

Formal Methods and Tools: Techniques and Experiences

Maurice H. ter Beek

FMT lab, ISTI-CNR, Pisa, Italy

<http://fmt.isti.cnr.it/>

Huawei Symposium on Foundations of Software
16-17 January, Paris, France

Introduction

About me

- Researcher in the Formal Methods and Tools (FMT) lab at CNR–ISTI since 2003
- M.Sc. and Ph.D. (2003) degrees from Leiden University in The Netherlands
- Formal methods, model checking, SPLE, SOC, team automata

About me

- Researcher in the Formal Methods and Tools (FMT) lab at CNR–ISTI since 2003
- M.Sc. and Ph.D. (2003) degrees from Leiden University in The Netherlands
- Formal methods, model checking, SPLE, SOC, team automata





Italy's National Research Council (CNR)

- Est. 1923, full range of scientific areas, annual budget 1B€
- More than 9000 FTE ($\frac{2}{3}$ scientists), >2000 junior scientists
- Mission: perform research, training, technology transfer, promote innovation and competitiveness in industry and society, advise the government and other public bodies

Italy's National Research Council (CNR)

- Est. 1923, full range of scientific areas, annual budget 1B€
- More than 9000 FTE ($\frac{2}{3}$ scientists), >2000 junior scientists
- Mission: perform research, training, technology transfer, promote innovation and competitiveness in industry and society, advise the government and other public bodies
- **Institute of Information Science and Technologies (ISTI), Pisa**



- Research campus: 123,300 m²
>1500 persons (\pm 1000 CNR)



Research Infrastructures

Research infrastructures are mainly located within the Research Parks, grouping together institutes with related scientific tasks, sharing common services. The use of some large research infrastructures is also made available to researchers belonging to other scientific institutions in Italy and abroad, such as marine vessels, or other facilities settled in remote locations (i.e. Svalbard islands and Himalaya region) for environmental research.

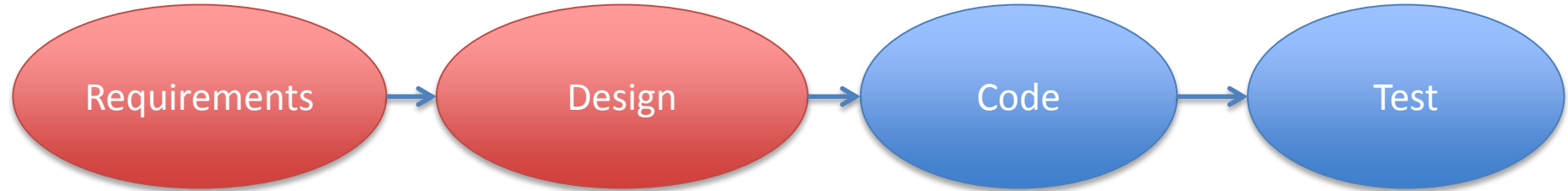


Formal Methods and Tools (FMT)



Formal Methods and Tools (FMT)

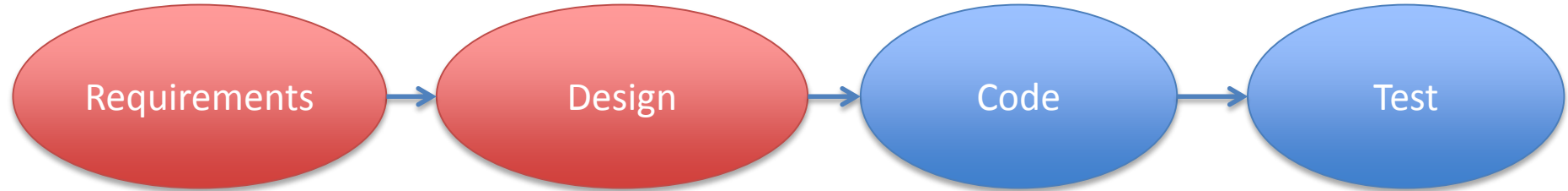
Software Development Process





Formal Methods and Tools (FMT)

Software Development Process



Model Checking

Product Line Engineering

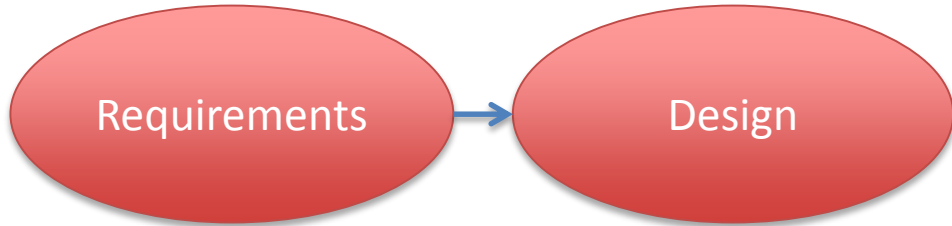
Natural Language Processing



Techniques and Tools

Application Domains

Formal Methods and Tools: Techniques



Natural Language Processing

Techniques and Tools

Natural Language Processing

The image displays the QuARS software interface, version 4.1, which is used for Natural Language Processing. The interface is divided into several sections:

- QuARS Dictionaries:** A panel on the left showing a list of words under the 'implicit' category, including 'Above', 'Below', 'Following', 'Previous', 'Them', 'These', 'This', and 'Those'. It includes buttons for 'New', 'Delete', 'Reload', 'Save', 'Clear', and 'Print'.
- QuARS Output:** A central panel displaying the results of a syntactic implicit analysis. It shows three examples of implicit sentences identified in a document, each with a line number and a description of the implicit element (e.g., 'implicit object', 'implicit determiner').
- QuARS Sentences Input file:** A panel at the bottom showing the input file 'requirements.txt' and a list of sentences from the document, numbered 1 through 7.

The QuARS Output panel contains the following text:

```
----- QuARS [Syntactic] implicit ANALYSIS -----  
  
The line number:  
 2. 222: each pp claim shall identify security objectives and it security requirements statements contain  
contains an implicit sentence: implicit object  
  
The line number:  
 115. 2564: the system shall provide the capability to maintain army universal task list (aut) informatio.  
contains an implicit sentence: implicit determiner.  
  
The line number:  
 124. 2577: the system shall facilitate the upkeep of help through a user profile. this requirement has n
```

The QuARS Sentences Input file panel shows the following sentences:

```
1 1. Requirements for the Functions and Performance of the System  
2 222: Each PP claim shall identify security objectives and IT security requirements statements contained in the ST that are in addition to those contained in the PP.  
3 417: The system shall provide capability to remove student from training.  
4 451: The system shall provide capability to deliver education/training products and materials, including safety materials, to the learner at home, units, training centre  
5 469: The system shall provide the capability to manage and manipulate the education/training catalog.  
6 479: The system shall maintain the education/training product catalog.  
7 489: The system shall provide a capability to search the education/training products catalog.
```

<http://quars.isti.cnr.it/>

Natural Language Processing

The screenshot displays the QuARS software interface, version 4.1. The main window is titled "QuARS [Syntactic] implicity ANALYSIS". The interface includes a menu bar (File, Edit, View, Analysis, Metrics & Logs, Options, ?), a toolbar with icons for file operations and analysis, and a main workspace. The workspace is divided into several panes: "QuARS Dictionaries: [Syntactic] implicity" (with sub-tabs for implicity, multiplicity, underspecification), "QuARS Output" (showing analysis results), "QuARS Sentences Input file: requirements.txt" (with a list of sentences), and a bottom toolbar with buttons for Load, ReLoad, Save, Save As, Clear, and Print.

Overlaid on the interface is a workflow diagram for the QuARS process:

- Input:** A yellow document icon labeled "sentences.txt" feeds into a light blue box labeled "Syntax Parser".
- Processing:** The "Syntax Parser" outputs to a light blue document icon labeled "Parsed.txt".
- Analysis:** The "Parsed.txt" file is processed by an orange box labeled "Lexical Parser" and a green box labeled "Indicators Detector".
- Output:** The "Indicators Detector" outputs to a green dashed box labeled "Log", which lists indicators: "metrics", "vague", "weak", "optional", "subjective", "multiple", "implicit", and "underspec".
- Support:** "Domain dictionaries" (represented by pink document icons) and "Indicator related dictionaries" (represented by purple document icons) provide input to the "Views derivation" (pink box) and "Indicators Detector" respectively.
- Visualization:** The "Views derivation" outputs to a "Graphics" window (represented by a line graph icon).

<http://quars.isti.cnr.it/>

QuARS at Work: Defect Examples

Type	Package	Responsibility	Source
Functional	Assist Troubleshooting	Schmalfeldt	TDS
FREQ1177		For each error message information concerning possible corrective measures and consequences shall be delivered.	

Lexical Analysis

Defects	Optionality	Subjectivity	Vagueness	Weakness
1	--	--	possible	--

Type	Package	Responsibility	Source
Functional	Converter cooling	Baddam	
FREQ1398		The cooling system delivers sufficient cooling to converter for all modes of operation, at a convenient pressure, volume and temperature. The cooling system and the conversion system are coupled with a convenient and defined interface (sensors, as fan speed, water/air temperatures, water flux, water conductivity, shall be present). If forced air cooling is applied, a filter with less maintenance effort shall be used.	

Lexical Analysis

Defects	Optionality	Subjectivity	Vagueness	Weakness
1	--	--	sufficient	--

QuARS at Work: Defect Examples

Type	Package	Responsibility	Source
Functional	Assist Troubleshooting	Schmalfeldt	TDS
FREQ1177		For each error message information concerning possible corrective measures and consequences shall be delivered.	

This usage of the vague word *possible* is not a defect

Lexical Analysis

Defects	Optionality	Subjectivity	Vagueness	Weakness
1	--	--	possible	--

Type	Package	Responsibility	Source
Functional	Converter cooling	Baddam	
FREQ1398		The cooling system delivers sufficient cooling to converter for all modes of operation, at a convenient pressure, volume and temperature. The cooling system and the conversion system are coupled with a convenient and defined interface (sensors, as fan speed, water/air temperatures, water flux, water conductivity, shall be present). If forced air cooling is applied, a filter with less maintenance effort shall be used.	

Lexical Analysis

Defects	Optionality	Subjectivity	Vagueness	Weakness
1	--	--	sufficient	--

QuARS at Work: Defect Examples

Type	Package	Responsibility	Source
Functional	Assist Troubleshooting	Schmalfeldt	TDS
FREQ1177		For each error message information concerning possible corrective measures and consequences shall be delivered.	

This usage of the vague word *possible* is not a defect

Lexical Analysis

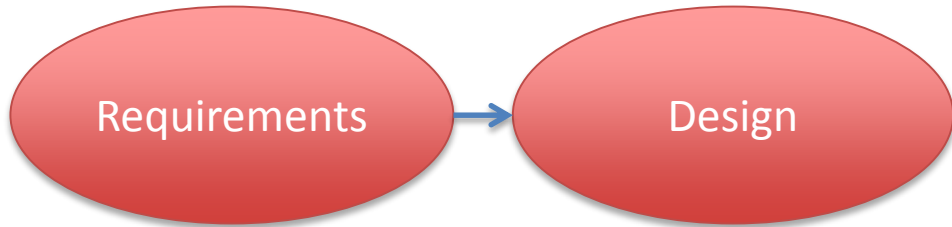
Defects	Optionality	Subjectivity	Vagueness	Weakness
1	--	--	possible	--

Type	Package	Responsibility	Source
Functional	Converter cooling	Baddam	
FREQ1398		The cooling system delivers sufficient cooling to converter for all modes of operation, at a convenient pressure, volume and temperature. The cooling system and the conversion system are coupled with a convenient and defined interface (sensors, as fan speed, water/air temperatures, water flux, water conductivity, shall be present). If air cooling is applied, a filter with less maintenance effort shall be used.	

This usage of the vague word *sufficient* might be an underspecification

Lexical Analysis

Defects	Optionality	Subjectivity	Vagueness	Weakness
1	--	--	sufficient	--



Techniques and Tools

“If debugging is the process of removing bugs, then programming must be the process of putting them in”

E.W. Dijkstra

Formal Methods

“Rigorous techniques, based on mathematical foundations, for the specification and verification of software (systems)”

Model Checking

- Automatically check whether a model satisfies a temporal logic property (LTL, CTL) and provide a counterexample if it does not
- Exhaustive, but suffers from the state space explosion problem
- BLAST, CADP, JPF, mCRL2, PRISM, (Nu)SMV, SPIN, Uppaal, ...

Probabilistic/Stochastic Model Checking (PMC)

- Model check whether a stochastic model satisfies a temporal logic property (PCTL, CSL) with a probability greater than a set threshold
- Model uncertainty/performance; do quantitative analysis (QoS, ...)
- CADP, LiQuor, MRMC, PARAM, PRISM, Uppaal PRO, ...

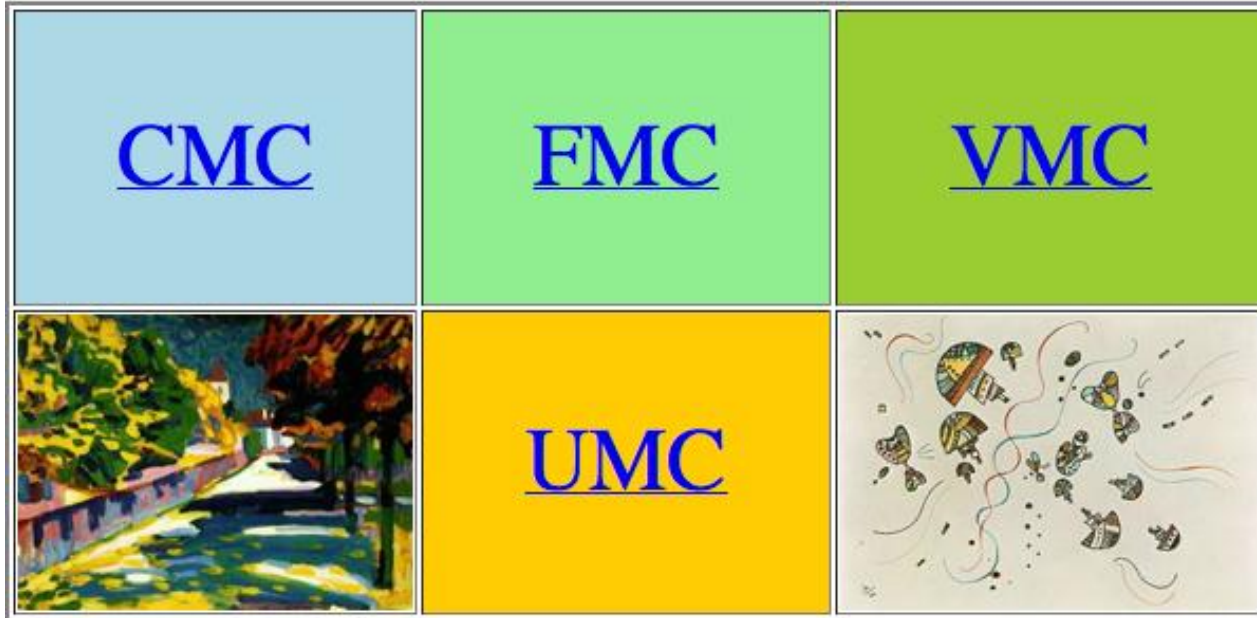
Statistical Model Checking (SMC)

- Simulation-based technique to statistically approximate (P)MC
- Highly parallelisable and automatable; tunable preciseness via CI
- PLASMA, PRISM, Uppaal SMC, (P)VeStA, MultiVeStA, QFLan, ...

Mean Field MC, Spatial/Spatio-temporal MC, ...

KandISTI

Family of model checkers developed by FMT for >2 decades

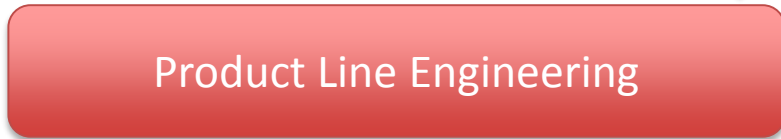
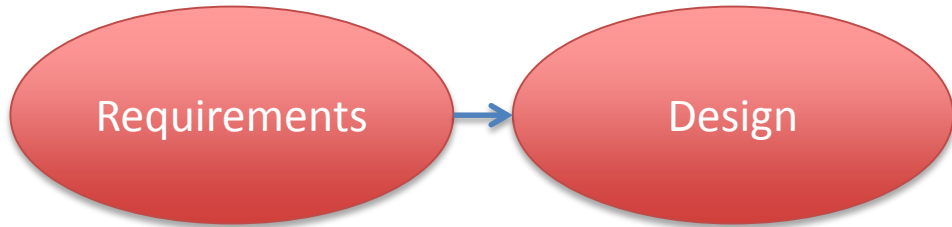


<http://fmt.isti.cnr.it/kandisti>

ACM TOSEM, SCP, JLAMP, FM, FASE, FMICS

Explicit-state on-the-fly model checking of properties in state- and action-based branching-time temporal logics
e.g. UCTL, SocL, v-ACTL

Complexity linear w.r.t. size of model and size of formula



Techniques and Tools

Software Product Line Engineering (SPLE)

Product: a valid combination (configuration) of features



Product line: a set of valid feature combinations of a domain



Software Product Line Engineering (SPLE)

Product: a valid combination (configuration) of features



Product line: a set of valid feature combinations of a domain



SPLE: develop and maintain a (software) product line using a shared architecture or platform (commonalities) and mass customisation (variabilities) to serve, e.g., different markets, thus facilitating (software) reuse

Scope: maximise commonalities whilst minimising cost of variations (i.e. of individual products)

Variability Modelling

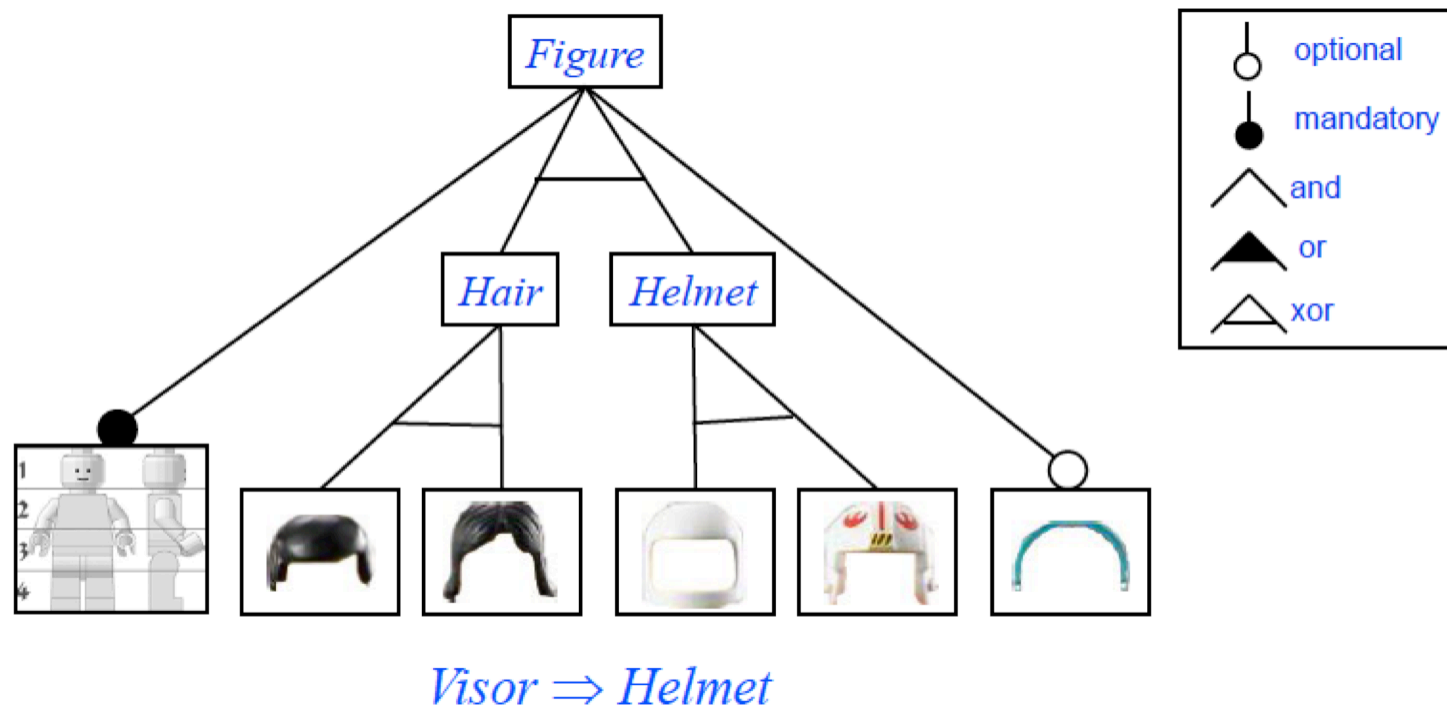
Variability in terms of features and constraints:

- Stakeholder-visible pieces of functionality of a system...
- ...which may be optional and/or may have alternatives
- Only specific feature combinations constitute products!

Variability Modelling

Variability in terms of features and constraints:

- Stakeholder-visible pieces of functionality of a system...
- ...which may be optional and/or may have alternatives
- Only specific feature combinations constitute products!



Feature model: compact representation of all products

Scalability

33 optional, independent
features



a unique product for every

person on this planet

(Behavioural) Variability Analysis

Rigorously establish critical system requirements (for quality assurance) with formal models and automated analysis tools

- For decades now successful in 'single' product engineering
- Not exploited broadly in SPLE, even though correctness of both artifacts for reuse and developed products is crucial!

(Behavioural) Variability Analysis

Rigorously establish critical system requirements (for quality assurance) with formal models and automated analysis tools

- For decades now successful in 'single' product engineering
- Not exploited broadly in SPLE, even though correctness of both artifacts for reuse and developed products is crucial!

Traditionally:

- Mainstream formal methods do *not* consider variability
- Formal methods that *have* been applied in SPLE typically focus on *structural* rather than on *behavioural* properties (feature model analysis, e.g. dead/false optional features)

Variability Analysis Strategies

Type checking, static analysis, model checking, theorem proving, testing

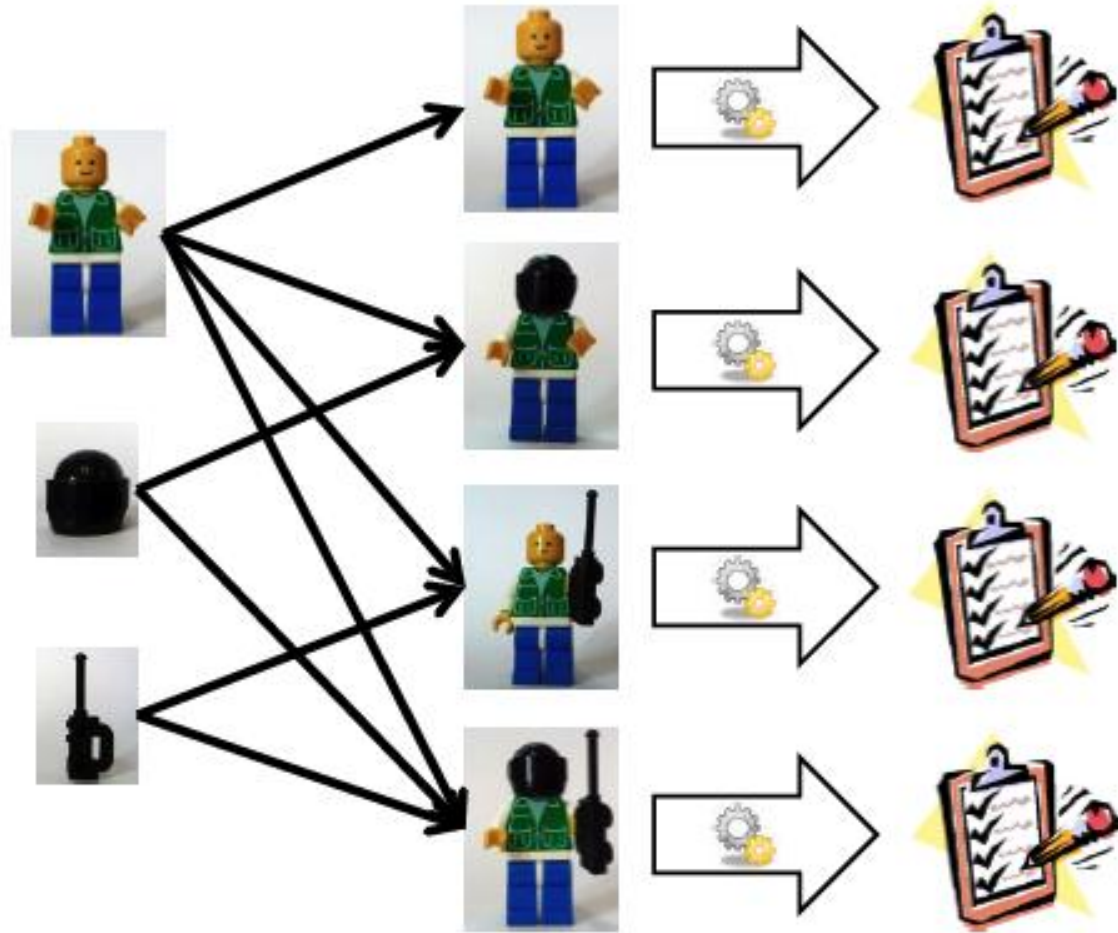


Product-Based Analysis

Family-Based Analysis

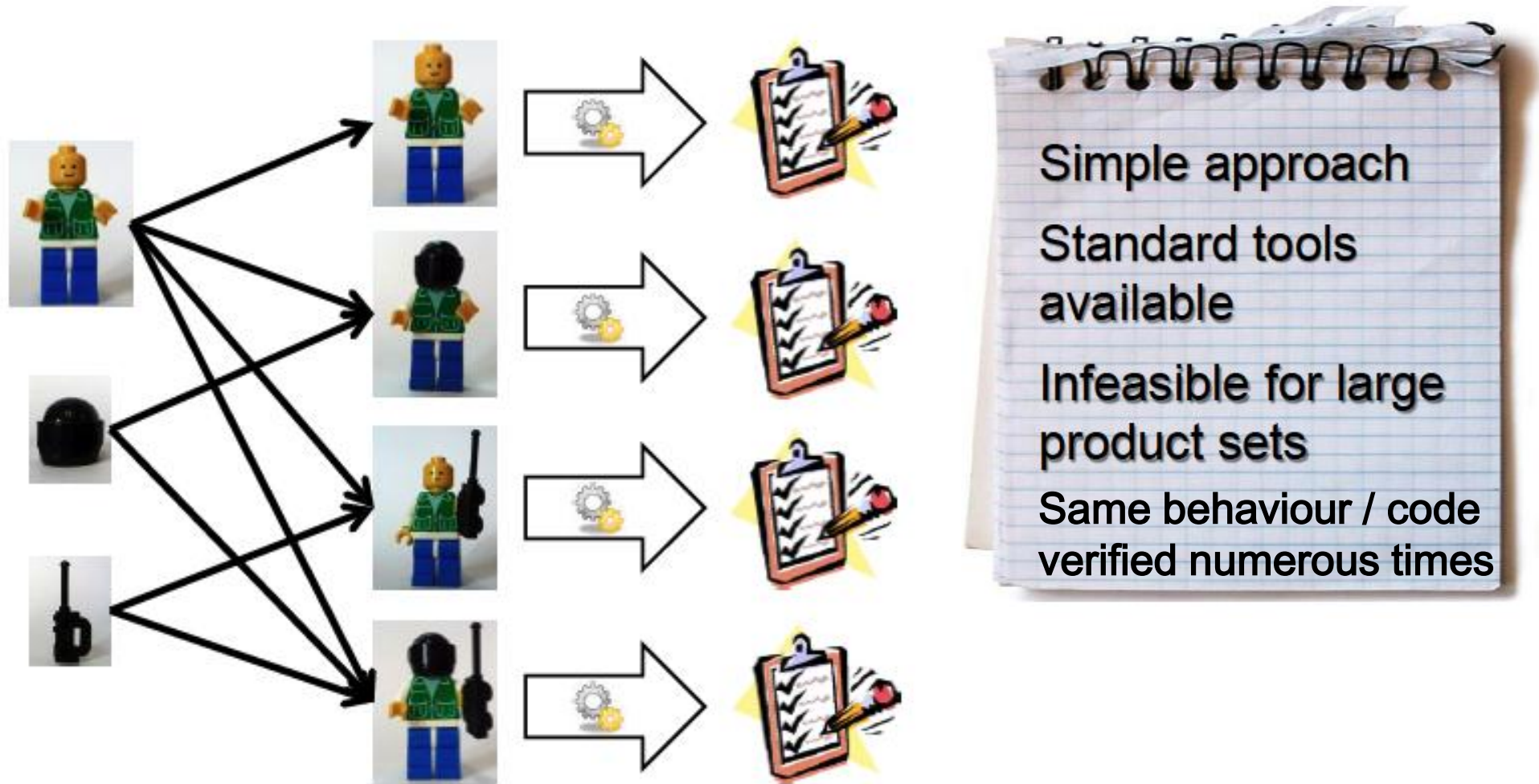
~~**Feature Based Analysis**~~

Product-based Analysis



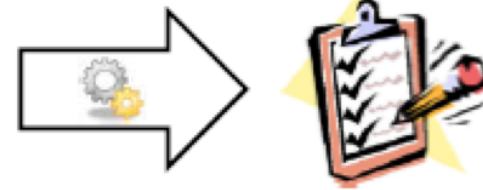
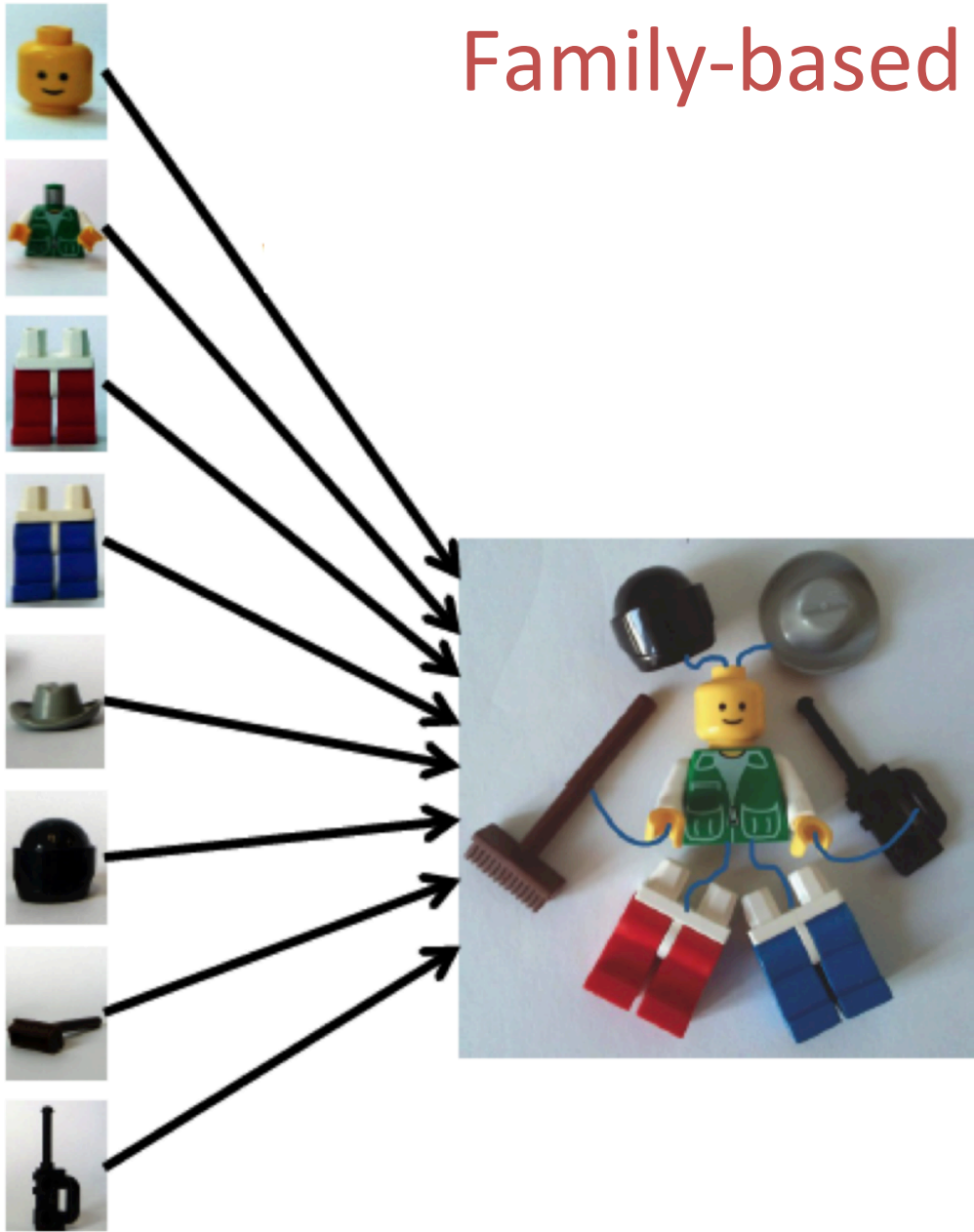
$O(2^n)$ for n features

Product-based Analysis

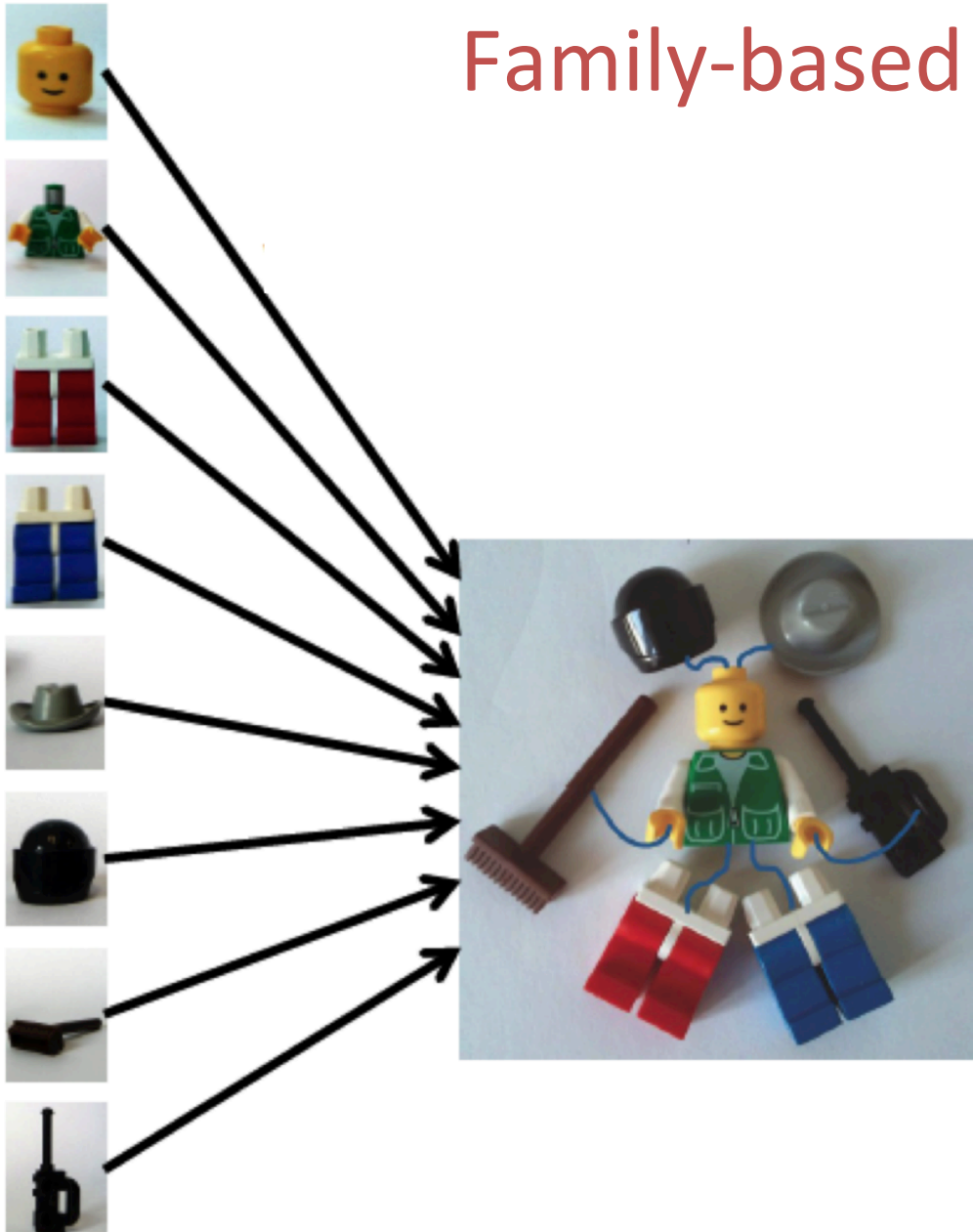


$O(2^n)$ for n features, each with large state space

Family-based Analysis



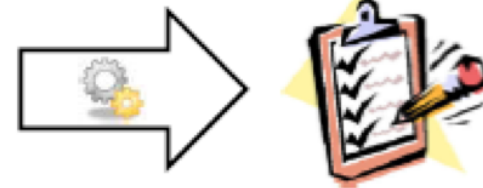
Family-based Analysis



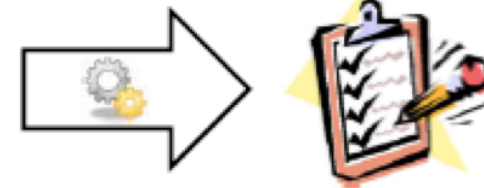
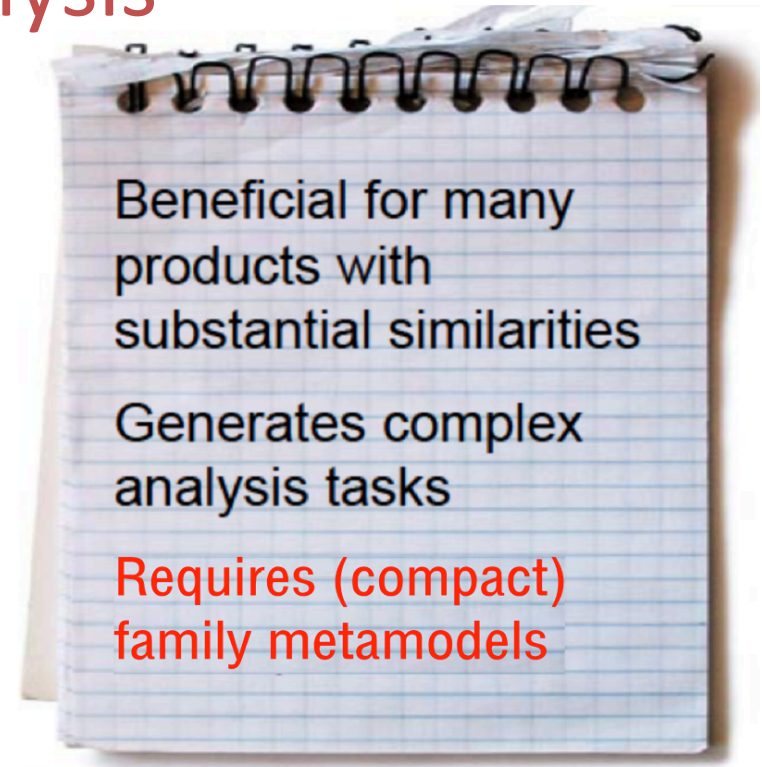
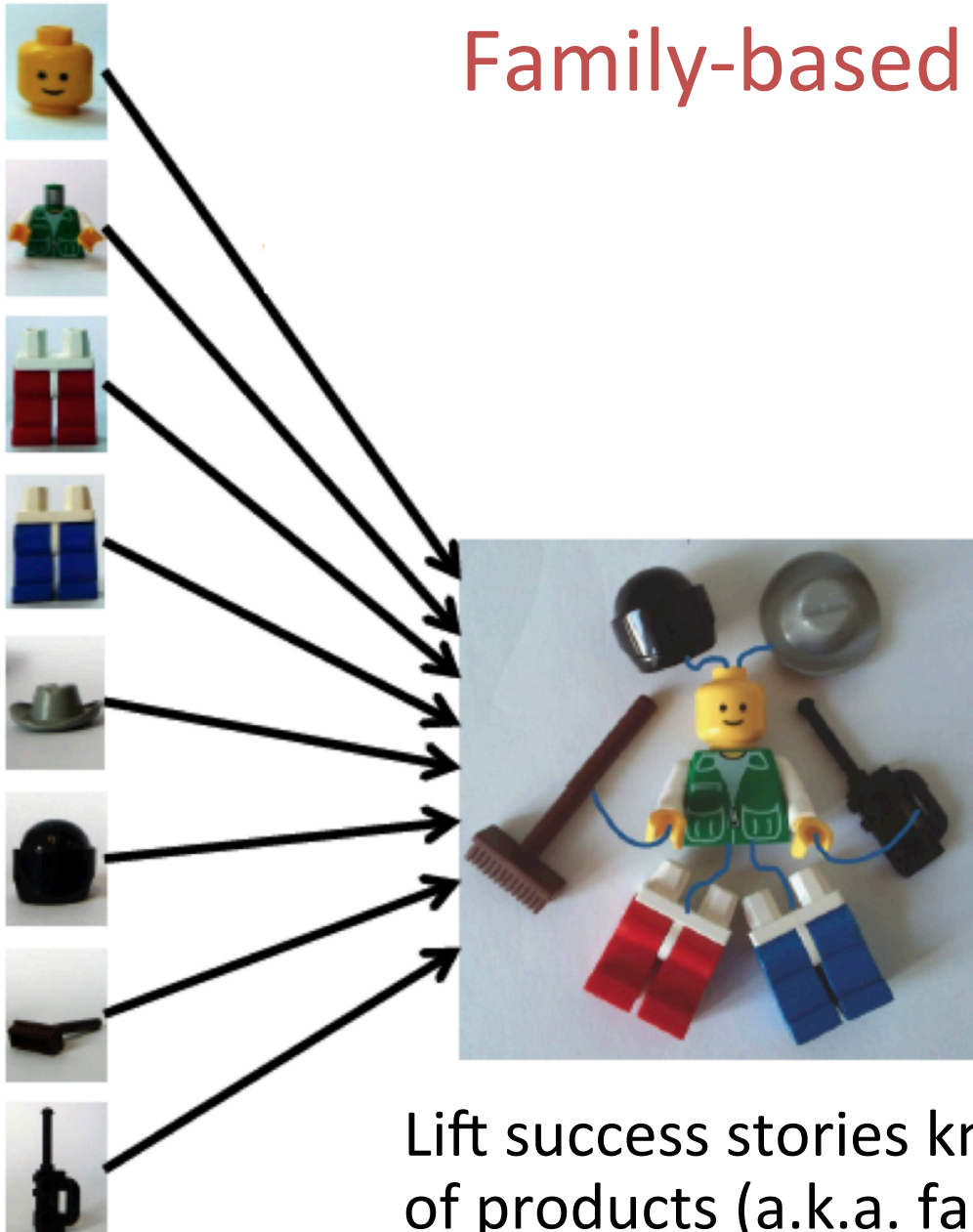
Beneficial for many products with substantial similarities

Generates complex analysis tasks

Requires (compact) family metamodels



Family-based Analysis



Lift success stories known for single products to sets of products (a.k.a. families) - by exploiting variability
⇒ challenges known models and tools by potentially high number of different products, each with large state space

Dedicated Variability Model Checker for MTSu

VMC offers both product-based and family-based analyses

VMC v6.1

● ● ● ● ●

- Edit Model
- View Current Model
- Explore the MTS
- Draw Family MTS
- Generate Products
- Welcome
- Quit



Kandisky 1908

The Formula:

```
[behaviour] not E[true {not tea} U {serveTea}true]
```

is TRUE

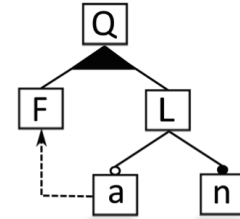
The formula holds for ALL the MTS products

(states generated= 11, computations fragments generated= 10, evaluation time= 0.000644000 sec.)

ACT-UCTL-SoCL-VACTL

```
[behaviour] not E [true {not tea} U {serveTea} true]
```

Quantitative Modelling and Analysis of Highly (re)Configurable Systems



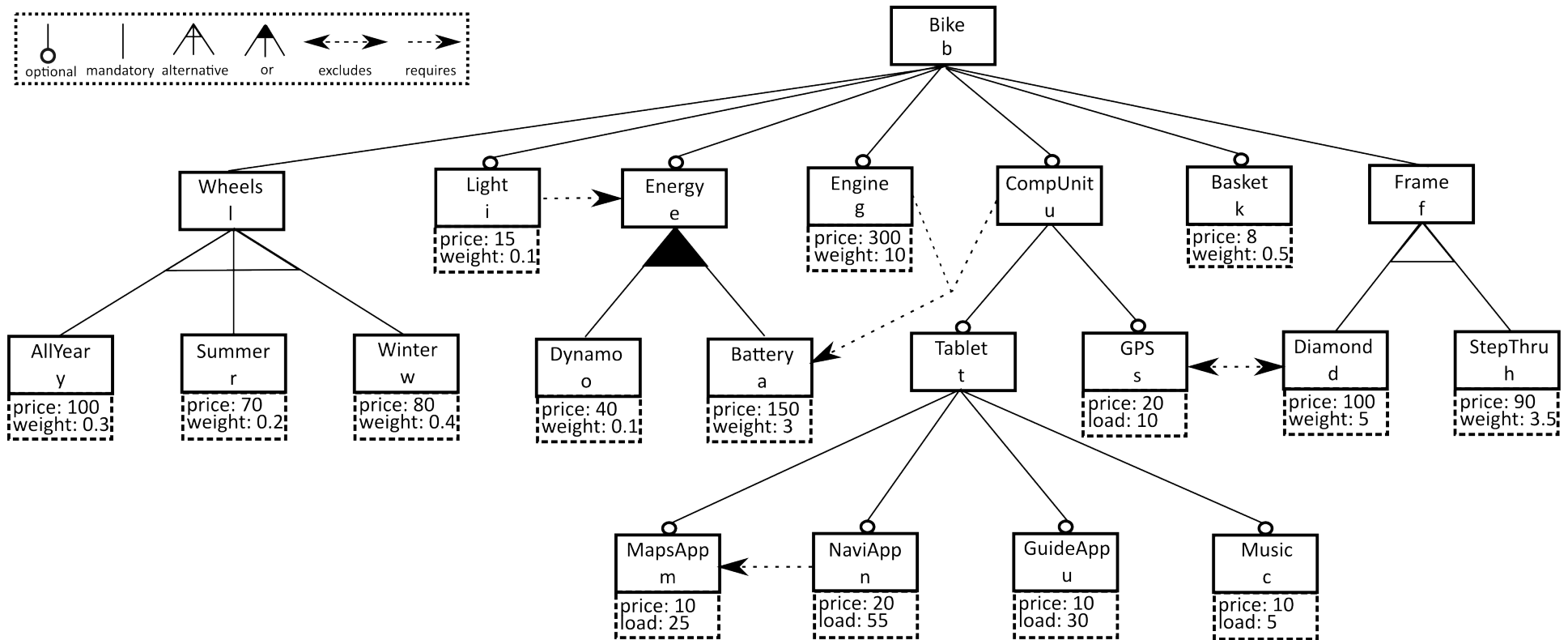
The screenshot displays the QFLan IDE interface with several components:

- Project Explorer:** Shows a project structure for 'fase2018' containing 'MultiVeStA_OUTPUT', 'src-gen', 'VendingMachine.qflan', and other related files.
- QFLan Editor:** Contains the model definition for 'VendingMachine', including variables, abstract features (Machine Beverage CoffeeBased), concrete features (Cocoa Tea Cappuccino Coffee), and a feature diagram with constraints like `Machine -> {?Cocoa, Beverage}` and `Beverage -XOR-> {CoffeeBased, Tea}`.
- Outline View:** Provides a hierarchical view of the model's structure, listing variables, abstract features, concrete features, feature relations, constraints, and actions.
- Console View:** Shows the execution output of the MultiVeStA client, including iteration counts, simulation runs, and convergence status.
- Plot View:** Displays a graph titled 'MultiVeStA analysis of VendingMachine.qflan' showing 'Means estimations' over time (x-axis) for five different observation steps (obs1AtStep(x) to obs5AtStep(x)). The plot shows convergence of means over time.

<https://github.com/qflanTeam/QFLan/>

TSE,
FM,
SPLC

A Smart Bike Product Line's Structural Constraints



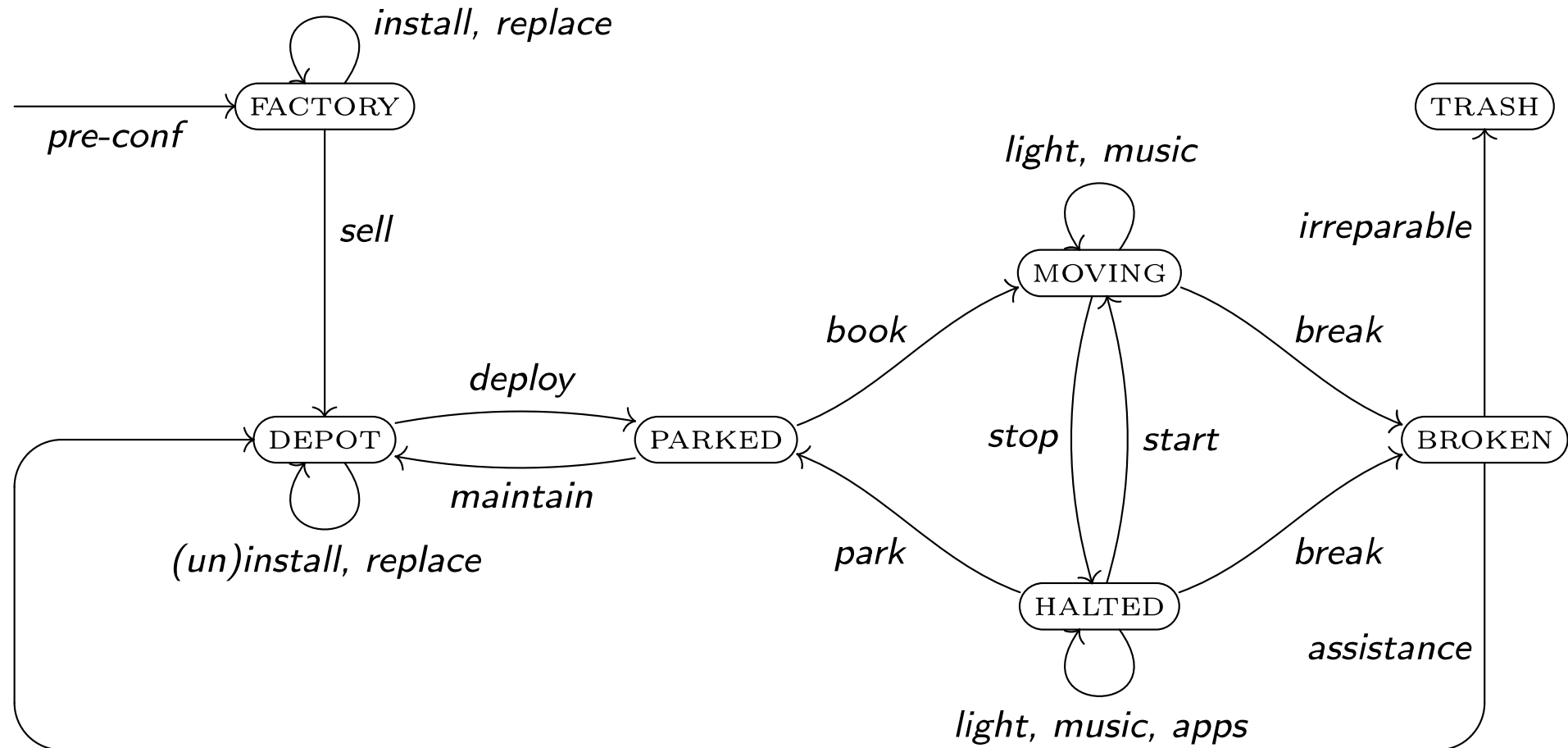
Additional feature constraints:

$\sum_{f \in \mathcal{P}_F} price(f) \leq 600$: a bike may cost at most 600 €

$\sum_{f \in \mathcal{P}_F} weight(f) \leq 15$: a bike may weigh up to 15 kg

$\sum_{f \in \mathcal{P}_F} load(f) \leq 100\%$: a bike's total computational load may not exceed 100%

A Smart Bike Product Line's Behavioural Constraints



Additional action constraints:

$$do(sell) \rightarrow \sum_{f \in \mathcal{P}_F} price(f) \geq 250$$

$$do(irreparable) \rightarrow \sum_{f \in \mathcal{P}_F} price(f) \leq 400$$

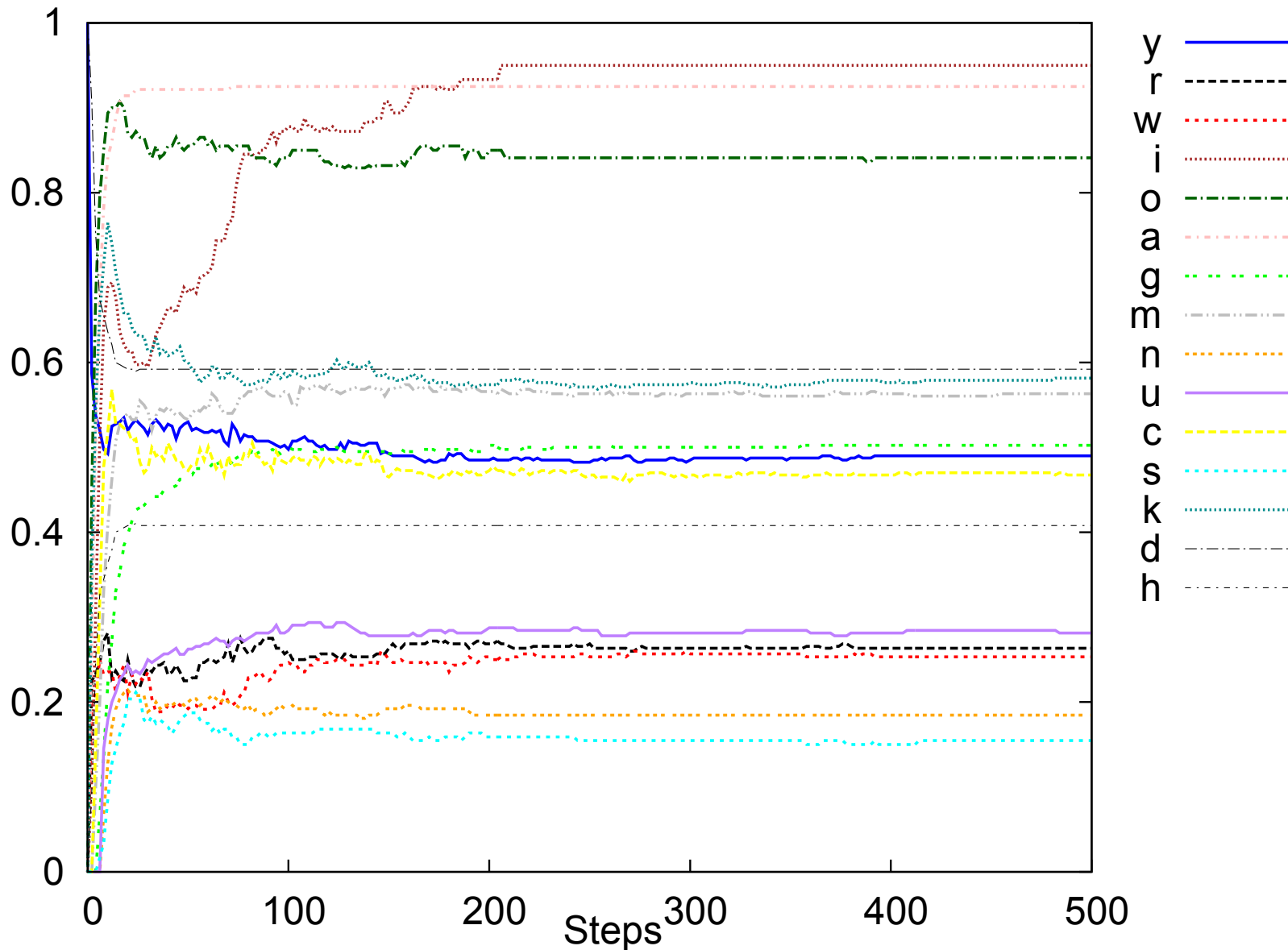
Statistical Model Checking with QFLan

Properties (over time):

- ① Average price, weight or load of a bike
- ② For each feature, the probability to be installed
- ③ The probability for a bike to be disposed of (irreparable)

Statistical Model Checking with QFLan

② For each feature, the probability to be installed



Family-based Model Checking with Off-the-Shelf Model Checkers

Dedicated model checkers must be maintained/optimised...

Family-based Model Checking with Off-the-Shelf Model Checkers

Dedicated model checkers must be maintained/optimised...

💡 Family-based model checking with  [Groote et al.]
analysing system behaviour

Φ	property	result	one-by-one	all-in-one
φ_1	<i>Absence of deadlock</i>	128/0	10.02	2.07
φ_3	<i>The controller cannot fairly receive each of the three message types</i>	0/128	24.33	0.25
φ_5	<i>The system cannot be in a situation in which the pump runs indefinitely in the presence of methane</i>	96/32	17.26	0.86
φ_6	<i>Assuming fairness (φ_3), the system cannot be in a situation in which the pump runs indefinitely in the presence of methane (φ_5)</i>	112/16	27.32	3.67
φ_7	<i>The controller can always eventually receive/read a message, i.e. it can return to its initial state from any state</i> (CTL\LTL formula)	128/0	18.36	2.40
φ_9	<i>Invariantly, when the level of methane rises, it inevitably decreases</i>	0/128	20.47	0.21
φ_{11}	<i>Products with feature Ct can always switch on the pump</i>	28/100	21.11	2.32
φ_{12}	<i>Products with features Ct, Ma and Lh can start the pump upon a high water level, but products without feature Lh cannot</i> (multi-feature formula)	128/0	13.35	3.36

Runtime improvement w.r.t. product-based model checking:
 average speed up of ± 31 - ranging from ± 4 (φ_{12}) to >97 (φ_3)

Formal Methods and Tools: Experiences



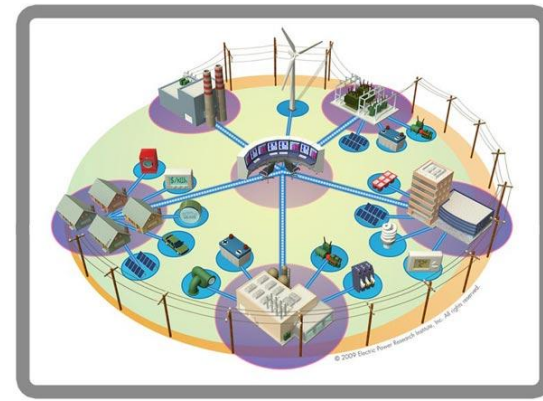
CAS for a smart society: smart urban transport, smart grids



Objective: support fair and efficient resource management in systems of heterogeneous components with competing goals



CAS for a smart society: smart urban transport, smart grids



Objective: support fair and efficient resource management in systems of heterogeneous components with competing goals

With bike-sharing systems as case study, FMT focussed on:

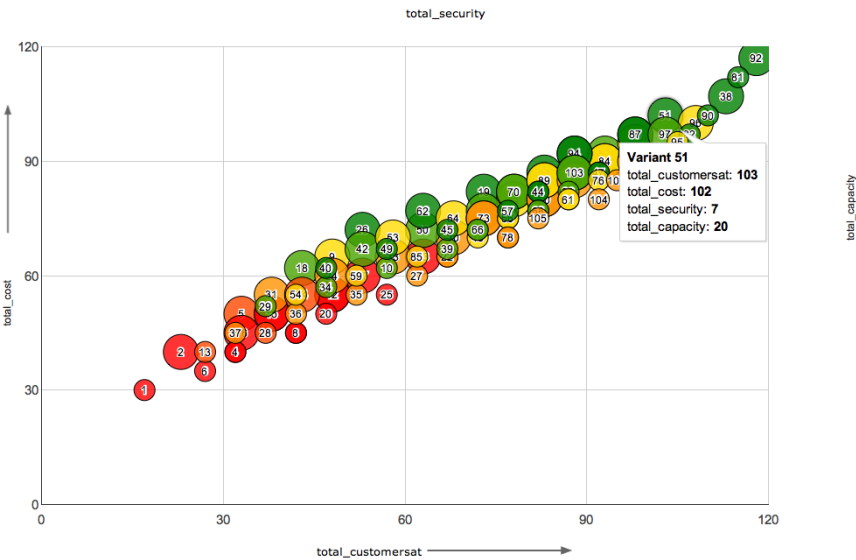
- Scalable formal verification approaches for CAS
- Quantitative business models and product lines

Variability Analysis of Pisa's Bike-sharing System

Aim: systematic evaluation of options for improvement (i.e. costs/benefits) *before* actually implementing them

Variability Analysis of Pisa's Bike-sharing System

Aim: systematic evaluation of options for improvement (i.e. costs/benefits) *before* actually implementing them

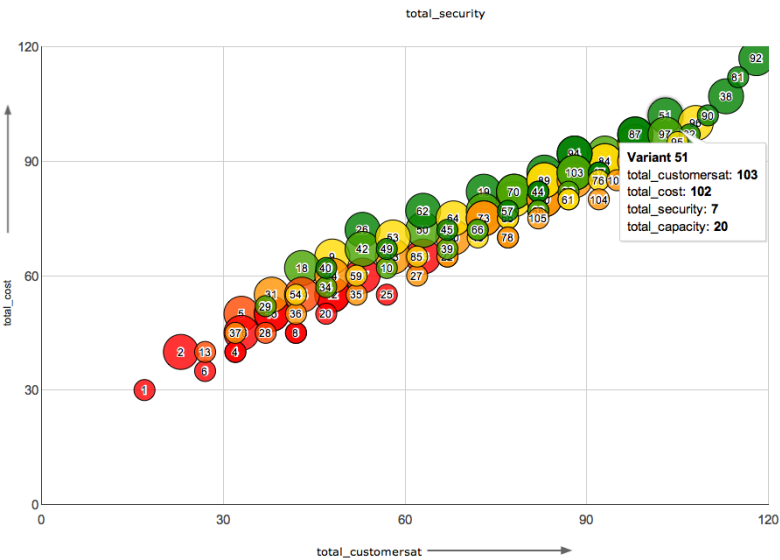


MOO attributed feature models with Clafer [Czarnecki et al.]:

- Compare system configurations w.r.t. various quality dimensions, select the most desirable variant (possibly resolving trade-offs) and understand the impact of re-configurations on a variant's quality dimensions

Variability Analysis of Pisa's Bike-sharing System

Aim: systematic evaluation of options for improvement (i.e. costs/benefits) *before* actually implementing them

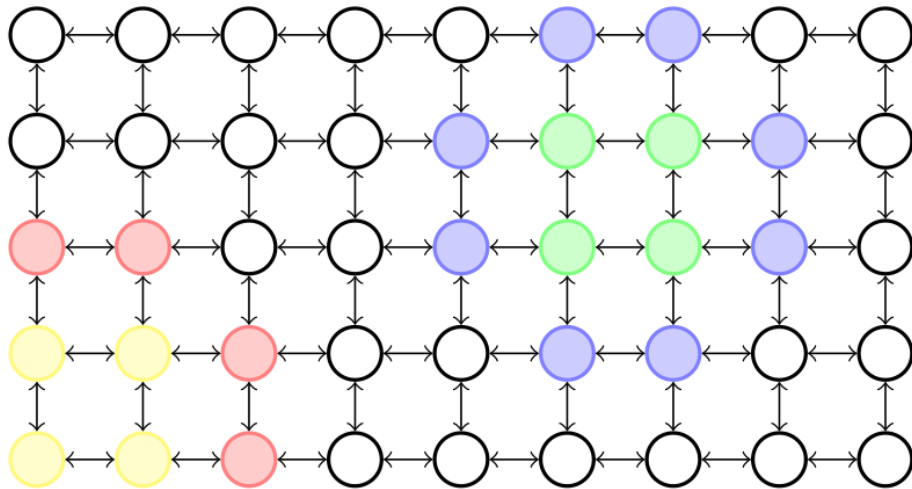


Mean field model checking with FlyFast:

- If we augment the capacity of some stations (e.g. in centre), then how does this effect the occupation level of all stations?
- If we add a station between the centre and a little used peripheral station, does the latter's usage frequency increase?

Statistical Spatio-temporal Model Checking

Reachability properties in graphs (discretised physical space)



$\Phi ::= p$	[ATOMIC PROPOSITION]
\top	[TRUE]
$\neg \Phi$	[NOT]
$\Phi \wedge \Phi$	[AND]
$\mathcal{N}\Phi$	[NEAR]
$\Phi \mathcal{S} \Phi$	[SURROUNDED]

Derived operators, like interior: $\mathcal{I}\Phi = \neg \mathcal{N} \neg \Phi$

All red and yellow points satisfy: $\mathcal{N} \text{yellow}$

One yellow point satisfies: $\mathcal{I} \text{yellow}$

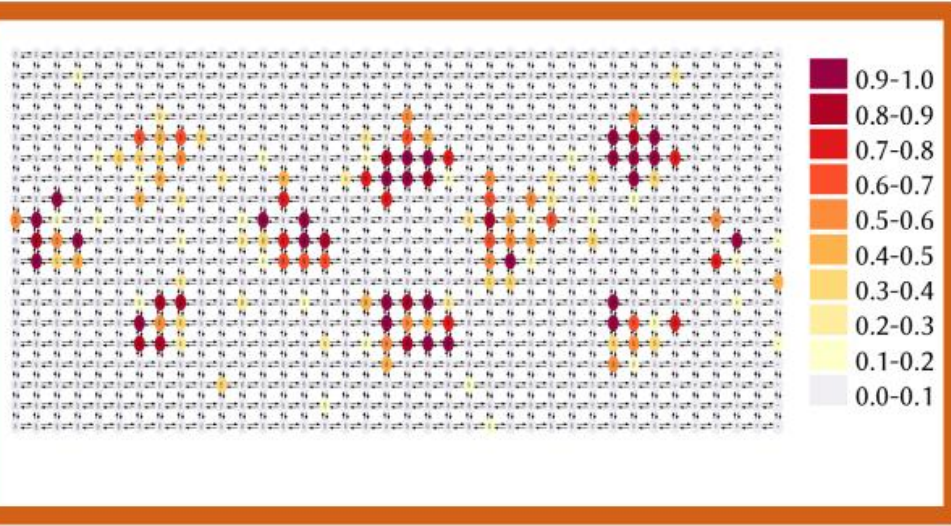
No points satisfy: $\mathcal{I} \text{green}$

Green points satisfy: $\text{green} \mathcal{S} \text{blue}$



Statistical Spatio-temporal Model Checking

Reachability properties in graphs (discretised physical space)



Φ	::=	p	[ATOMIC PROPOSITION]
		\top	[TRUE]
		$\neg\Phi$	[NOT]
		$\Phi \wedge \Phi$	[AND]
		$\mathcal{N}\Phi$	[NEAR]
		$\Phi \mathcal{S} \Phi$	[SURROUNDED]

Derived operators, like interior: $\mathcal{I}\Phi = \neg\mathcal{N}\neg\Phi$

Spot congestion in bike-sharing system:

full = [vacantPlaces == 0]

cluster = I full

eventuallyCluster = EF cluster



H2020 Shift2Rail Initiative: 920M€ (2014–20)

“Formal methods are fundamental for safe and reliable technological advances to increase the competitiveness of the European rail industry”

H2020 Shift2Rail Initiative: 920M€ (2014–20)

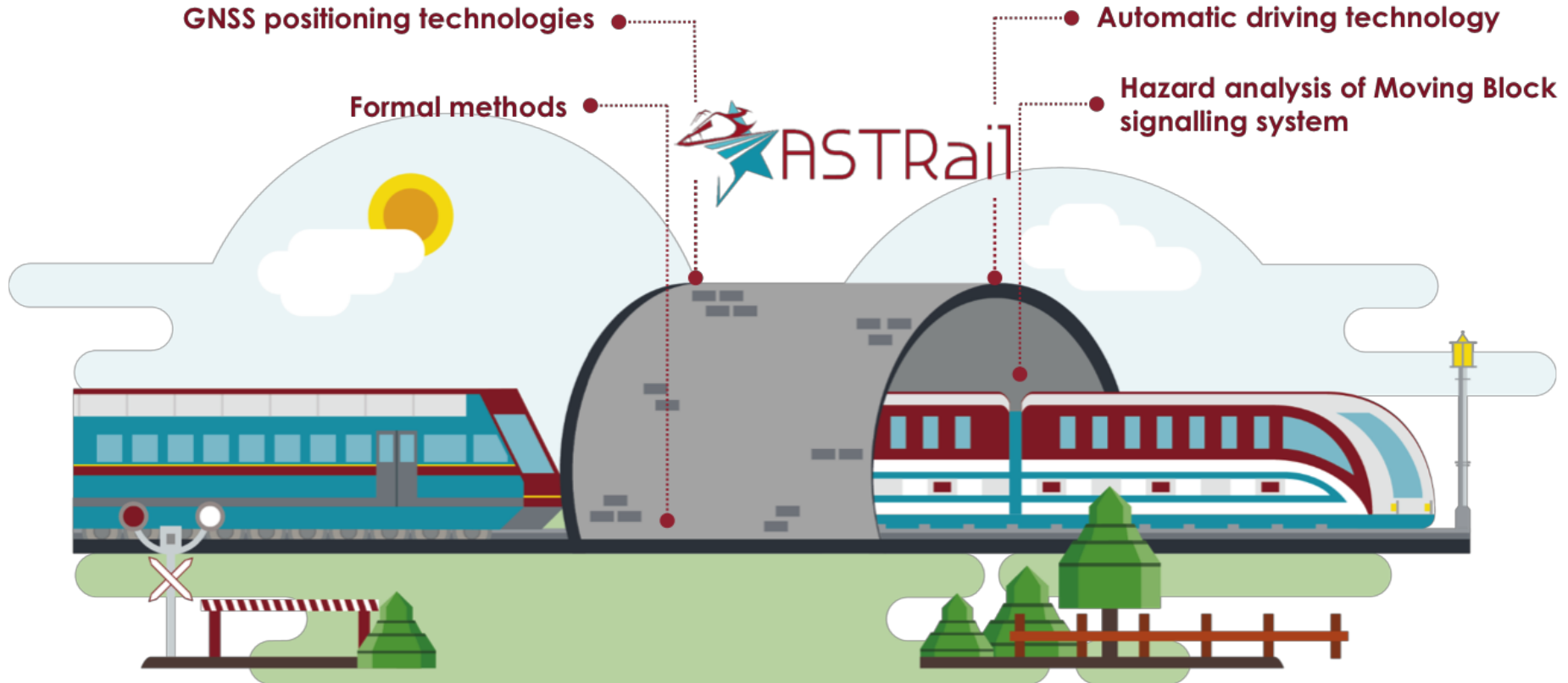
“Formal methods are fundamental for safe and reliable technological advances to increase the competitiveness of the European rail industry”

GNSS positioning technologies ●

● Automatic driving technology

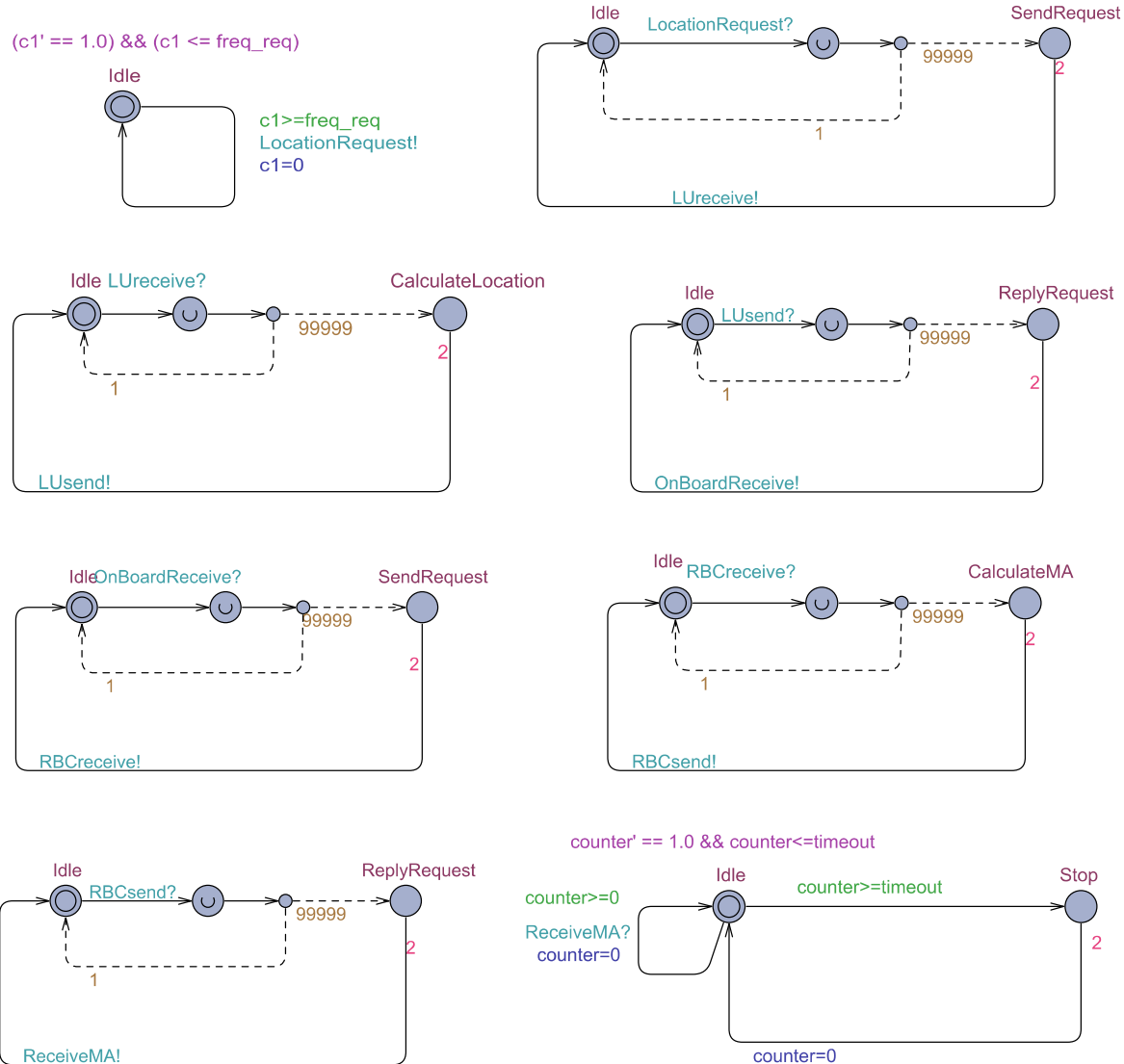
Formal methods ●

● Hazard analysis of Moving Block signalling system

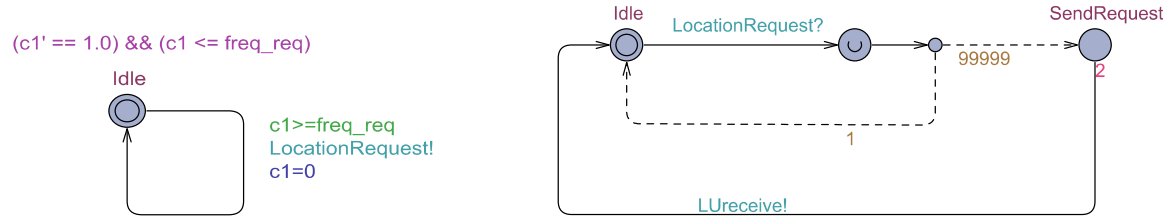


<http://www.astrail.eu/>

Statistical Model Checking of a Moving Block Railway Signalling Scenario with Uppaal SMC



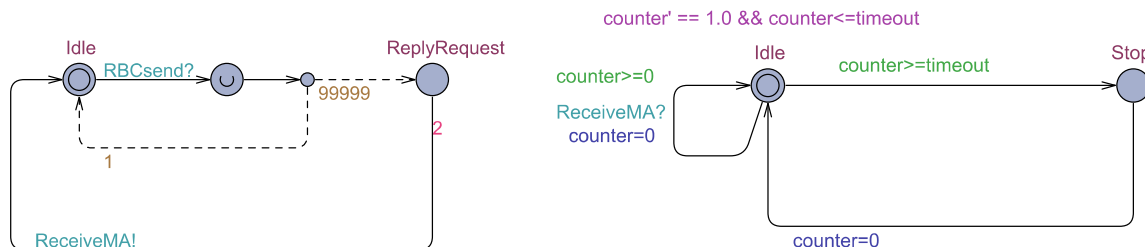
Statistical Model Checking of a Moving Block Railway Signalling Scenario with Uppaal SMC



Probability that train enters safe state Stop upon timeout:

$$\mathbb{P}_M(\diamond_{\leq(\text{timeout})} \text{Controlling.Stop})$$

Uppaal SMC [Larsen et al.] reports that this probability is in the interval $[0, 9.99994e-005]$, with confidence 0.995, obtained from 59912 runs in ± 5 minutes (M is the model)





Unione Europea
Fondo Europeo di Sviluppo Regionale

Regione Toscana

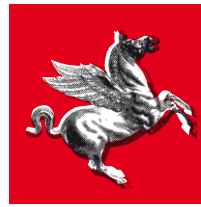


Renew the role of railway stations in the future's smart cities



Unione Europea
Fondo Europeo di Sviluppo Regionale

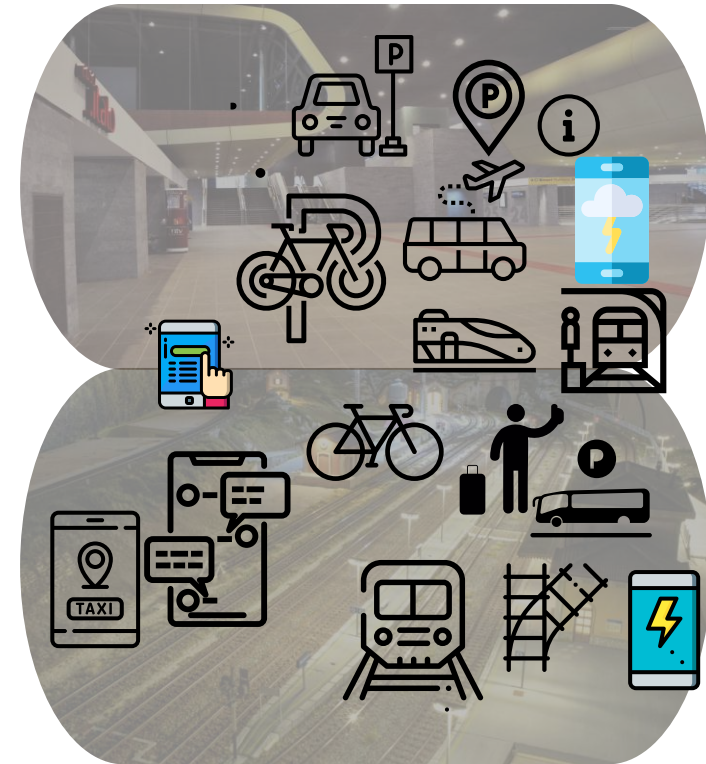
Regione Toscana



Renew the role of railway stations in the future's smart cities

Revisit station communication infrastructure, integrating power line and wireless technologies in order to:

- Realize LAN network over station plants
- Implement remote monitoring and control of the station equipment
- Create value-added services for both customers and staff, such as connectivity, energy management, environmental surveying, video surveillance, fault prediction, and infomobility



FM'19

3rd world
congress
on formal
methods

Thanks for your attention!

FM week

FM 2019

23rd Symposium on Formal Methods

LOPSTR 2019

29th International Symposium on Logic-Based Program
Synthesis and Transformation

MPC 2019

13th International Conference on Mathematics of Program Construction

PPDP 2019

21st International Symposium on Principles and Practice
of Declarative Programming

RV 2019

19th International Conference on Runtime Verification

SAS 2019

26th International Static Analysis Symposium

TAP 2019

13th International Conference on Tests and Proofs

UTP 2019

7th International Symposium on Unifying Theories of Programming

VECoS 2019

13th International Conference on Verification and Evaluation
of Computer and Communication Systems

Doctoral Symposium

Industry Day

**and several Workshops
and Tutorials!**

7 — 11
october
2019

Alfandega Porto
Congress Centre
Porto, Portugal

