CAUSAL] : proc
```

```
transition req_ack([i] j)
requires { St[i]=Out && Req'j? <= Clock[i] }
{ Perm'j!i; }
```

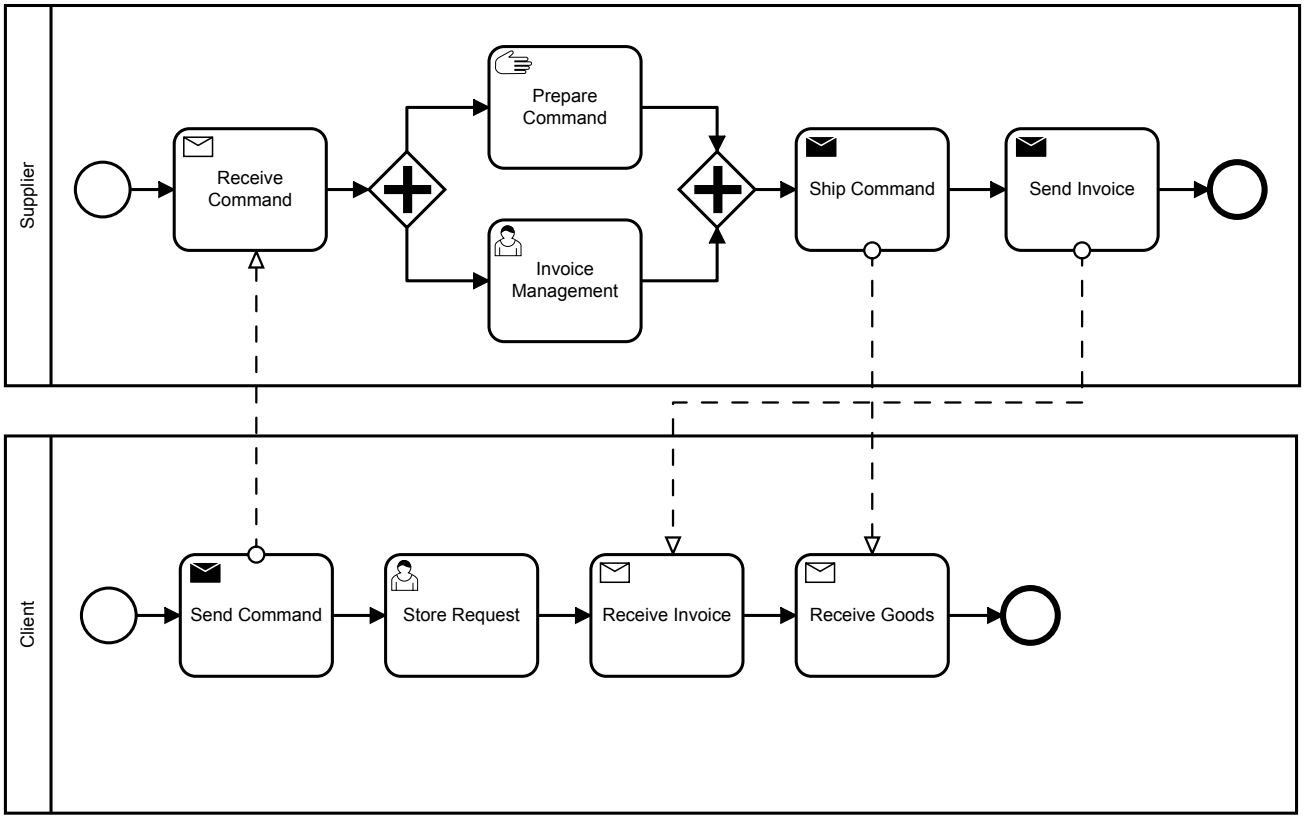
Résultats – PWS

UPMC, Univ. Paris Sud, Toulouse INP

Parameterized Workflows

- Représentation d'un workflow par un BPMN Graph
  $G = (N, E, cat_N, cat_E, M, type)$ avec boîtes à lettres structurées par le modèle de communication
- Représentation d'une collaboration entre workflows par un BPMN Collaboration Graph $\bar{G} = (G, Processes)$
- Sémantique formelle « directe » en FOL vs. sémantique transformationnelle
  $\rightarrow$ notion d'état avant / après très proche de TLA$^+$
- Paramétrage de la communication isolé dans des prédicats
  $\approx$ prédicats TLA$^+$ développés dans le framework

Résultats – PWS (2)



- Communication causale : blocage
- Communication FIFO 1-1 : blocage si même entité envoie la commande et la facture
- Communication asynchrone : OK
- Plusieurs clients ? Plusieurs fournisseurs ?

Diffusion

- Site web `http://pardi.enseeiht.fr/`
- 7 publications : OPODIS17, ICFEM17, IJCAR18...
  (mais toutes mono-site)
- 5 exposés invités : FRIDA Vienne (DISC 2017) $\times 2$, VDS Maroc 2018, FRIDA Oxford (FloC 2018), IRIF 2018
- Cubicle avec canaux :
  `https://github.com/cubicle-model-checker/`
- Framework TLA$^+$: `http://vacs.enseeiht.fr/`

Perspectives

1. Arrivée de Igor Konnov à INRIA Nancy, équipe Veridis : Byzantine (Parameterized) Model Checker, model checker symbolique pour TLA$^+$ $\rightarrow$ transformation de spécifications paramétrées vers Cubicle ou ce model checker
2. Cubicle avec canaux : amélioration de la convergence + nouveaux exemples
3. Coopération Cubicle $\leftrightarrow$ Why3 (thèse Mattias Roux)
4. Codage de la sémantique FOL de BPMN paramétré en TLA$^+$ (séjour de Sara Houhou à Toulouse)
5. Paramétrage multi-instance de workflows, données
6. Défaillances : algorithmes structurés en *round*, plus aisés à prouver : modèle de communication opérationnel + stratégie de choix des messages $\rightarrow$ modèle Heard-Of (thèse Adam Shimi)