Nantes Université

Laboratoire des Sciences du Numérique de Nantes

Journée de rentrée EUR MaSTIC

2025-2026

Perdez votre temps et votre argent (et celui des autres) Égarez les données de vos expériences

Étienne André

Nantes Université

Etienne.Andre@univ-nantes.fr

Version diapositives : 18 novembre 2025

Outline

Un premier mauvais exemple (2010) (1/2)

2010 : publication d'un *tool paper* décrivant un logiciel de *model checking* paramétré (« IMITATOR ») calculant des contraintes à partir de modèles [And10]

Tableau d'expériences :

Example	PTAs	loc./PTA	$\|X\|$	$\|P\|$	iter.	$\|K_0\|$	states	trans.	Time1	Time2
SR-latch	3	[3,8]	3	3	5	2	4	3	0.11	0.007
Flip-flop [13]	5	[4,16]	5	12	9	6	11	10	1.6	0.122
And–Or [12]	3	[4,8]	4	12	14	4	13	13	1.81	0.15
Valmem Latch	7	[2,5]	8	13	12	6	18	17	14.4	0.345
CSMA/CD [22]	3	[3,8]	3	3	19	2	219	342	41	1.01
RCP [21]	5	[6,11]	6	5	20	2	327	518	64	2.3
SPSMALL_1 [11]	10	[3,8]	10	26	32	23	31	30	4680	2.6
BRP [24]	6	[2,6]	7	6	30	7	429	474	901	34
SPSMALL_2 [11]	28	[2,11]	28	5	92	8	472	548	-	1755

Table 2: Summary of experiments for the inverse method

. [And10] Étienne André. « IMITATOR II : A Tool for Solving the Good Parameters Problem in Timed Automata ». In : *INFINITY*. T. 39. Electronic Proceedings in Theoretical Computer Science. Sept. 2010, p. 91-99

Un premier mauvais exemple (2010) (1/2)

2010 : publication d'un *tool paper* décrivant un logiciel de *model checking* paramétré (« IMITATOR ») calculant des contraintes à partir de modèles [And10]

Tableau d'expériences :

Example	PTAs	loc./PTA	$\|X\|$	$\|P\|$	iter.	$\|K_0\|$	states	trans.	Time1	Time2
SR-latch	3	[3,8]	3	3	5	2	4	3	0.11	0.007
Flip-flop [13]	5	[4,16]	5	12	9	6	11	10	1.6	0.122
And–Or [12]	3	[4,8]	4	12	14	4	13	13	1.81	0.15
Valmem Latch	7	[2,5]	8	13	12	6	18	17	14.4	0.345
CSMA/CD [22]	3	[3,8]	3	3	19	2	219	342	41	1.01
RCP [21]	5	[6,11]	6	5	20	2	327	518	64	2.3
SPSMALL_1 [11]	10	[3,8]	10	26	32	23	31	30	4680	2.6
BRP [24]	6	[2,6]	7	6	30	7	429	474	901	34
SPSMALL_2 [11]	28	[2,11]	28	5	92	8	472	548	-	1755

Table 2: Summary of experiments for the inverse method

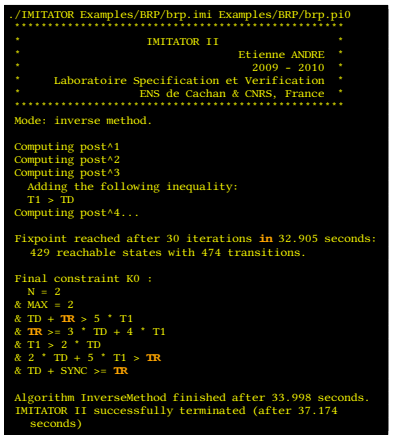
- ☹ pas de numéro de version du logiciel
  - oui oui, « `imitator-nouveau-modif-v2-copie` »
- ☹ pas de code archivé sur git ou SVN
  - oui oui, sur un disque dur d'ordinateur portable non synchronisé
- ☹ version des modèles non sauvegardée

. [And10] Étienne André. « IMITATOR II : A Tool for Solving the Good Parameters Problem in Timed Automata ». In : *INFINITY*. T. 39. Electronic Proceedings in Theoretical Computer Science. Sept. 2010, p. 91-99

Un premier mauvais exemple (2010) (2/2)

Conséquences pour moi :

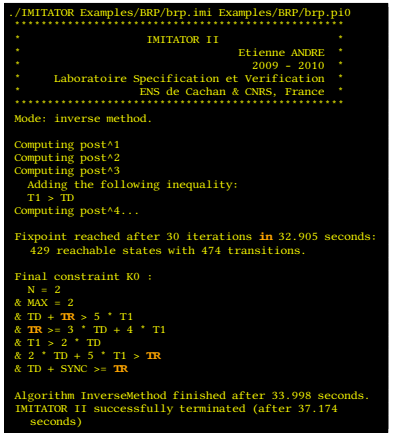
- ☹ impossible de reproduire ces expériences quelques années plus tard alors que je voulais montrer la supériorité d'une nouvelle version du logiciel
- ☹ résultats parfois différents
  - venant peut-être d'une version du logiciel ou des modèles différente ?
- ☹ certaines analyses ne terminent plus alors qu'elles terminaient selon le *tool paper* de 2010
  - là encore, pas de certitude d'avoir accès aux bons modèles



Un premier mauvais exemple (2010) (2/2)

Conséquences pour moi :

- ☹ impossible de reproduire ces expériences quelques années plus tard alors que je voulais montrer la supériorité d'une nouvelle version du logiciel
- ☹ résultats parfois différents
  - venant peut-être d'une version du logiciel ou des modèles différente ?
- ☹ certaines analyses ne terminent plus alors qu'elles terminaient selon le *tool paper* de 2010
  - là encore, pas de certitude d'avoir accès aux bons modèles



Conséquences pour la communauté :

- ☹ impossible de comparer de nouveaux logiciels à ce travail
- ☹ manque de confiance dans les résultats (non-reproductibilité)

Un second mauvais exemple (2013) (1/2)

2013 : développement d'un logiciel (« USMMC ») pour la vérification formelle automatisée (par *model checking*) des diagrammes états-transitions UML [Liu+13]

USMMC: a self-contained model checker for UML state machines

Authors: Shuang Liu, Yang Liu, Jun Sun, Manchun Zheng, Bimlesh Wadhwa, Jin Song Dong Authors Info & Claims

ESEC/FSE 2013: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering • August 2013 • Pages 623–626 • https://doi.org/10.1145/2491411.2494595

Online: 18 August 2013 Publication History

5 220 Get Access

ABSTRACT

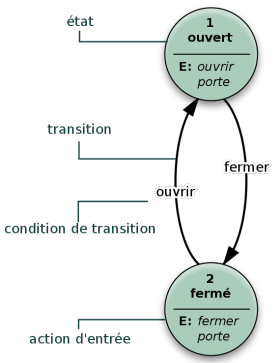
UML diagrams are gaining increasing usage in Object-Oriented system designs. UML state machines are specifically used in modeling dynamic behaviors of classes. It has been widely agreed that verification of system designs at an early stage will dramatically reduce the development cost. Tool support for verification UML designs can also encourage consistent usage of UML diagrams

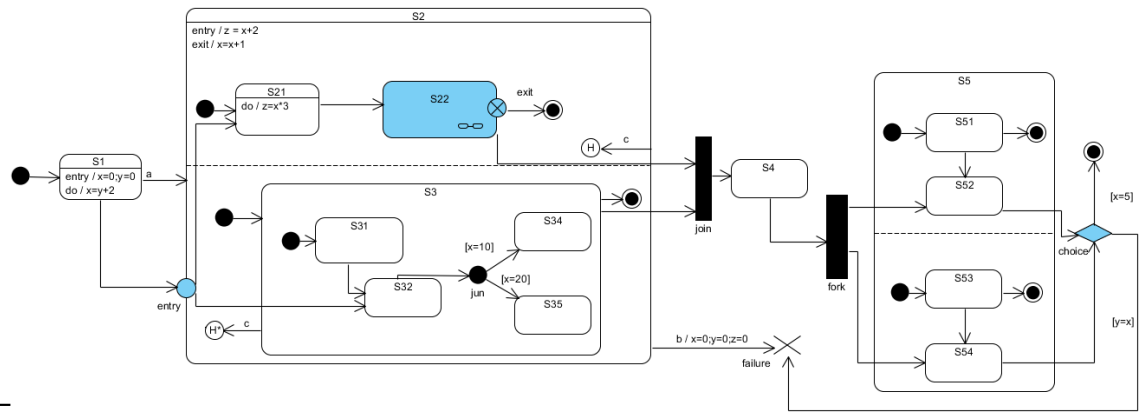
Table 1: Evaluation results

Model	Property	Result	USMMC				HUGO				
			Time(s)	State	Transition	Mem (KiB)	TTime(s)	ETime(s)	State	Transition	Mem (KiB)
RailCar	Prop1	not valid	0.013	30	34	43, 342	-	-	-	-	-
RailCarO	Prop1	valid	0.011	44	54	43, 058	-	-	-	-	-
BankATM	Prop2	valid	0.009	25	28	917.5	0.231	0.050	578	1, 133	98, 528
TollGate	Prop3	valid	0.110	36	50	43, 345	0.197	0.505	61, 451	256, 807	100, 578
DP2	deadlock	not valid	0.005	39	65	2, 318	0.196	0.111	12, 766	42, 081	98, 918
DP3	deadlock	not valid	0.039	237	589	10, 145	0.242	379.009	4, 626, 838	23, 897, 077	276, 067
DP4	deadlock	not valid	0.34	1, 519	5, 079	21, 059	1.117	8944.754	57, 213, 708	339, 761, 530	3, 059, 807
DP5	deadlock	not valid	3.11	9, 634	40, 366	92, 369	-	-	-	-	-
DP6	deadlock	not valid	27.87	63, 069	324, 275	226, 271	-	-	-	-	-
DP7	deadlock	not valid	232.64	398, 101	2, 385, 361	2, 852, 672	-	-	-	-	-

Prop1=$\Box(alert100 \rightarrow \Diamond arriveAck)$, Prop2=$\Box(retain \rightarrow ((!cardValid \wedge numIncorrect \geq maxNumIncorrect))$, Prop3=$\Box(TurnGreen \rightarrow \Diamond carExit)$.

- Exemples de diagrammes états-transitions UML :





. [Liu+13] Shuang Liu, Yang Liu, Jun Sun, Manchun Zheng, Bimlesh Wadhwa et Jin Song Dong. « USMMC : A self-contained model checker for UML state machines ». In : *ESEC/FSE*. ACM, 2013, p. 623-626

Un second mauvais exemple (2013) (2/2)

2016 : un expert du domaine développeur d'un outil concurrent nous a contactés pour obtenir notre logiciel pour pouvoir s'y comparer

Un second mauvais exemple (2013) (2/2)

2016 : un expert du domaine développeur d'un outil concurrent nous a contactés pour obtenir notre logiciel pour pouvoir s'y comparer

… mais l'intégralité du code de notre outil de 2013 a disparu :

- pas de code sur git ou SVN ; pas même d'exécutable (binaire)
- pas de modèles ni résultats des expériences
- La doctorante l'ayant développé a quitté son université, laquelle a effacé tous ses fichiers le soir-même de sa soutenance de thèse
- Il y aurait bien une copie sur un disque dur externe quelque part en Chine… mais on ne l'a jamais retrouvée
  - soupçon possible de violation de

Un second mauvais exemple (2013) (2/2)

2016 : un expert du domaine développeur d'un outil concurrent nous a contactés pour obtenir notre logiciel pour pouvoir s'y comparer

… mais l'intégralité du code de notre outil de 2013 a disparu :

- pas de code sur git ou SVN ; pas même d'exécutable (binaire)
- pas de modèles ni résultats des expériences
- La doctorante l'ayant développé a quitté son université, laquelle a effacé tous ses fichiers le soir-même de sa soutenance de thèse
- Il y aurait bien une copie sur un disque dur externe quelque part en Chine… mais on ne l'a jamais retrouvée
  - soupçon possible de violation de l'intégrité scientifique

Conséquences :

- ☹ perte pour l'équipe de développement (valorisation)
- ☹ perte pour la communauté (améliorations, comparaisons)

Outline

Les données de la recherche

Définition (les données de la recherche (OCDE))

Enregistrements factuels (chiffres, textes, images et sons) qui sont utilisés comme sources principales pour la recherche scientifique et sont généralement reconnus par la communauté scientifique comme nécessaires pour valider des résultats de recherche

Les données de la recherche

Définition (les données de la recherche (OCDE))

Enregistrements factuels (chiffres, textes, images et sons) qui sont utilisés comme sources principales pour la recherche scientifique et sont généralement reconnus par la communauté scientifique comme nécessaires pour valider des résultats de recherche

- Les données ne sont pas (seulement) du code informatique
- L'accès aux données de la recherche est différent de l'accès aux publications (lequel est également essentiel)
  - NB : données ouvertes et publications ouvertes font partie de la science ouverte

Les jeux de données en informatique théorique

Exemples :

Les jeux de données en informatique théorique

Exemples :

- Des programmes rédigés dans un langage informatique

```
for i in range(1, 1138):
    j += i
    print('Hello ' + j)
```

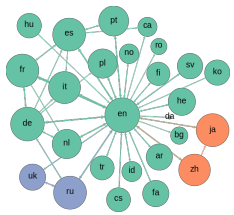
Les jeux de données en informatique théorique

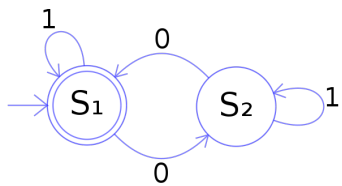
Exemples :

- Des programmes rédigés dans un langage informatique

```python
for i in range(1, 1138):
    j += i
    print('Hello ' + j)
```

- Des graphes ou des automates
  - représentés sous forme textuelle






```
s1 -(1)-> s1
s1 -(0)-> s2
s2 -(0)-> s1
s2 -(1)-> s2
```

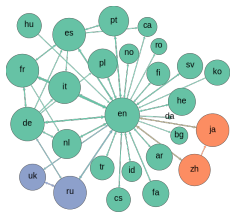
Les jeux de données en informatique théorique

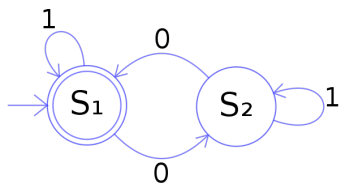
Exemples :

- Des programmes rédigés dans un langage informatique

```python
for i in range(1, 1138):
    j += i
    print('Hello ' + j)
```

- Des graphes ou des automates
  - représentés sous forme textuelle






```
s1 -(1)-> s1
s1 -(0)-> s2
s2 -(0)-> s1
s2 -(1)-> s2
```

- Des données numériques (logs, suites de nombres…)

```
@t=2.3: temperature=2.3; vitesse=4.5
@t=2.7: temperature=3.2; vitesse=6.9
@t=4.9: temperature=5.1; vitesse=11.38
```

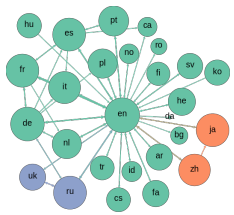
Les jeux de données en informatique théorique

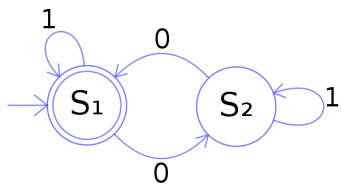
Exemples :

- Des programmes rédigés dans un langage informatique

```
for i in range(1, 1138):
    j += i
    print('Hello ' + j)
```

- Des graphes ou des automates
  - représentés sous forme textuelle






```
s1 -(1)-> s1
s1 -(0)-> s2
s2 -(0)-> s1
s2 -(1)-> s2
```

- Des données numériques (logs, suites de nombres…)

```
@t=2.3: temperature=2.3; vitesse=4.5
@t=2.7: temperature=3.2; vitesse=6.9
@t=4.9: temperature=5.1; vitesse=11.38
```

- Des modèles formels exprimés dans un langage formel (ex : TLA+)

```
VARIABLE clock
Init == clock \in {0, 1}
Tick == IF clock = 0 THEN clock' = 1 ELSE clock' = 0
Spec == Init /\ [][Tick]_«clock»
```

Les jeux de données en informatique théorique : spécificités

- Généralement du texte
  - Code
  - Modèles formels codés dans des formats textuels (XML, JSON, CSV…)
  - Suites de nombres
  - À la différence de : données d'imageries (médecine, géographie…)
- Taille relativement modeste
  - Taille typique d'un programme ou d'un modèle : quelques dizaines ou centaines de kio
  - Un programme même de grande taille excède très rarement 100 Mio
  - À la différence de : banques d'images (médecine, géographie, apprentissage artificiel…), relevés météorologiques, etc.
    - Par exemple, 1 répondant·e sur 5 mentionne dans une enquête des données de plus de 1 To [Le +22]
- Rarement de questions de confidentialité ou d'éthique
  - À la différence de : médecine, sociologie…
  - (Contre-exemples : collaborations industrielles avec accord de confidentialité…)

. [Le +22] Mariannig Le Béchec, Aline Bouchard, Philippe Charrier, Claire Denecker, Gabriel Gallezot et Stéphanie Rennes. *Pratiques et usages des outils numériques dans les communautés scientifiques en France*. Research Report. Comité pour la science ouverte, jan. 2022

La reproductibilité : une évidence ?

Dans de nombreux domaines, il est extrêmement difficile voire impossible de reproduire des expériences à l'identique

- Médecine, biologie, sociologie, linguistique…
- Cause principale :

La reproductibilité : une évidence ?

Dans de nombreux domaines, il est extrêmement difficile voire impossible de reproduire des expériences à l'identique

- Médecine, biologie, sociologie, linguistique…
- Cause principale : impossibilité d'appliquer deux fois la même méthode sur exactement les mêmes données avec le même environnement

En informatique, un algorithme ou un programme avec les mêmes entrées devrait produire le même résultat

- Les temps d'exécution devraient être similaires
  - Mais pas identiques (par ex :

La reproductibilité : une évidence ?

Dans de nombreux domaines, il est extrêmement difficile voire impossible de reproduire des expériences à l'identique

- Médecine, biologie, sociologie, linguistique…
- Cause principale : impossibilité d'appliquer deux fois la même méthode sur exactement les mêmes données avec le même environnement

En informatique, un algorithme ou un programme avec les mêmes entrées devrait produire le même résultat

- Les temps d'exécution devraient être similaires
  - Mais pas identiques (par ex : bruit lié à l'occupation de la machine)
- Pas universel pour autant (Contre-exemples :

La reproductibilité : une évidence ?

Dans de nombreux domaines, il est extrêmement difficile voire impossible de reproduire des expériences à l'identique

- Médecine, biologie, sociologie, linguistique…
- Cause principale : impossibilité d'appliquer deux fois la même méthode sur exactement les mêmes données avec le même environnement

En informatique, un algorithme ou un programme avec les mêmes entrées devrait produire le même résultat

- Les temps d'exécution devraient être similaires
  - Mais pas identiques (par ex : bruit lié à l'occupation de la machine)
- Pas universel pour autant (Contre-exemples : algorithmes distribués, programmes fonctionnant sur des réseaux, programme avec une part d'aléatoire…)

La reproductibilité en informatique : les obstacles

Obstacles à la reproductibilité des expériences en informatique :

La reproductibilité en informatique : les obstacles

Obstacles à la reproductibilité des expériences en informatique :

- Absence de partage des logiciels, ou manque d'information (version, environnement)

La reproductibilité en informatique : les obstacles

Obstacles à la reproductibilité des expériences en informatique :

- Absence de partage des logiciels, ou manque d'information (version, environnement)
- Absence de partage des données d'entrée ou manque d'information (version)

La reproductibilité en informatique : les obstacles

Obstacles à la reproductibilité des expériences en informatique :

- Absence de partage des logiciels, ou manque d'information (version, environnement)
- Absence de partage des données d'entrée ou manque d'information (version)
- Absence de partage des résultats attendus

La reproductibilité en informatique : les obstacles

Obstacles à la reproductibilité des expériences en informatique :

- Absence de partage des logiciels, ou manque d'information (version, environnement)
- Absence de partage des données d'entrée ou manque d'information (version)
- Absence de partage des résultats attendus
- Problématique de l'environnement (matériel, système d'exploitation)

Point-clé

Reproductibilité des expériences

Il devrait être extrêmement facile de reproduire des expériences à l'identique en informatique théorique

Cette reproductibilité repose en grande partie sur

Point-clé

Reproductibilité des expériences

Il devrait être extrêmement facile de reproduire des expériences à l'identique en informatique théorique

Cette reproductibilité repose en grande partie sur l'ouverture des données de la recherche

Outline

Outline

Partager car c'est obligatoire (1/4)

De plus en plus de congrès internationaux en informatique théorique demandent ou recommandent des *artifact evaluations*, notamment pour les *tool papers* :

- Obligatoire à CAV (Computer Aided Verification)
- Obligatoire à TACAS (Tools and Algorithms for the Construction and Analysis of Systems)
- Recommandé à OOPSLA, POPL (Principles of Programming Languages)...

Artifact submission and evaluation

Regular tool papers and tool demonstration papers must be accompanied by an artifact, registered by the paper submission deadline and submitted by the specific deadline shortly after.

Principes

- Expériences reproduites par un comité d'évaluation dédié
- Article pouvant être rejeté faute de reproductibilité
- ACM (*Association for Computing Machinery*) : 3 badges (évaluation, disponibilité, validation)

Objectifs :

Partager car c'est obligatoire (1/4)

De plus en plus de congrès internationaux en informatique théorique demandent ou recommandent des *artifact evaluations*, notamment pour les *tool papers* :

- Obligatoire à CAV (Computer Aided Verification)
- Obligatoire à TACAS (Tools and Algorithms for the Construction and Analysis of Systems)
- Recommandé à OOPSLA, POPL (Principles of Programming Languages)...

Artifact submission and evaluation

Regular tool papers and tool demonstration papers must be accompanied by an artifact, registered by the paper submission deadline and submitted by the specific deadline shortly after.

Principes

- Expériences reproduites par un comité d'évaluation dédié
- Article pouvant être rejeté faute de reproductibilité
- ACM (*Association for Computing Machinery*) : 3 badges (évaluation, disponibilité, validation)

Objectifs :

- ☺ Permettre la reproductibilité des expériences

Partager car c'est obligatoire (1/4)

De plus en plus de congrès internationaux en informatique théorique demandent ou recommandent des *artifact evaluations*, notamment pour les *tool papers* :

- Obligatoire à CAV (Computer Aided Verification)
- Obligatoire à TACAS (Tools and Algorithms for the Construction and Analysis of Systems)
- Recommandé à OOPSLA, POPL (Principles of Programming Languages)...

Artifact submission and evaluation

Regular tool papers and tool demonstration papers must be accompanied by an artifact, registered by the paper submission deadline and submitted by the specific deadline shortly after.

Principes

- Expériences reproduites par un comité d'évaluation dédié
- Article pouvant être rejeté faute de reproductibilité
- ACM (*Association for Computing Machinery*) : 3 badges (évaluation, disponibilité, validation)

Objectifs :

- ☺ Permettre la reproductibilité des expériences
- ☺ Permettre la dissémination des données et logiciels

Partager car c'est obligatoire (2/4)

- L'ANR (Agence Nationale de la Recherche) demande la rédaction d'un plan de gestion des données pour tous les projets acceptés

  - Principe : « ouvrir les données autant que possible »

Partager car c'est obligatoire (3/4)

Les laboratoires de recherche doivent :

> *... veiller à la mise en œuvre par leur personnel de plans de gestion de données et contribuent aux infrastructures qui permettent la conservation, la communication et la réutilisation des données et des codes sources*
>
> *Décret n°2021-1572 du 3 décembre 2021, article 6*

Partager car c'est obligatoire (4/4)

- Le programme-cadre de l'UE Horizon Europe demande à ce que les données des programmes financés soient par défaut ouvertes

  Licences CC-BY (*attribution required*) ou CC-0 (domaine public)

  - Sauf exceptions (intérêts du bénéficiaire, confidentialité, ...)

Principe « aussi ouvert que possible, aussi fermé que nécessaire »

Outline

Partager car c'est dans votre intérêt (1/2)

. [Col+20] Giovanni Colavizza, Iain Hrynaszkiewicz, Isla Staden, Kirstie Whitaker et Barbara McGillivray. « The citation advantage of linking publications to research data ». In : *PLoS One* 15 (4 2020)

Partager car c'est dans votre intérêt (1/2)

- ☺ Pouvoir réutiliser ses propres expériences

. [Col+20] Giovanni Colavizza, Iain Hrynaszkiewicz, Isla Staden, Kirstie Whitaker et Barbara McGillivray. « The citation advantage of linking publications to research data ». In : *PLoS One* 15 (4 2020)

Partager car c'est dans votre intérêt (1/2)

- ☺ Pouvoir réutiliser ses propres expériences
- ☺ Diminue fortement la probabilité de perte de données

. [Col+20] Giovanni Colavizza, Iain Hrynaszkiewicz, Isla Staden, Kirstie Whitaker et Barbara McGillivray. « The citation advantage of linking publications to research data ». In : *PLoS One* 15 (4 2020)

Partager car c'est dans votre intérêt (1/2)

- ☺ Pouvoir réutiliser ses propres expériences
- ☺ Diminue fortement la probabilité de perte de données
- ☺ Amélioration du nombre de citations
  - Data Publications Correlate with Citation Impact :

> *We also find an association between articles that include statements that link to data in a repository and up to 25.36% (± 1.07%) higher citation impact on average, using a citation prediction model.*
>
> *[Col+20]*

. [Col+20] Giovanni Colavizza, Iain Hrynaszkiewicz, Isla Staden, Kirstie Whitaker et Barbara McGillivray. « The citation advantage of linking publications to research data ». In : *PLoS One* 15 (4 2020)

Partager car c'est dans votre intérêt (2/2)

Possibilité de rédiger un data paper visant à publiciser un jeu de données sous forme de publication :

- ☺ une publication est (potentiellement) mieux valorisée qu'un jeu de données
- ☺ citation à chaque utilisation de votre jeu de données
- ☺ amélioration de la reconnaissance dans la communauté

A Benchmark Suite for Hybrid Systems Reachability Analysis

Xin Chen[1], Stefan Schupp[1(✉)], Ibtissem Ben Makhlouf[1], Erika Ábrahám[1], Goran Frehse[2], and Stefan Kowalewski[1]

[1] RWTH Aachen University, Aachen, Germany
[2] Verimag, Gières, France
stefan.schupp@cs.rwth-aachen.de

Abstract. Since about two decades, formal methods for continuous and hybrid systems enjoy increasing interest in the research community. A wide range of analysis techniques were developed and implemented in

[Che+15]

. [Che+15] Xin Chen, Stefan Schupp, Ibtissem Ben Makhlouf, Erika Ábrahám, Goran Frehse et Stefan Kowalewski. « A Benchmark Suite for Hybrid Systems Reachability Analysis ». In : *NFM*. T. 9058. Lecture Notes in Computer Science. Springer, 2015, p. 408-414

Outline

Partager car c'est facile (1/2)

En informatique théorique :

- les données sont souvent de relativement petite taille
  - À la différence de : traitement d'images, fouille de données, géographie…
- il y a généralement peu de questions éthiques ou de confidentialité
  - À la différence de : médecine, sociologie…

Exemple de taille d'*artifact* pour un de mes articles présentant une technique de *model checking* pour les automates temporisés paramétrés [And+22] :

Nature	**Taille**
modèles d'entrée et scripts	8,3 Mio
logiciel	11,9 Mio
résultats attendus	51,3 Mio (dupliqués 5 fois !)
total	**72,3 Mio**

https://zenodo.org/record/6806915

(Soit

. [And+22] Étienne André, Dylan Marinho, Laure Petrucci et Jaco van de Pol. « Efficient Convex Zone Merging in Parametric Timed Automata ». In : *FORMATS*. T. 13465. Lecture Notes in Computer Science. Springer, 2022, p. 200-218

Partager car c'est facile (1/2)

En informatique théorique :

- les données sont souvent de relativement petite taille
  - À la différence de : traitement d'images, fouille de données, géographie…
- il y a généralement peu de questions éthiques ou de confidentialité
  - À la différence de : médecine, sociologie…

Exemple de taille d'*artifact* pour un de mes articles présentant une technique de *model checking* pour les automates temporisés paramétrés [And+22] :

Nature	**Taille**
modèles d'entrée et scripts	8,3 Mio
logiciel	11,9 Mio
résultats attendus	51,3 Mio (dupliqués 5 fois !)
total	**72,3 Mio**

https://zenodo.org/record/6806915

(Soit 0,11 % de l'espace offert par une clé USB de 64 Gio)

. [And+22] Étienne André, Dylan Marinho, Laure Petrucci et Jaco van de Pol. « Efficient Convex Zone Merging in Parametric Timed Automata ». In : *FORMATS*. T. 13465. Lecture Notes in Computer Science. Springer, 2022, p. 200-218

Partager car c'est facile (2/2)

Exemples de jeux de données : les grandes banques de modèles (*benchmarks*)

- la banque de modèles d'une compétition de vérification automatisée (Petri Nets model checking contest [Kor+21])
  - Format PNML (basé sur XML, donc textuel)
  - 1617 modèles concrets et 310 464 formules (édition 2022)
  - 899 Mio au total (soit

. [Kor+21] Fabrice Kordon, Lom-Messan Hillah, Francis Hulin-Hubard, Loïg Jezequel et Emmanuel Paviot-Adet. « Study of the efficiency of model checking techniques using results of the MCC from 2015 To 2019 ». In : *International Journal on Software Tools for Technology Transfer* 23.6 (2021), p. 931-952

. [Sut17] Geoff Sutcliffe. « The TPTP Problem Library and Associated Infrastructure - From CNF to THo, TPTP v6.4.o ». In : *Journal of Automated Reasoning* 59.4 (2017), p. 483-502

Partager car c'est facile (2/2)

Exemples de jeux de données : les grandes banques de modèles (*benchmarks*)

- la banque de modèles d'une compétition de vérification automatisée (Petri Nets model checking contest [Kor+21])
  - Format PNML (basé sur XML, donc textuel)
  - 1617 modèles concrets et 310 464 formules (édition 2022)
  - 899 Mio au total (soit 1,4 % de l'espace d'une clé de 64 Gio)

Mauvaise pratique : modèles placés sur un site institutionnel (laboratoire), et non sous forme d'un dépôt dans un entrepôt de données

- TPTP (*Thousands of Problems for Theorem Provers*) : 25 257 problèmes, 864 Mio

Mauvaise pratique : modèles placés sur un site de laboratoire, et non sous forme d'un dépôt dans un entrepôt de données à long terme

- SMT-LIB (gitlab) : 360 507 modèles (format standard SMT-LIB), ≈ 50 Gio [Sut17]

. [Kor+21] Fabrice Kordon, Lom-Messan Hillah, Francis Hulin-Hubard, Loïg Jezequel et Emmanuel Paviot-Adet. « Study of the efficiency of model checking techniques using results of the MCC from 2015 To 2019 ». In : *International Journal on Software Tools for Technology Transfer* 23.6 (2021), p. 931-952

. [Sut17] Geoff Sutcliffe. « The TPTP Problem Library and Associated Infrastructure - From CNF to THo, TPTP v6.4.o ». In : *Journal of Automated Reasoning* 59.4 (2017), p. 483-502

Outline

Partager avant tout pour la science

- Reproductibilité des expériences $\Rightarrow$ amélioration de la confiance
  - Meilleure

Partager avant tout pour la science

- Reproductibilité des expériences $\Rightarrow$ amélioration de la confiance
  - Meilleure intégrité scientifique
  - Confiance accrue de la population envers la science

- Partage de connaissances $\Rightarrow$ aider les futur·e·s scientifiques qui viendront améliorer l'état de l'art
  - Réutilisation et amélioration de logiciels existants
  - Réutilisation et amélioration de jeux de données existants
  - Comparaison facilitée avec de nouvelles techniques

Outline

Que partager ?

Si la taille le permet : tout

Un *artifact* peut contenir :

- logiciel (code + binaire)
- données brutes d'entrée (modèles…)
- résultats attendus
- système complet ? $\Rightarrow$ machine virtuelle / image docker
- des instructions détaillées (fichier `README.md`)
- des scripts aussi automatisés que possibles
  - Dans l'idéal : un simple script tel que `run.sh`

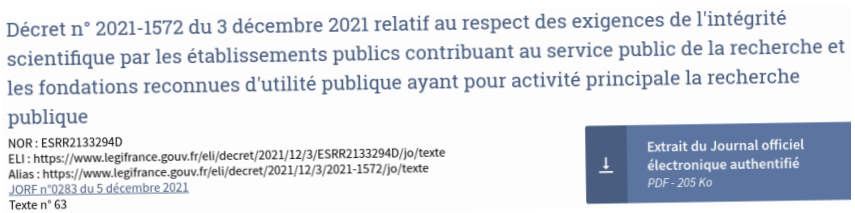
Ne pas oublier :

- Documentation : système, bibliothèques nécessaires, etc.
- Versions
- Description formelle ou (au moins) informelle des formats

Que partager ? (suite)

Penser à partager (et publier) les résultats négatifs

- Décret n°2021-1572 du 3 décembre 2021 : incite « à la publication des résultats de recherche dits négatifs »
- Évite à d'autres universitaires de réessayer une « mauvaise » solution



https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000044411360

Où partager : dans des entrepôts de données

Où partager les données de la recherche ?

Les données de la recherche doivent être partagées dans des entrepôts de données accessibles à très long terme (au strict minimum une décennie, idéalement plusieurs).

Où partager : dans des entrepôts de données

Où partager les données de la recherche ?

Les données de la recherche doivent être partagées dans des entrepôts de données accessibles à très long terme (au strict minimum une décennie, idéalement plusieurs).

Éviter absolument :

- ☹ les clés USB, disques durs externes…
- ☹ les pages personnelles
- ☹ les pages professionnelles
- ☹ les pages sur un site commercial (Google…)
- ☹ les sites de laboratoires ou d'universités

Ces endroits disparaissent beaucoup vite que l'on ne le croit !

Exemple : pertes de logiciels académiques

Sur 11 logiciels de vérification de diagrammes états-transitions UML codés entre 1999 et 2022, seuls 4 étaient encore disponibles quelque part en 2023 [And+23]

- tous les autres étaient disponibles via des URL de pages de laboratoires, depuis disparues

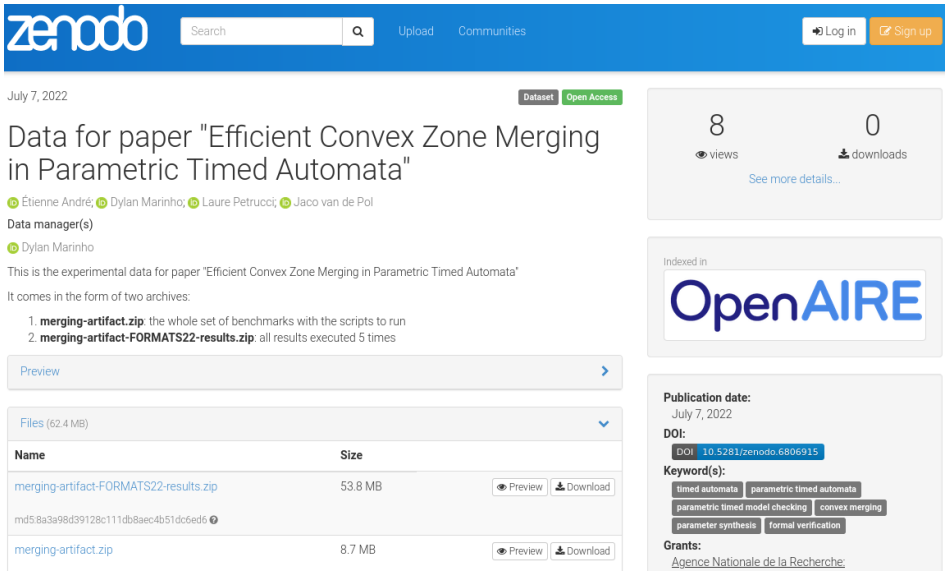
. [And+23] Étienne André, Shuang Liu, Yang Liu, Christine Choppy, Jun Sun et Jin Song Dong. « Formalizing UML State Machines for Automated Verification – A Survey ». In : *ACM Computing Surveys* 55.13 (juill. 2023), 277 :1-277 :47

Où partager : les entrepôts disciplinaires

Données et code permettant la reproductibilité :

- Meilleur choix (à titre personnel (?)) : Zenodo
- Autre excellente option en France : Entrepôt national Recherche Data Gouv
  - Propose également des ateliers de la donnée dans toute la France

Obtention systématique d'un DOI (Digital Object Identifier), qui peut ensuite être cité



Versions du code :

- forge logiciel (par exemple GitHub)
- Software Heritage

Software Heritage

- plateforme lancée en 2016
- archive ouverte pour les codes source des logiciels
- soutiens : Inria, CNRS, UNESCO…
- fondateurs : Roberto Di Cosmo et Stefano Zacchiroli

- Collectage automatique depuis GitHub, GitLab.com, Bitbucket
- connexion à HAL

GitHub ou pas GitHub ?

- ☺ pérenne (?)
- ☺ gratuit
- ☺ open source
- ☺ connexion à Software Heritage

GitHub ou pas GitHub ?

- ☺ pérenne (?)
- ☺ gratuit
- ☺ open source
- ☺ connexion à Software Heritage
- ☹ peut être racheté à tout moment (cas de Twitter/X)

Exemple : article et jeu de données

Timed Automata Verification and Synthesis via Finite Automata Learning*

Ocan Sankur(✉)

Univ Rennes, Inria, CNRS, Rennes, France
ocan.sankur@cnrs.fr

Abstract. We present algorithms for model checking and controller synthesis of timed automata, seeing a timed automaton model as a parallel composition of a large finite-state machine and a relatively smaller timed automaton, and using compositional reasoning on this composition. We use automata learning algorithms to learn finite automata approximations of the timed automaton component, in order to reduce the problem at hand to finite-state model checking or to finite-state controller synthesis. We present an experimental evaluation of our approach.

1 Introduction

Timed automata [1] are a well-known formalism for modeling and verifying real-time systems. They can be used to model systems as finite automata, while using, in addition, clocks to impose timing constraints on the transitions. Using clock variables have advantages. They allow one to describe models that are expressive thanks to real-valued clock values; moreover, the use of specific clock variables enable optimizations such as sound and complete abstractions, also known as extrapolation operators [5]. Model checking algorithms have been developed and implemented in tools such as Uppaal [8], TChecker [28], PAT [50].

One approach for model checking timed automata is based on representing the set of clock values with *zones*, which are particular polyhedra, and using explicit enumeration on the discrete states. There has been extensive research on sound and complete abstractions on zones, which improved the performance of the model checking tools, and made it possible to handle models with more complex time constraints; see [11] for a survey. However this approach does not scale to models with large discrete spaces due to explicit enumeration. Several authors have developed algorithms to remedy this issue, and to attempt to adapt efficient model checking techniques finite-state systems to timed systems. Extensions of binary decision diagrams (BDD) with clock constraints have been considered both for continuous time [53,10,23] and discrete time [42,51]. Another approach is to use predicate abstraction on clock variables that enables efficient finite-state verification techniques based on BDDs or SAT solvers [17,16,46].

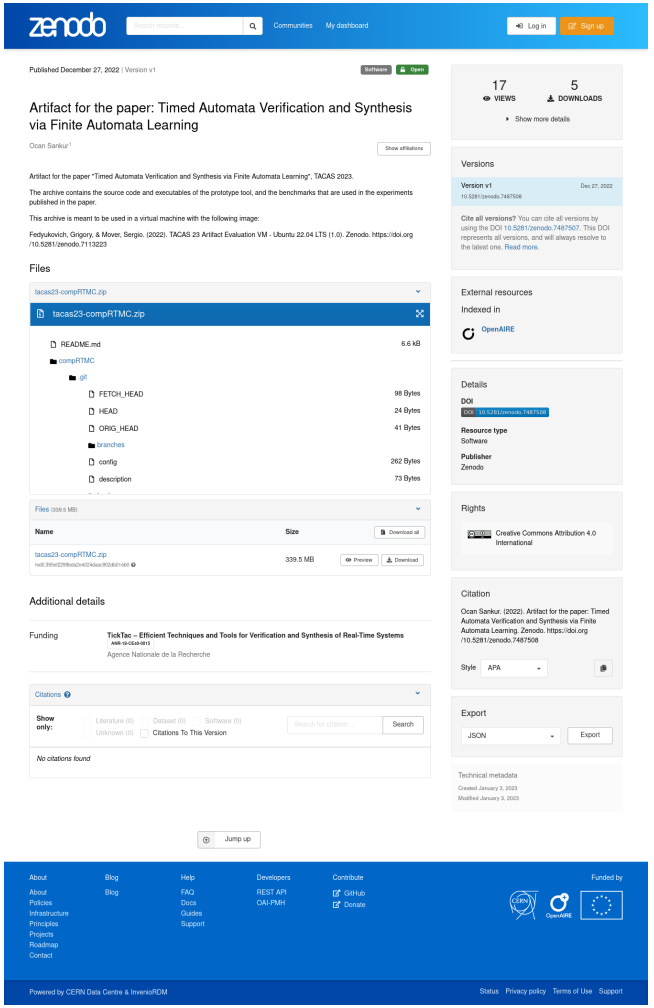
Controller synthesis is a related problem in which some transitions of the system are controllable and some are uncontrollable, and the objective is to

* This work was partially funded by ANR project Ticktac (ANR-18-CE40-0015).

S. Sankaranarayanan and N. Sharygina (Eds.): TACAS 2023, LNCS 13994, pp. 329–349, 2023.
https://doi.org/10.1007/978-3-031-30820-8_21

Article

10.1007/978-3-031-30820-8_21



Données

10.5281/zenodo.7487508

Une bonne pratique de la science ouverte

1. placement des données sur Zenodo, obtention d'un DOI
2. intégration du DOI dans l'article
3. version de l'article placé sur arXiv (version auteur ou *preprint*)
4. données bib importées vers HAL
5. publication d'un *tool paper* référençant l'outil et les données
6. publication d'un *data paper* décrivant les données

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB
- un disque dur externe

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB
- un disque dur externe
- le serveur d'une université

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB
- un disque dur externe
- le serveur d'une université
- un entrepôt (tel que *Software Heritage*)

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB
- un disque dur externe
- le serveur d'une université
- un entrepôt (tel que *Software Heritage*)
- le serveur d'un laboratoire de recherche

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB
- un disque dur externe
- le serveur d'une université
- un entrepôt (tel que *Software Heritage*)
- le serveur d'un laboratoire de recherche
- un Google drive

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB
- un disque dur externe
- le serveur d'une université
- un entrepôt (tel que *Software Heritage*)
- le serveur d'un laboratoire de recherche
- un Google drive
- une forge logicielle (telle que GitHub)

Quiz

Parmi les supports suivants, quels sont ceux adaptés au partage à long terme de données de la recherche ?

- une clé USB
- un disque dur externe
- le serveur d'une université
- un entrepôt (tel que *Software Heritage*)
- le serveur d'un laboratoire de recherche
- un Google drive
- une forge logicielle (telle que GitHub)
- un cloud répliqué en 3 endroits géographiques différents (tel que PCloud ou Dropbox)

Outline

Perspectives

Grandes questions

- manque encore de facilité à reproduire les expériences
  - reproduisez des *artifacts* existants
  - participez à des *artifact evaluation committee* (Formal Methods 2026 !)
- dépendances aux plateformes privées
  - risques de fermetures, rachats, purges…
- comment exécuter un code de 2025 dans 20 ans ?
  Dans 50 ans ? Dans 100 ans ?

Perspectives (version « le monde part en live »)

Grandes questions (de moins en moins irréalistes)

- sécurité physique des données
  - quid d'un conflit mondial majeur ?
- risque de corruption des institutions
  - quid si un gouvernement fasciste ferme ou expurge les serveurs de HAL, Zenodo, arXiv…?

Bibliographie

Bibliographie

- Ouvrir les données de recherche en informatique théorique : qu'a-t-on à y gagner ? (2023)

  10.60538/ouvrir_dr_informatique_theorique

Pour aller plus loin

Ressources généralistes

- Recherche Data Gouv
- DoRANum
- Rubrique « données » du site Science Ouverte de Couperin
- Mini-guide « Partager les données liées aux publications scientifiques – Guide pour les chercheurs » `https://www.ouvrirlascience.fr/partager-les-donnees-liees-aux-publications-scientifiques-guide-pour-les-chercheurs/`

Références supplémentaires

- Le guide des licences ouvertes de DoRANum
- Rédiger un plan de gestion de données et de logiciels avec DMP OPIDoR
- Les principes de l'ACM pour l'*artifact review* : `https://www.acm.org/publications/policies/artifact-review-and-badging-current`
- Plesser : Reproducibility vs. Replicability: A Brief History of a Confused Terminology (2018)
- NFDIxCS (National Research Data Infrastructure for and with Computer Science)

Références I

[And+22] Étienne André, Dylan Marinho, Laure Petrucci et Jaco van de Pol. « Efficient Convex Zone Merging in Parametric Timed Automata ». In : *FORMATS* (12-17 sept. 2022). Sous la dir. de Sergiy Bogomolov et David Parker. T. 13465. Lecture Notes in Computer Science. Warsaw, Poland : Springer, 2022, p. 200-218. DOI : 10.1007/978-3-031-15839-1_12.

[And+23] Étienne André, Shuang Liu, Yang Liu, Christine Choppy, Jun Sun et Jin Song Dong. « Formalizing UML State Machines for Automated Verification – A Survey ». In : *ACM Computing Surveys* 55.13 (juill. 2023), 277 :1-277 :47. DOI : 10.1145/3579821.

[And10] Étienne André. « IMITATOR II : A Tool for Solving the Good Parameters Problem in Timed Automata ». In : *INFINITY* (21 sept. 2010). Sous la dir. d'Yu-Fang Chen et Ahmed Rezine. T. 39. Electronic Proceedings in Theoretical Computer Science. Singapore, sept. 2010, p. 91-99. DOI : 10.4204/EPTCS.39.7.

[Che+15] Xin Chen, Stefan Schupp, Ibtissem Ben Makhlouf, Erika Ábrahám, Goran Frehse et Stefan Kowalewski. « A Benchmark Suite for Hybrid Systems Reachability Analysis ». In : *NFM* (27-29 avr. 2015). Sous la dir. de Klaus Havelund, Gerard J. Holzmann et Rajeev Joshi. T. 9058. Lecture Notes in Computer Science. Pasadena, CA, USA : Springer, 2015, p. 408-414. DOI : 10.1007/978-3-319-17524-9_29.

[Col+20] Giovanni Colavizza, Iain Hrynaszkiewicz, Isla Staden, Kirstie Whitaker et Barbara McGillivray. « The citation advantage of linking publications to research data ». In : *PLoS One* 15 (4 2020). DOI : 10.1371/journal.pone.0230416.

Références II

[Kor+21] Fabrice Kordon, Lom-Messan Hillah, Francis Hulin-Hubard, Loïg Jezequel et Emmanuel Paviot-Adet. « Study of the efficiency of model checking techniques using results of the MCC from 2015 To 2019 ». In : *International Journal on Software Tools for Technology Transfer* 23.6 (2021), p. 931-952. DOI : 10.1007/s10009-021-00615-1.

[Le +22] Mariannig Le Béchec, Aline Bouchard, Philippe Charrier, Claire Denecker, Gabriel Gallezot et Stéphanie Rennes. *Pratiques et usages des outils numériques dans les communautés scientifiques en France*. Research Report. Comité pour la science ouverte, jan. 2022. DOI : 10.52949/5.

[Liu+13] Shuang Liu, Yang Liu, Jun Sun, Manchun Zheng, Bimlesh Wadhwa et Jin Song Dong. « USMMC : A self-contained model checker for UML state machines ». In : *ESEC/FSE* (18-26 août 2013). Sous la dir. de Bertrand Meyer, Luciano Baresi et Mira Mezini. Saint Petersburg, Russian Federation : ACM, 2013, p. 623-626. DOI : 10.1145/2491411.2494595.

[Sut17] Geoff Sutcliffe. « The TPTP Problem Library and Associated Infrastructure - From CNF to THo, TPTP v6.4.o ». In : *Journal of Automated Reasoning* 59.4 (2017), p. 483-502. DOI : 10.1007/s10817-017-9407-7.

Remerciements

- Laëtitia Bracco : Conservatrice des bibliothèques, *Data librarian* à l'Université de Lorraine
- Olivier Lu : ingénieur pédagogique, Urfist-Lyon. Médiatisation numérique
- David Bernal : ingénieur pédagogique, Université Sorbonne Paris Nord
- Relecture disciplinaire : Fabrice Kordon, Engel Lefaucheux, Stephan Merz
- Beta-tester cours en ligne : Dylan Marinho

Licence de ce document

Auteur : Étienne André

Merci à Cyril Banderier pour sa relecture

(Source LaTeX disponible aux enseignant·e·s sur demande)

Nantes Université

Laboratoire des Sciences du Numérique de Nantes

Version : 18 novembre 2025