Informal Prospects in Formal Methods

Hélène Kirchner
Inria

Journées « Futur de l'Informatique »

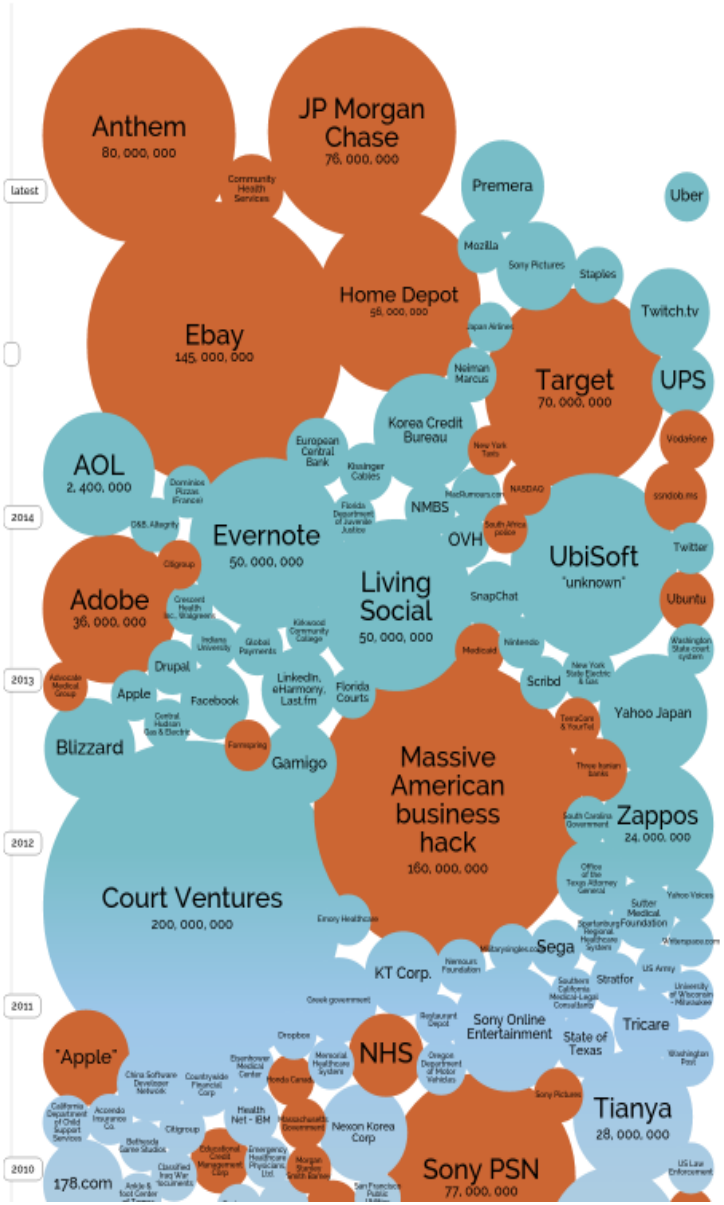
Grenoble, 05 Avril 2018

CYBERSECURITY STRATEGY OF THE EU

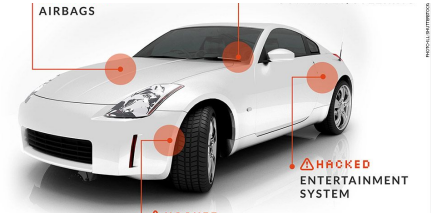
“For new connected technologies to take off, including e-payments, cloud computing or machine-to-machine communication, citizens will need trust and confidence. Unfortunately, […] almost a third of Europeans are not confident in their ability to use the internet for banking or purchases”

CONNECTED OBJECTS

- **2016 5.5 BILLION**
- **2020 20.8 BILLION**



What do we want to trust ?



Credit card, passport

Phone, Computer

Car, Plane

Medical devices (pacemaker)

Infrastructures: water, electricity, public transports,

Internet, IoT

Electronic voting


iNods
Lulu
BETA
blish
The easiest place to buy or sell digital content
flagr
FireAnt
simply hired
Veoh
theadcloud
rbloc.com
gather
cafepress
Renkoo
pixagogo
standpoint
meebo
EXTRA TASTY!
last.fm
YEDDA
dabble
Writeboard
SHOUTWIRE
KaNOODLE
AirSet
gOFFICE
zigtag
Findory
backfence
Zoozio
blogbeat
Ziggs
Boltfolio
wink
AllPeers
riya
Audible Wordcast
Opinity
Because your reputation matters!
reddit
measure map
gumshoo
bluepulse
imvu
BETA
STREAMLOAD
Freedom for Your Digital Lifestyle
nativetext
CONGOO
BETA
Google
FeedTier
phanfare
Fruitcast
PubSub
dPolls
flickr
zoominfo
People, Companies, Relationships
W Hazel St
N Rose St
W Bella St
N 32nd St
0.30
miles
2:00
33 mph
1:54
ProjectSpaces
FeedBurner
Bloglines
purevolume
Yub.com
Spot Runner
Allmydata.com
gabt
a social
YouTube
TM
riffs
ajchat
alpha
Jambo
NETWORKS
PROTOPAGE
ClipShack
beta
squishr
WebJay
PLAZES
Noodly
box
Jots
Xdrive
NETFLIX

What are Formal Methods useful for?

Formal modeling and verification methods have been successful in improving the safety and security of software in areas such as aeronautics : formal methods are no longer theoretical artifacts, but have shown that they can contribute significantly in industrial sectors where software correctness and certification is a concern *PS 2018-2022 Inria*

Dual use of Formal Methods

- to build reliable software: give trust and certify software
- to analyse and understand existing software or processes: provide explainability and accountability

What are Formal Methods useful for?

- Chasing bugs in software : bugs are source of vulnerabilities
- Modeling environment (conforming, adversarial, uncertain, unknown) and expected (mis-)behaviour
- Risk analysis : accidental or intentional threats, faults or attacks
- Recovery mechanisms
- Certification
- (Safety / Security / Privacy) by design

Cyber systems requirements

with impact on user's trust and empowerment:

- Safety
- Security
- Privacy
- Transparency
- Accountability
- Certification
- Ethics

01

Safety

Formal modeling and verification methods have been successful in improving the safety and security of software in areas such as aeronautics…

…it is essential to improve on the foundations and interconnection of tools and formalisms for interactive and automated program verification such as Coq, Why3, F*, TLA+, as well as various static analyzers, such as Astrée. The automation and the expressivity of these tools must be improved so that they **can scale to the verification of larger software systems, and certify both qualitative and quantitative properties**.

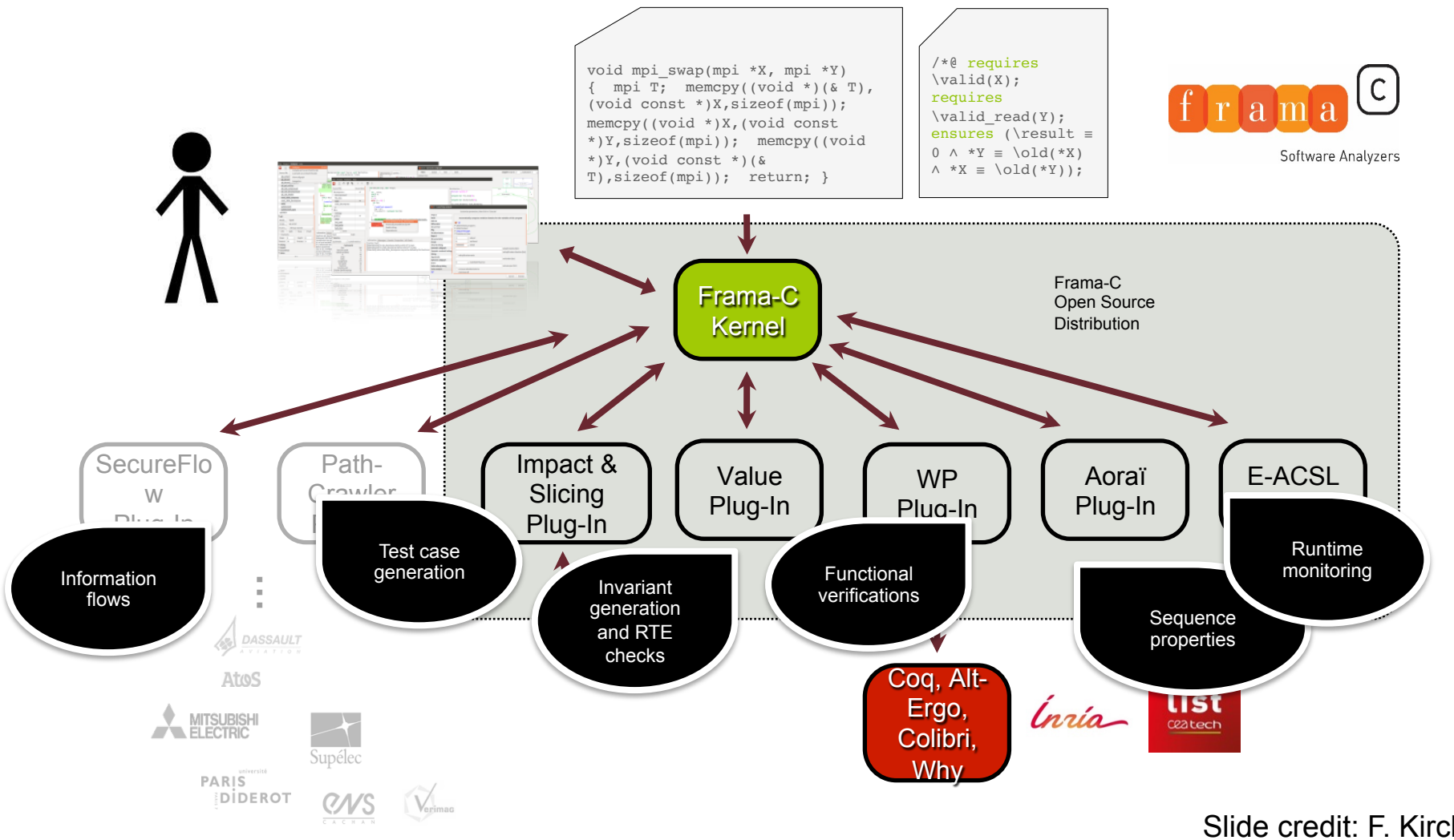
PS 2018-2022 Inria

More infrastructure

[A] significant part of the effort in existing projects was spent on the further development of verification tools, on formal models for low-level programming languages and paradigms, and on general proof libraries... *Future efforts will be able to build on these tools and reach far-ranging verification goals faster, better, and cheaper.*

Gerwin Klein, Formal OS Verification—An Overview

CODE ANALYSES



Slide credit: F. Kirchner

Ever-running software systems

PS 2018-2022 Inria

Ex: smart phones, automated buildings, railway systems, and energy distribution systems…

In all these contexts, software must run continuously, while maintaining its ability to evolve, thus addressing new needs and requirements, technology changes, and bug fixes.

Challenges due to complex production environments, software that evolve dynamically, reliability and security constraints required.

Ever-running software systems

PS 2018-2022 Inria

- How to express ever-running software systems (specification, programming language, integrated development environments, etc.)?
- How to process their definitions (compilation, verification, instrumentation, etc.)?
- How to run them (monitoring, libraries, runtime support, etc.)?
- How to make them evolve (introspection, runtime code generation,self-adaptation and autonomic computing, reconfiguration control, etc.)?

02

Security

Cybersecurity relies on

- **Cryptography**: confidentiality, integrity, anonymity, authenticity
- **Security Policy**: set of rules that specify how sensitive and critical resources are protected
- **Prevention**: to early detect vulnerabilities
- **Cyber-resilience** : capacity to tolerate attacks (hardware and software), to detect malware
- **Security by design**

Towards provable security

CryptoVerif
EasyCrypt
CertiCrypt

- cryptographic **primitives** - mathematical proofs, use theorem proving and program verification to achieve computer checked proofs.
- automated verification tools to analyze the **protocol** specifications and find vulnerabilities in the protocol logic
- producing verified **implementations**.

miTLS: A Verified Reference Implementation of TLS

https://mitls.org/

Malware analysis

Needs to identify

- the targets of this malware (a particular end-user, a company, any machine under a specific operating system, etc.)
- the actions it intends to perform to attack the targets (sensitive information leakage, encryption and ransom, etc.),
- the way it succeeds to bypass the security mechanisms protecting the targets, the way it protects itself against malware detection engines (obfuscation).

Methods : automatic classification, reverse malicious code, static analysis, deobfuscation, morphological analysis based on control flow graph comparison

03

Privacy

Privacy

Ability for individuals to control their personal data and decide what to reveal to whom and under what condition.

Privacy leaks :

- Social networks
- Geolocation information
- Web tracking
- Smart world
- Internet : wireless access networks, core internet services,malicious web site detection systems

Privacy and GDPR

French and European regulations (European General Data Protection Regulation, GDPR) define personal data, Sensitive Personal Information, user empowerment and responsibility of all stakeholders.

Obligation:

- To conduct data protection impact assessments
- To implement privacy by design
- To comply with the regulations
- To enforce user empowerment through control (consent) and transparency.

Trade-off between privacy and utility

Privatics

A formal framework based on epistemic logic to express data minimization requirements as properties defining for each stakeholder the information that she is (or is not) allowed to know

- a language to define privacy architectures
- a logic for reasoning about architectures.
- axiomatization to prove that a given architecture meets the expected privacy and integrity requirements.

applied in particular to compare different architectures for biometric access control and to provide a rationale for the choice of specific options.

Formal methods and Privacy

- **Data anonymization**: detect private information that can be inferred about individuals using the anonymized data with prior (or background) knowledge
- **Differential privacy** : protect an individual's data while publishing aggregate information. Capability for users to obfuscate their personal data, adding noise by themselves
- Empowering users with **personal clouds** : individualized management and control over one's personal data. Ensure security and extensibility
- **Privacy preserving protocols** and communication technologies : **Homomorphic and functional encryption schemes** to operate on encrypted data, and **proofs of knowledge** to get evidence that outsourced computation is performed correctly

04

Transparency

Transparency

TransAlgo : une plate-forme scientifique pour juger de la transparence des algorithmes

Inria, IMT, CNRS, en coopération avec le Conseil National du Numérique (CNNum), la Direction Générale de la Concurrence, de la Consommation et de la Répression des Fraudes (DGCCRF) et la Direction Générale des Entreprises (DGE)

Un algorithme est transparent si l'on peut facilement vérifier sa « responsabilité », par exemple, s'il ouvre son code, s'il explicite à la fois la provenance des données qu'il a utilisées, et celles qu'il produit, s'il explique ses résultats, ou encore s'il publie des traces de ses calculs. Notons que nous considérerons aussi les situations où le code n'est pas ouvert car il n'y a aucune obligation de divulgation de celui-ci.

https://www.inria.fr/actualite/actualites-inria/transalgo?utm_content=buffera72f5&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer

ML-based Automated Decision Making

FM strongly contribute to ensure safety, security and accountability of software and hardware systems.

Yes, but…

ML-based Automated Decision Making systems differ fundamentally from prior computer applications. Automated Decision Making systems will make mistakes. The assumption that computers are accurate and nearly infallible, while generally appropriate for tasks such as bookkeeping, is dangerously incorrect for ADM systems.

When Computers Decide: European Recommendations on machine-Learned Automated Decision Making

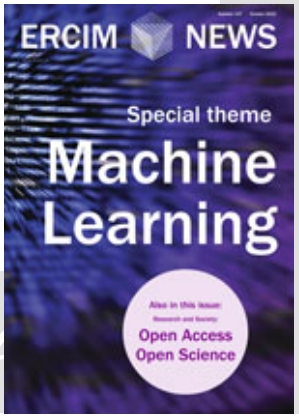
Informatics Europe and EU-ACM, 2018

Machine Learning and Automated Decision Making

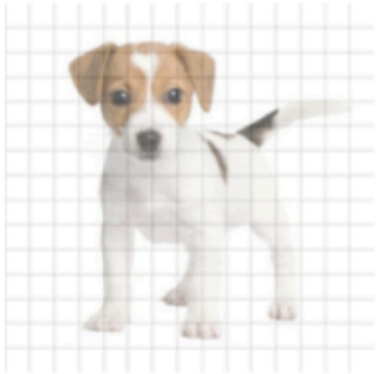
How to program computers to automatically recognise complex patterns and make intelligent decisions based on data?

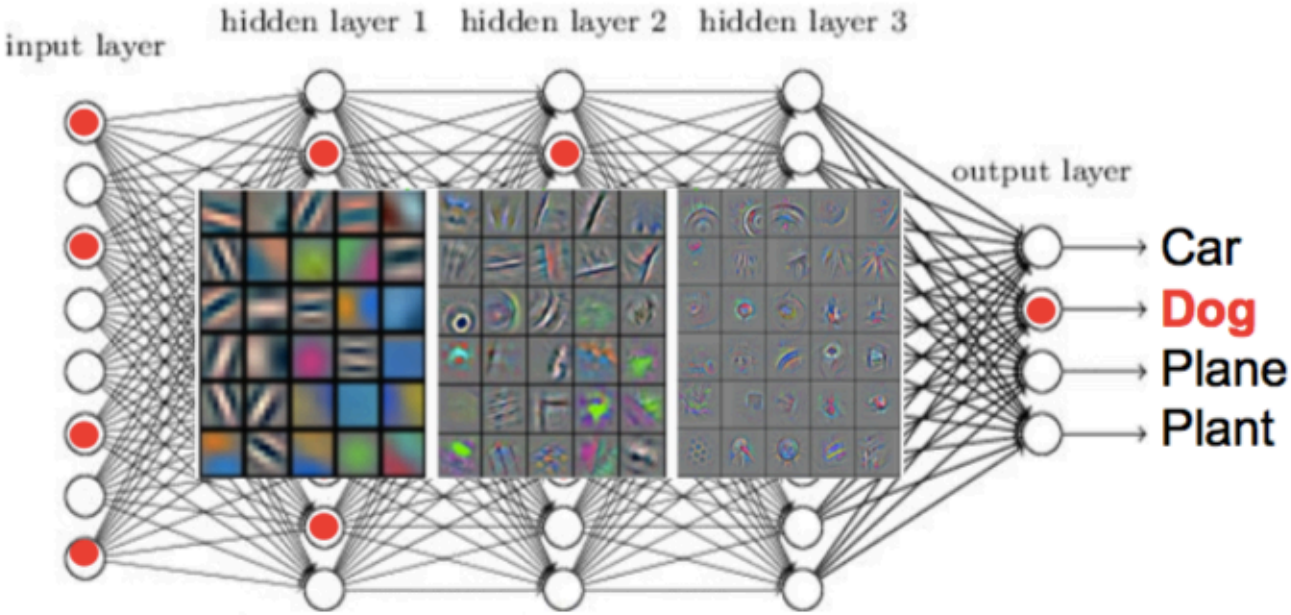
Applications: vision, language processing, forecasting, games, data mining, expert systems and robotics.

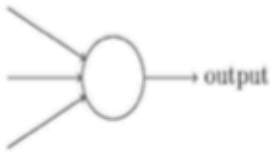
- Logical approach: symbolic rules in expert systems, first automated decision making. How to deal with uncertainty ?
- Learning from data: (un-)supervised ML, reinforcement, deep learning based on neural networks



Deep neural networks







$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

Slide credit: A.Joly

Formal methods and Deep Neural Network

- identify weaknesses of deep learning strategies
- analyse what kinds of attacks are possible and propose mechanisms for protecting convolutional neural networks against adversarial attacks.
- monitor the internal activations that flow between the layers of a deep network
- make the overall process more robust
- Privacy issues: use of machine learning techniques to infer possibly sensitive data, can an attacker who has access to the trained network gain information about the training data?

05

Accountability

Accountability

Principle which requires that organizations put in place appropriate technical and organizational measures and are able to demonstrate their compliance with the regulation.

Example and challenges of blockchain

Blockchain

A blockchain implements a secure electronic ledger:

- data registered in the ledger cannot be removed or modified
- integrity of past history relies on cryptographic hash functions: the hash of the last trusted block certifies the integrity of the whole ledger since its inception.
- the role of certifying the blockchain is decentralized
- blockchain is replicated between participants: all must have the same view of the blockchain to avoid attacks

Yet, their real security and level of trust need to be properly asserted with analysis both from the cryptography and distributed systems communities.

K Framework Enables Verification of Smart Contracts

https://runtimeverification.com/blog/

Grigore Rosu's Formal Systems Laboratory (FSL) at UIUC and Runtime Verification (RV) have used the K framework to successfully build and test a mathematical model of the Ethereum Virtual Machine, which makes it possible to formally verify the accuracy of smart contracts.

We present recent academic and commercial results in developing blockchain languages and virtual machines that come directly equipped with formal analysis and verification tools. The main idea is to generate all these automatically, correct-by-construction from a formal specification. We demonstrate the feasibility of the proposed approach by applying it to two blockchains, Ethereum and Cardano.

Grigore Rosu

Invited talk at FSCD 2018

http://www.cs.le.ac.uk/events/fscd2018/

Formal Methods and Blockchain

Low-level network attacks: a threat assessment model needs to be established and adapted to each blockchain technology.

Blockchain-specific network policies must be defined, deployed and verified automatically

Software failures : as any software system, due to bugs, attacks, undefined behavior, and so on.

Blockchains as building bricks for higher level protocols : providing a ledger, on which higher level programs and protocols can be implemented.

Privacy lacking by default: ensure privacy using zero-knowledge proofs and other advanced cryptography

06

Certification

Certification

Certification refers to evaluation of the level of confidence in the strength of a product, system, solution, service or organisation

It relies on

- Norms and standards
- Graduation of cybersecurity levels
- Audits

Related to sovereignty: the European case

Cyber-Physical Systems

CPS are involved in SCADA applications, in medical devices, in robotics, part of Internet of Things (IoT)

Characteristics:

- autonomy and dynamicity, self-reconfigurable
- part of critical systems
- resources limited

Formal methods and Cyber-Physical Systems

Challenges of building a safe, efficient and secure CPS:

- semantic formalisms, programming paradigms, modeling and verification techniques for hybrid systems. mixing discrete-time and continuous-time dynamics.
- certification of the underlying software infrastructures and operating systems
- resource-constrained devices : to certify properties of quantitative nature (use of resources such as time and energy).
- lightweight cryptography : efficient, certified cryptographic primitives for resource-constrained devices

07

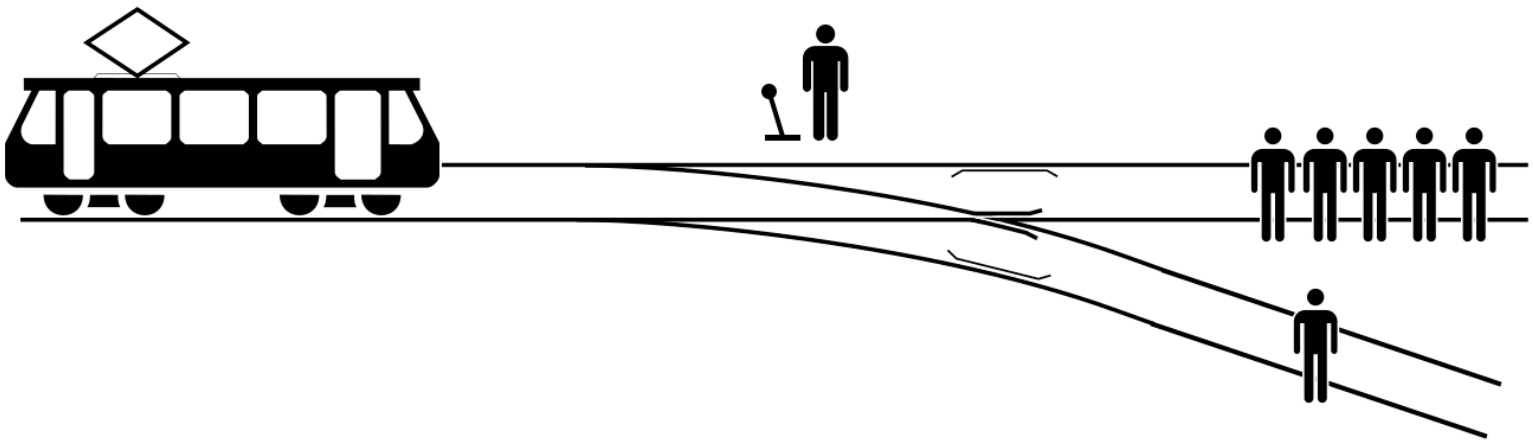
Ethics

Ethics

Ethical reasoning : reasoning taking into account ethical considerations (principals that govern a person's behaviour or the conducting of an activity)

Define a formalism to:

- drop ambiguities of natural language
- allow the computation of judgements helping decision making of operator



The trolley problem: should you pull the lever to divert the runaway trolley onto the side track?

By Zapyon - Own work based on: Trolley problem.png & BSicon TRAM1.svg, Rozjazd pojedynczy.svgThis file was derived from:Trolley problem.png:BSicon TRAM1.svg:Rozjazd pojedynczy.svg:Person icon BLACK-01.svg:, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=67107784

Formal methods and Ethics

Identify and formalise a number of values and concepts for an ethical reasoning

Formalise ethical dilemna (how to characterise a dilemna?)

Design a reasoning framework:

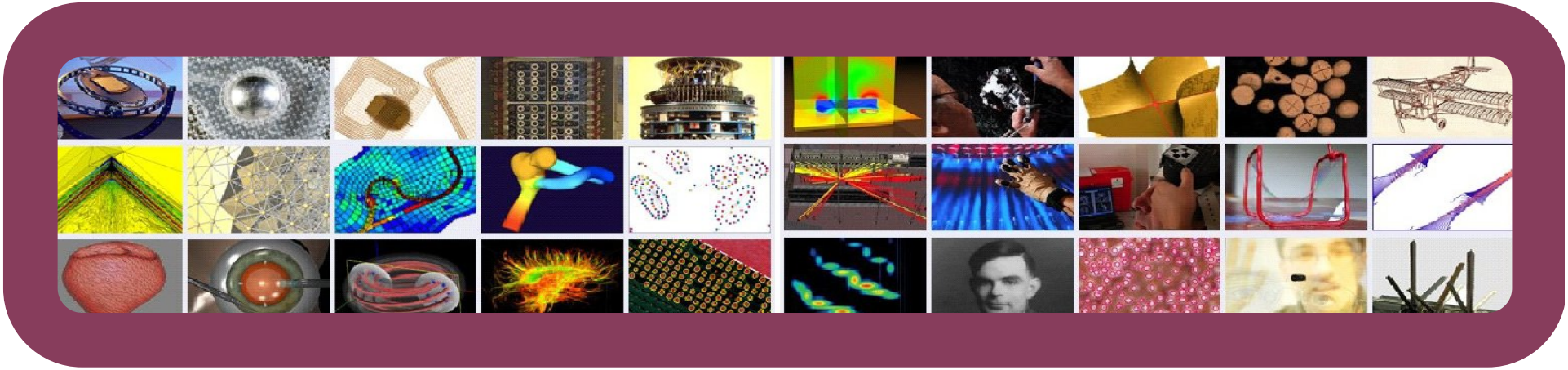
- A system of values: judgement may need a partial order on values
- Take into account several points of view, identify sources of subjectivity
- Take into account uncertainty in real world

Validate through experimentation with human participants

Wrap-up : Ethics for Cyber Systems

Hierarchy of values :

- Safety
- Security
- Privacy
- Transparency
- Accountability
- Certification



Merci

helene.kirchner@inria.fr