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

Consiglio Nazionale delle Ricerche

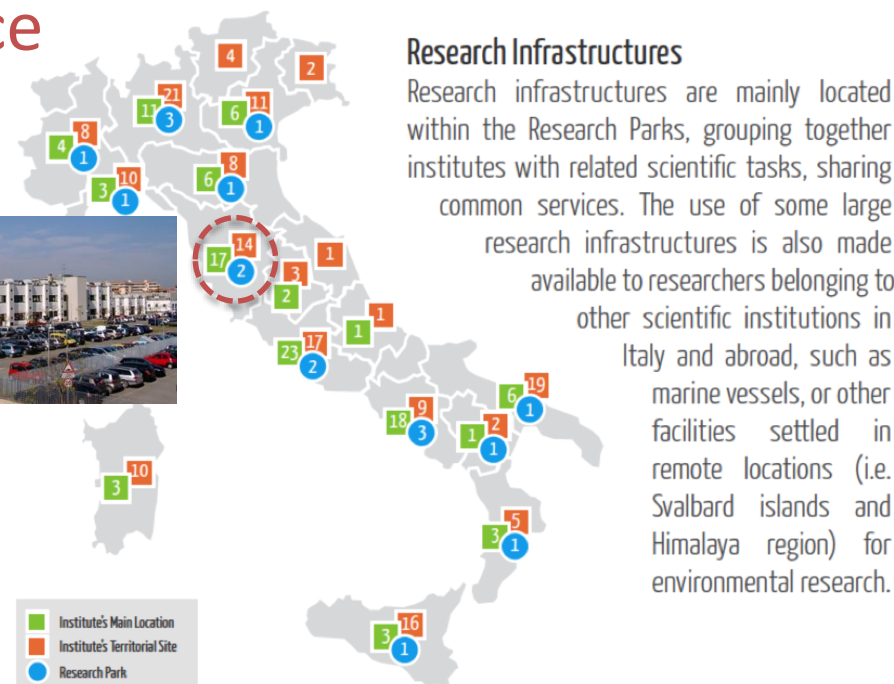
Italy's National Research Council (CNR)

- Est. 1923, full range of scientific areas, annual budget 1B€
- More than 9000 FTE (⅔ 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

Consiglio Nazionale delle Ricerche

Italy's National Research Council (CNR)

- Est. 1923, full range of scientific areas, annual budget 1B€
- More than 9000 FTE (⅔ 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 m^2
  >1500 persons (±1000 CNR)

Formal Methods and Tools (FMT)

Formal Methods and Tools (FMT)

Software Development Process

Requirements → Design → Code → Test

Formal Methods and Tools (FMT)

Software Development Process

Requirements → Design → Code → Test




- Model Checking
- Product Line Engineering
- Natural Language Processing

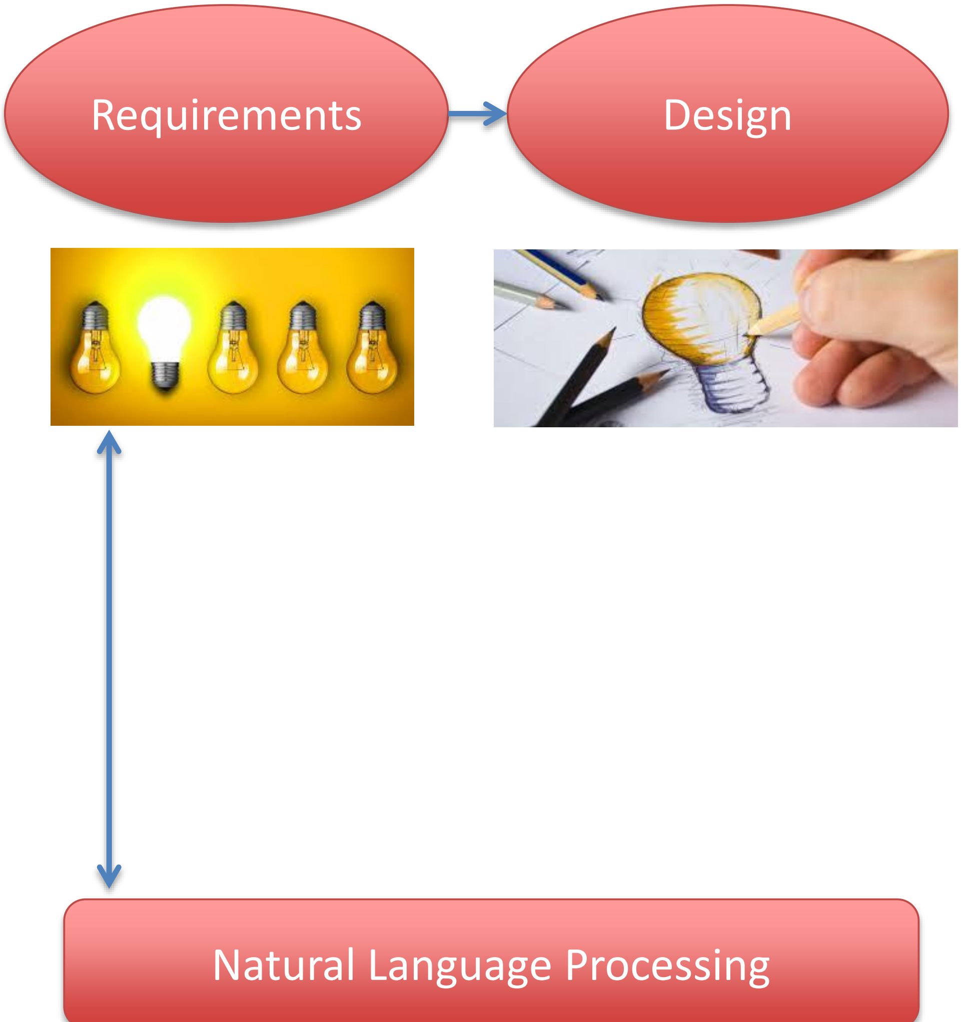



eGov

Techniques and Tools

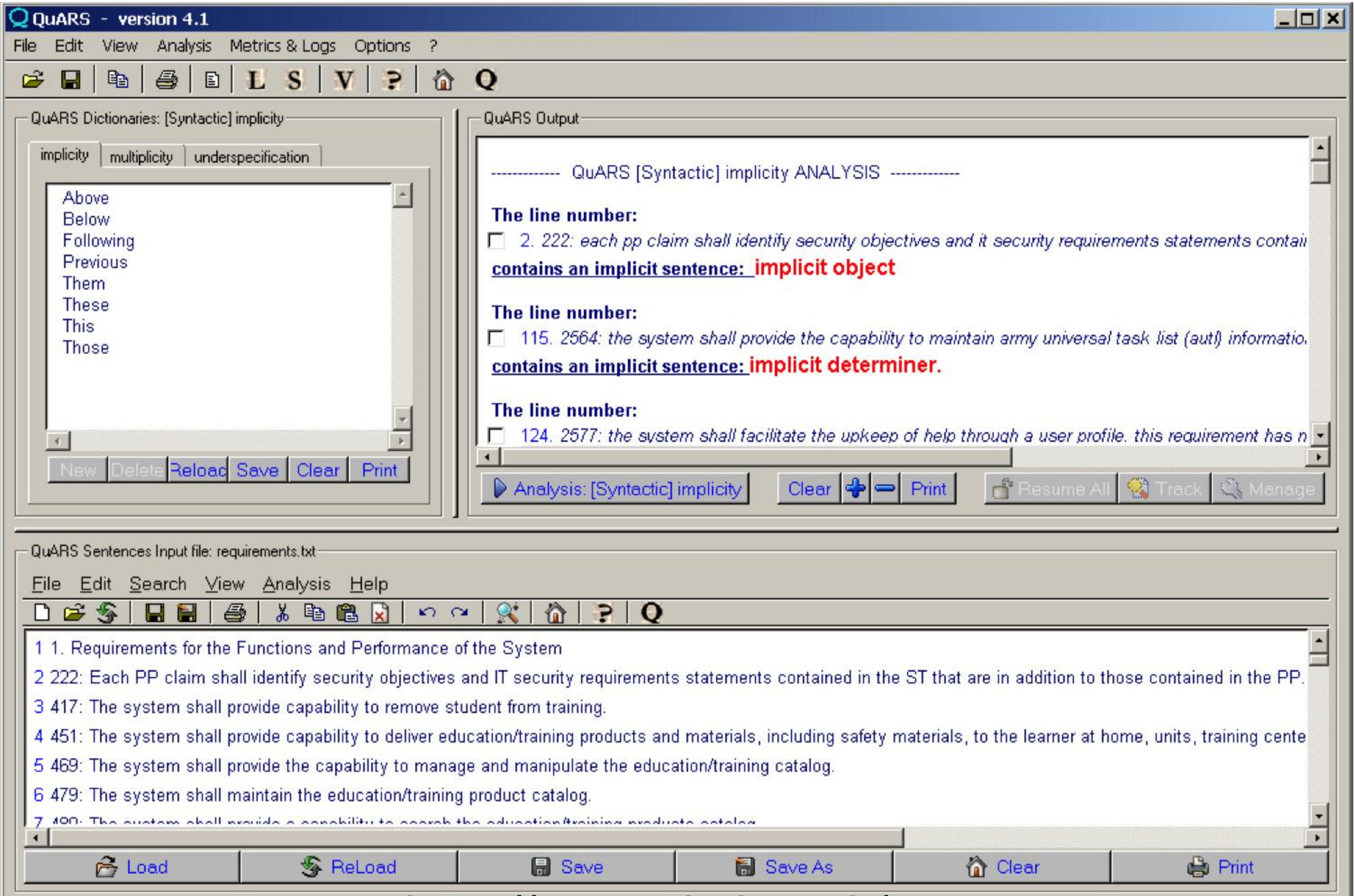
Application Domains

Formal Methods and Tools: Techniques



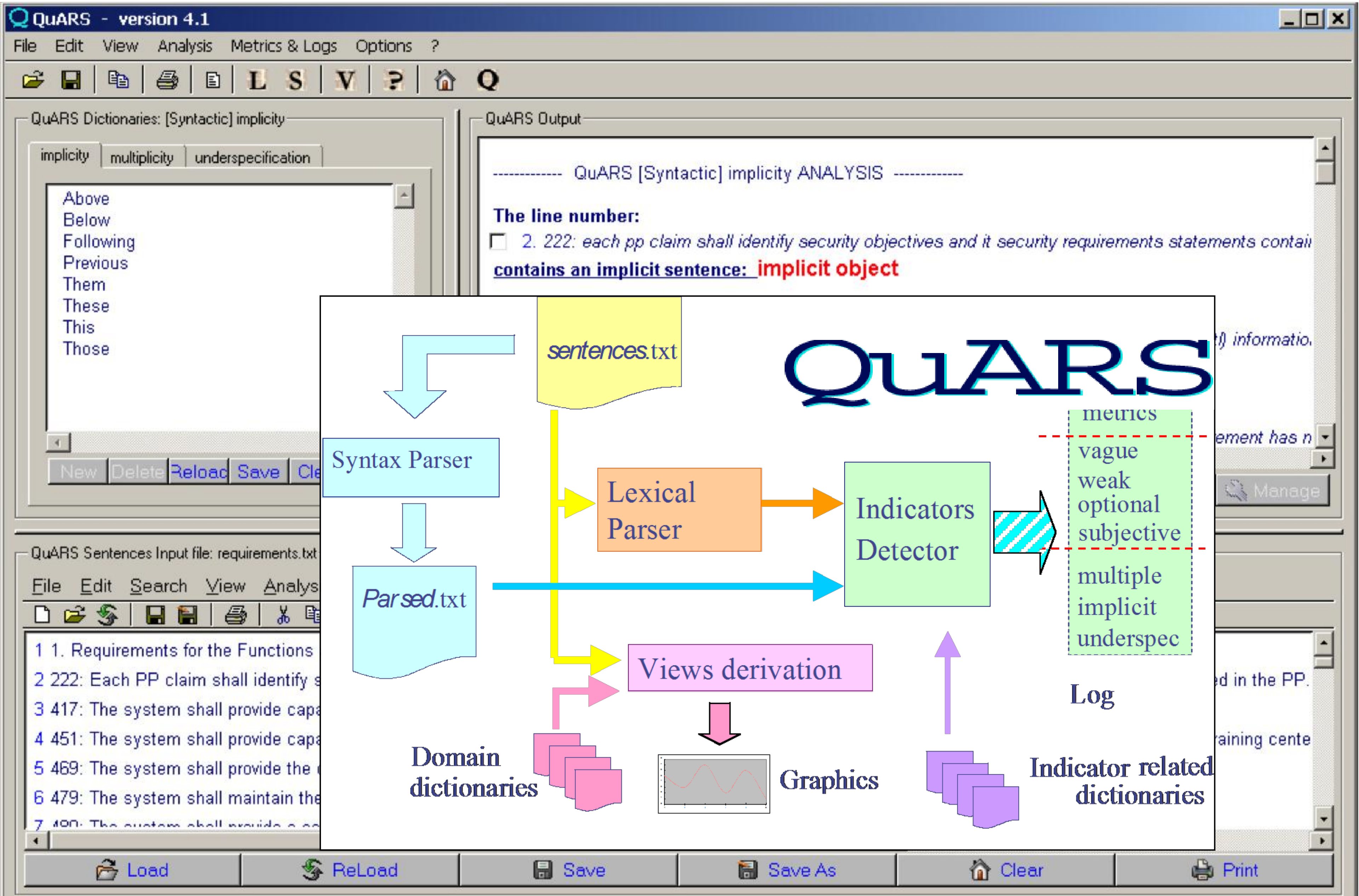
Techniques and Tools

Natural Language Processing



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

Natural Language Processing



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

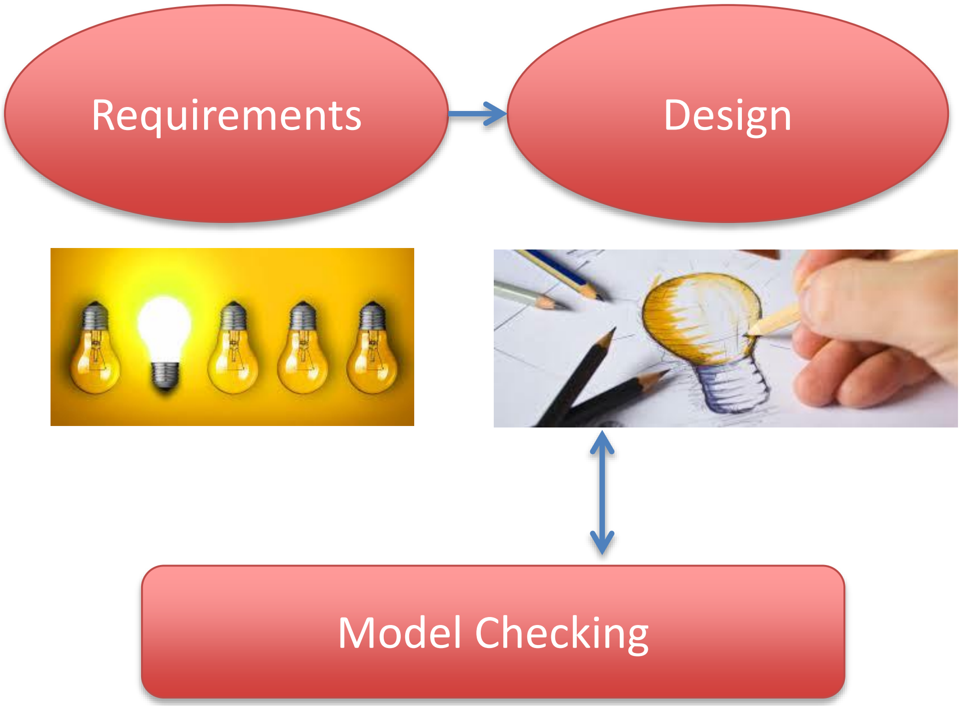
Defects	Optionality	Subjectivity	Vagueness	Weakness
1	--	--	possible	--

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 inte[illegible] (sensors, as fan speed, water/air temperatures, water flux, water conductivity, shall be present[illegible] 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

Requirements → Design

Product Line Engineering

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

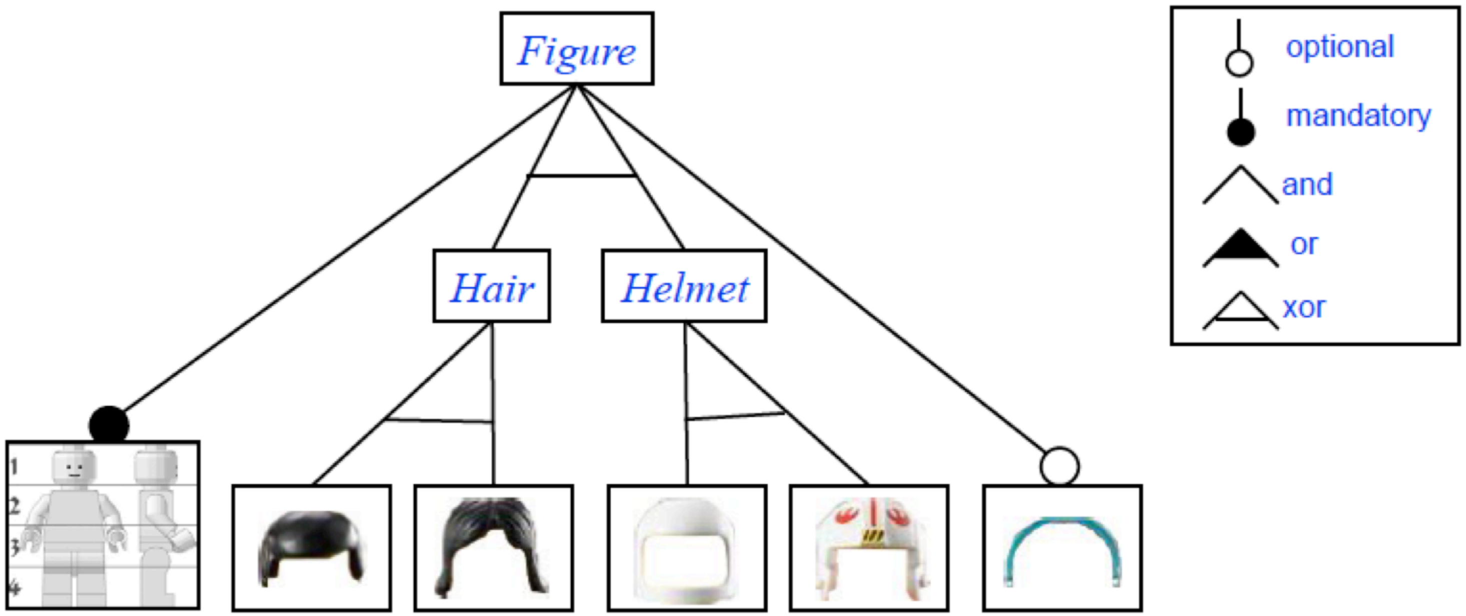
Variability in terms of features and constraints:

- 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!



Visor $\Rightarrow$ *Helmet*

Feature model: compact representation of all products

Lego example by S. Apel

Scalability

33 optional, independent features

a unique product for every

person on this planet

slide by C. Kästner

(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

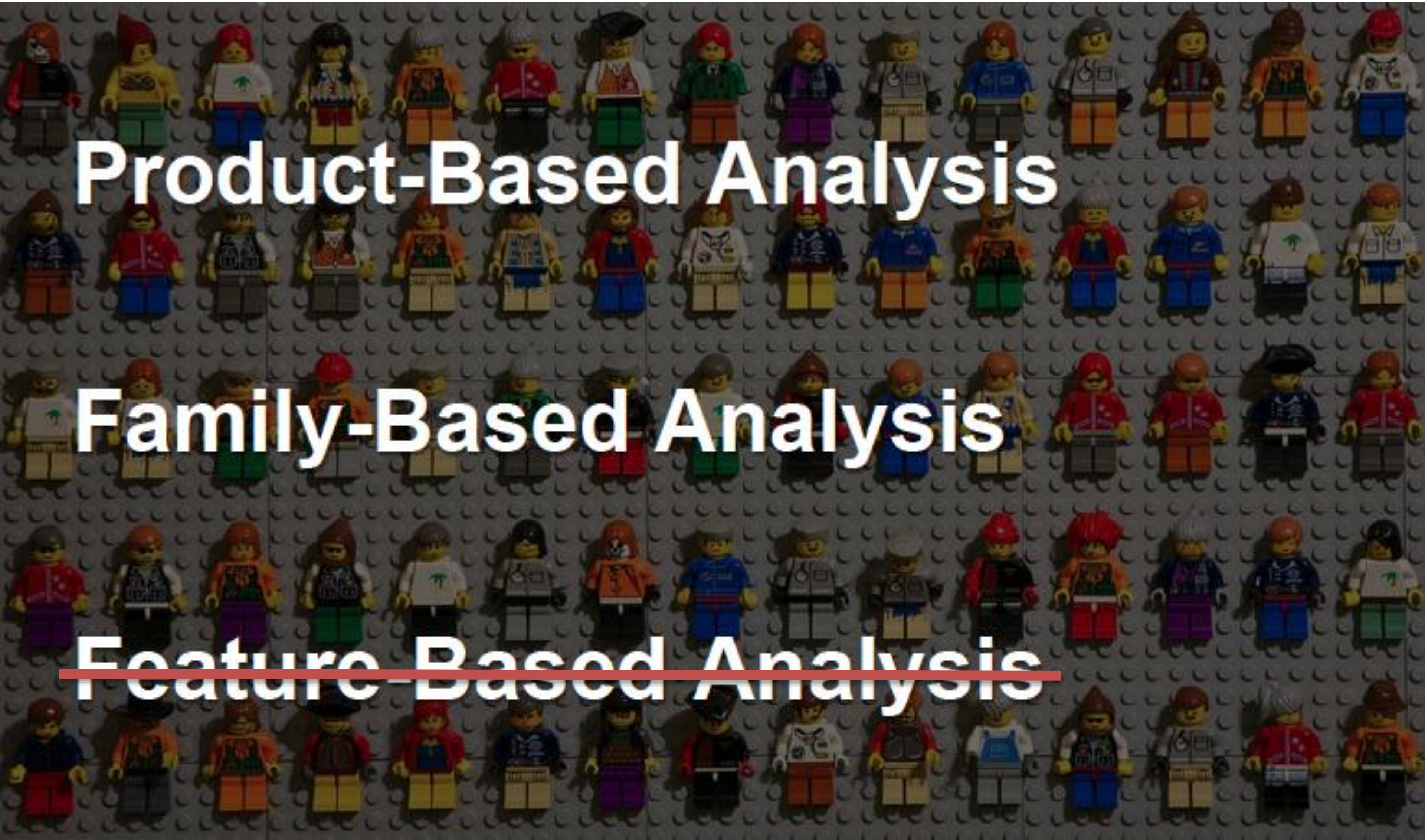
- For decades now successful in 'single' product engineering
- Not exploited broadly in SPLE, even though correctness of both artifacts for reuse and developed products is crucial!

Traditionally:

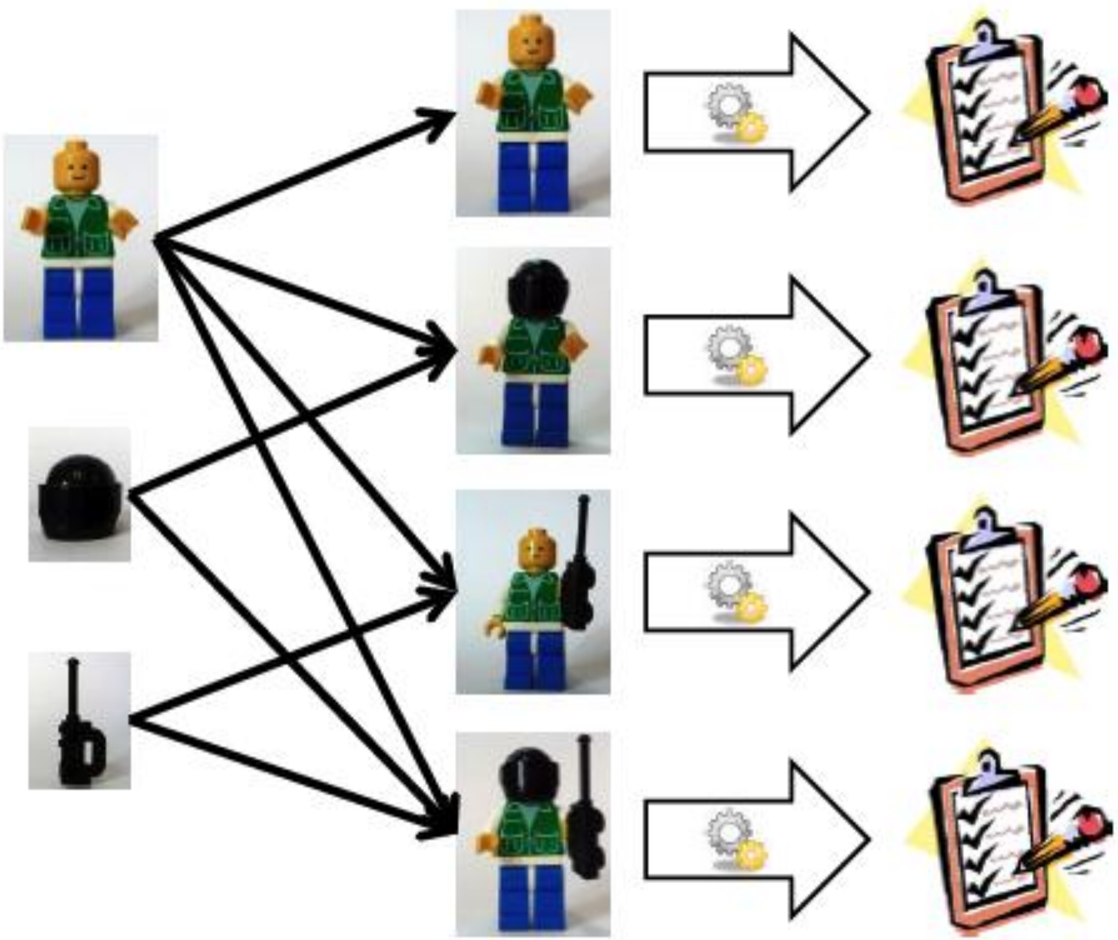
- Mainstream formal methods do *not* consider variability
- Formal methods that *have* been applied in SPLE typically focus on *structural* rather than on *behavioural* properties (feature model analysis, e.g. dead/false optional features)

Variability Analysis Strategies

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

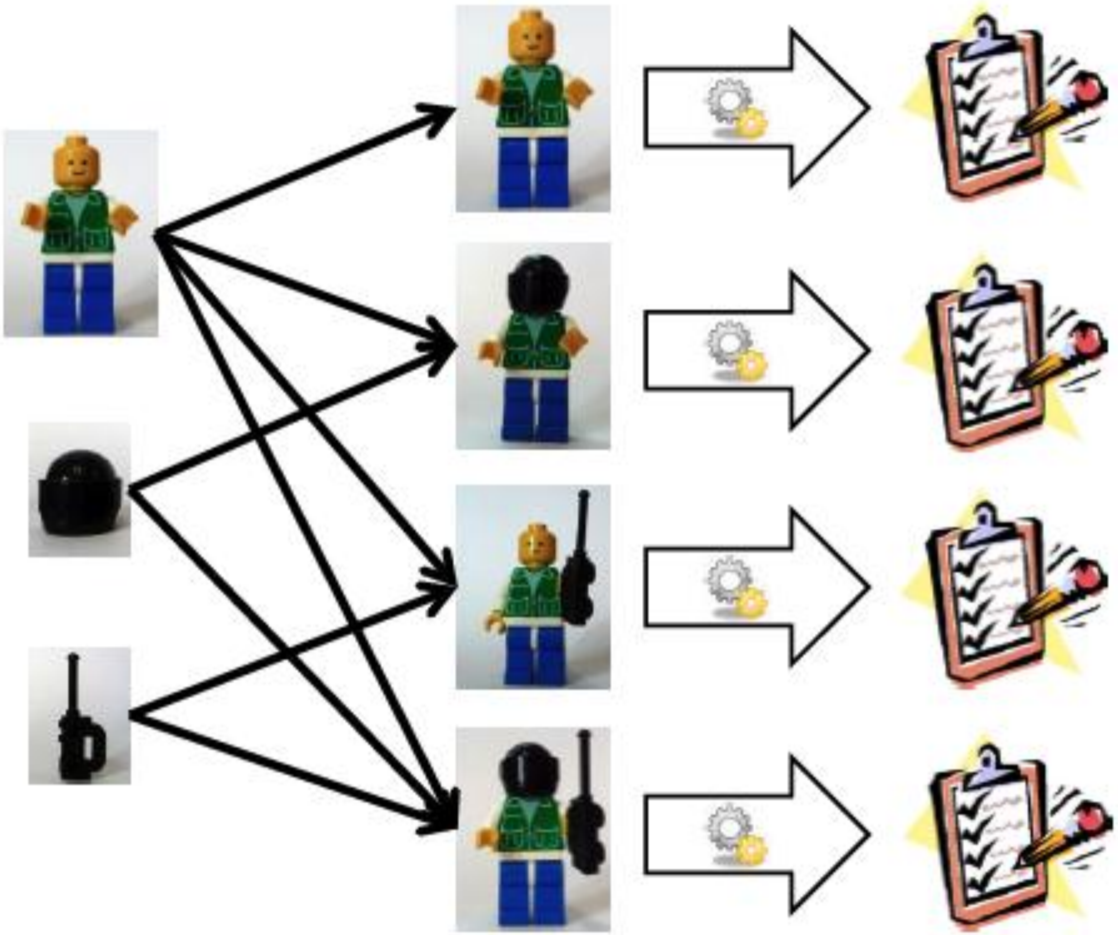


Product-based Analysis



$O(2^n)$ for n features

Product-based Analysis



Simple approach

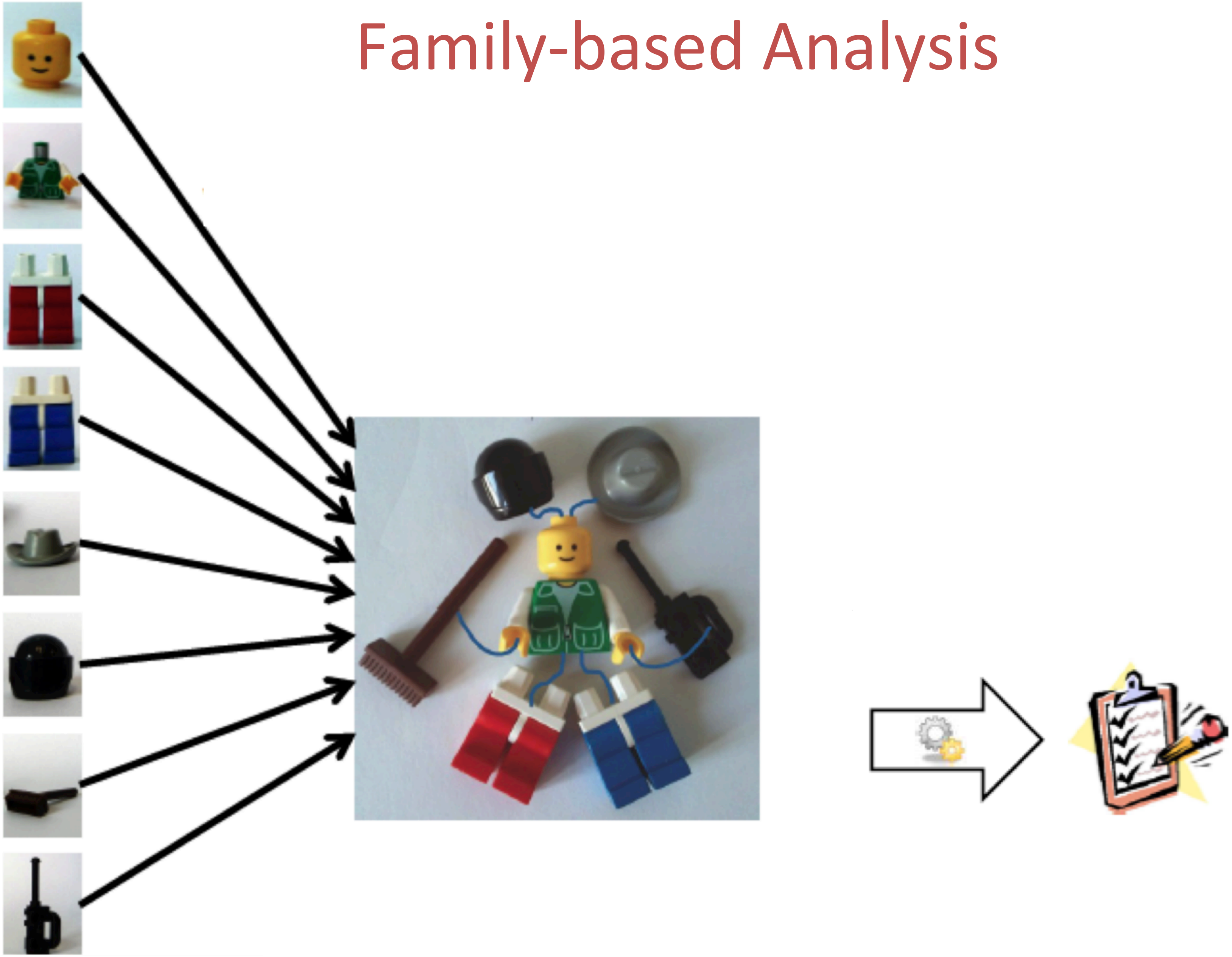
Standard tools available

Infeasible for large product sets

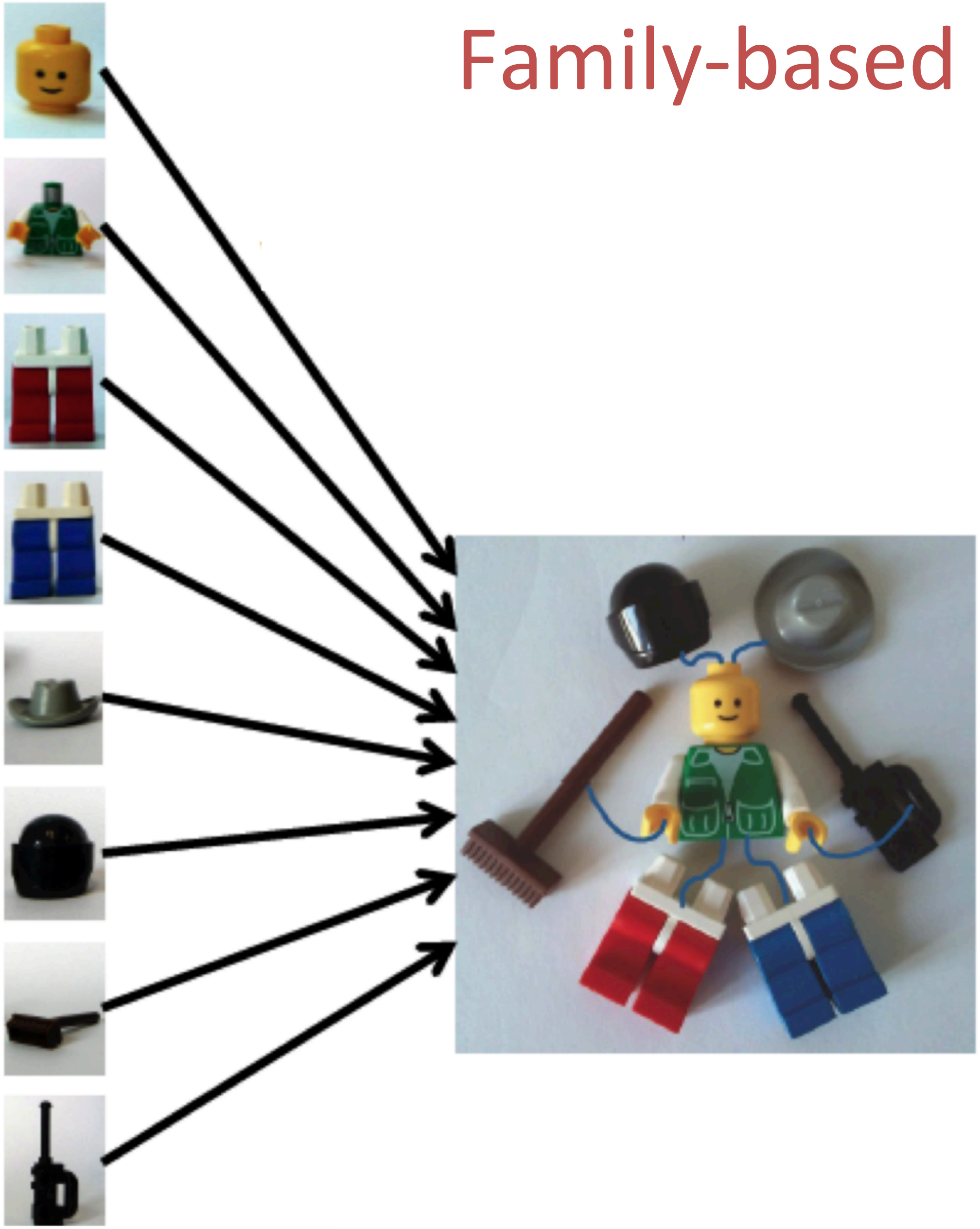
Same behaviour / code verified numerous times

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

Family-based Analysis



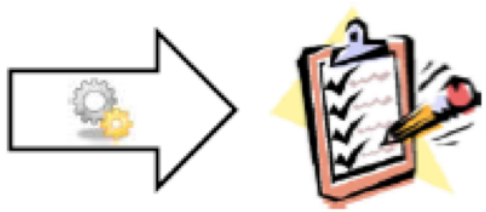
Family-based Analysis



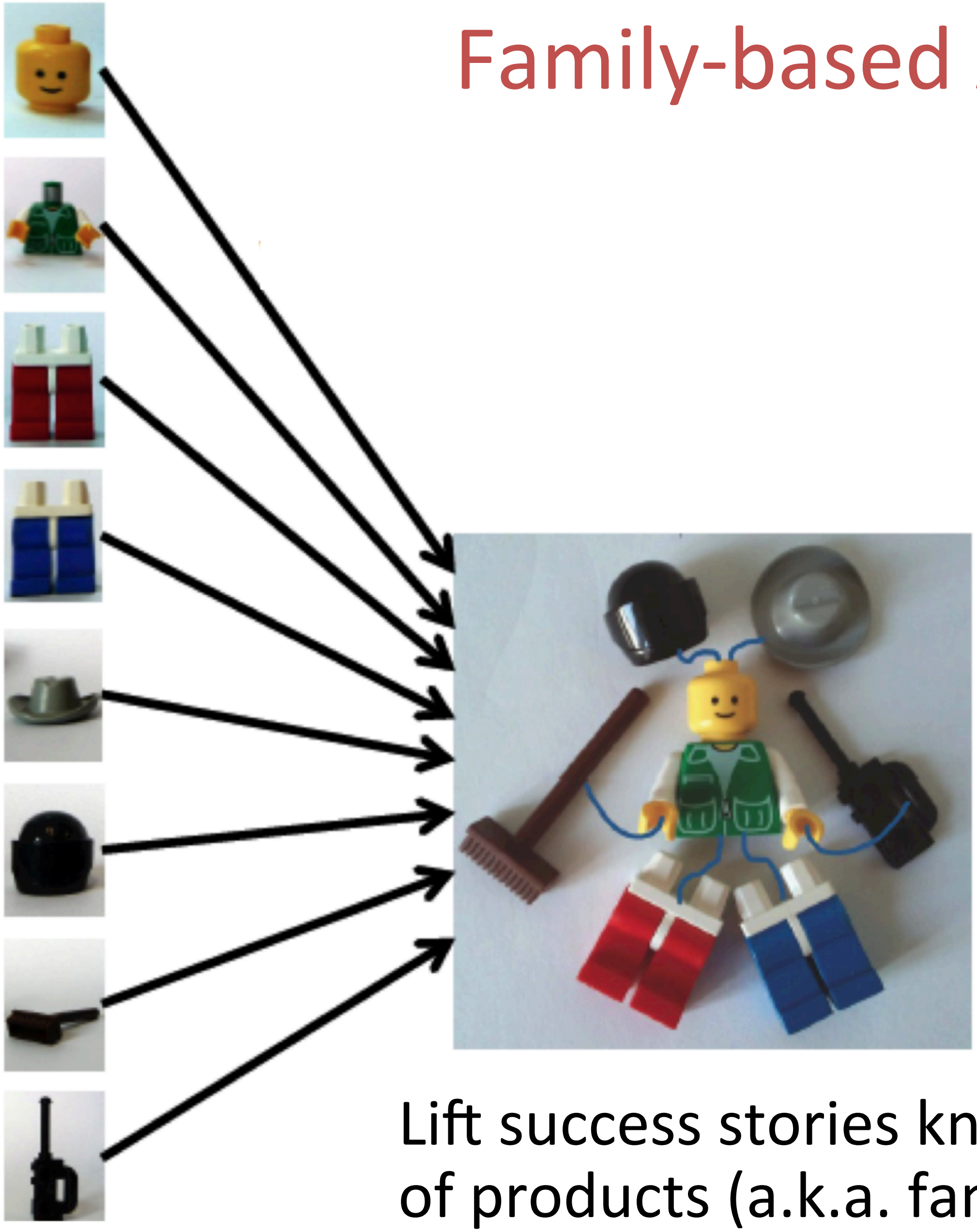
Beneficial for many products with substantial similarities

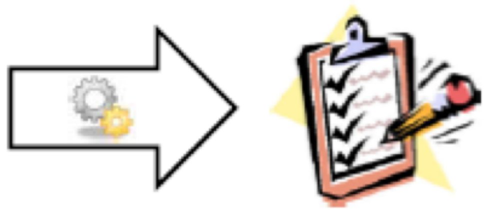
Generates complex analysis tasks

Requires (compact) family metamodels



Family-based Analysis

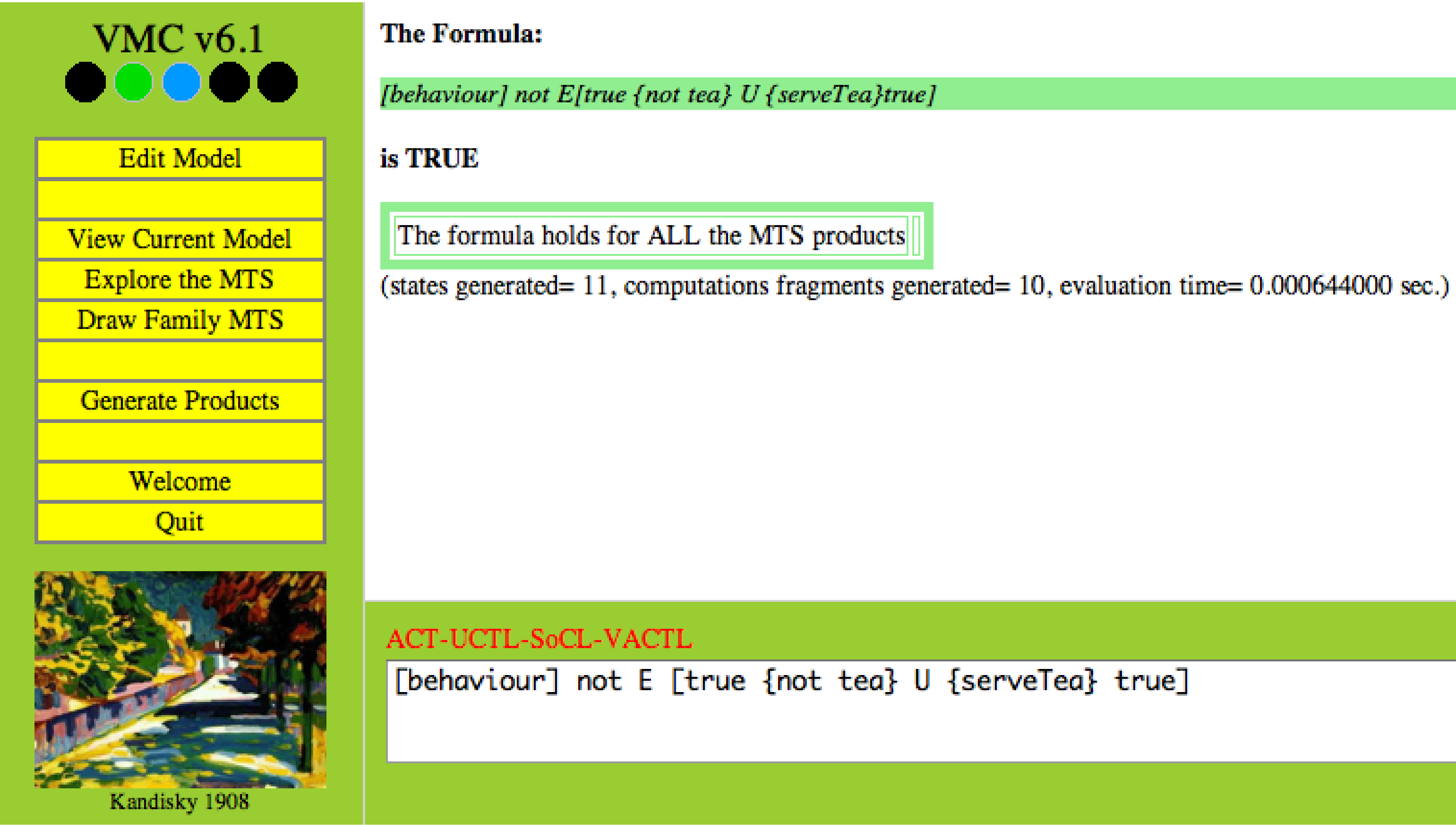




Lift success stories known for single products to sets of products (a.k.a. families) - by exploiting variability
$\Rightarrow$ challenges known models and tools by potentially high number of different products, each with large state space

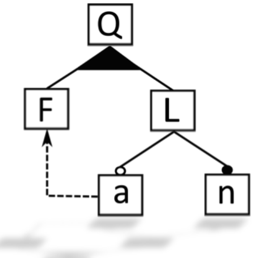
Dedicated Variability Model Checker for MTSu

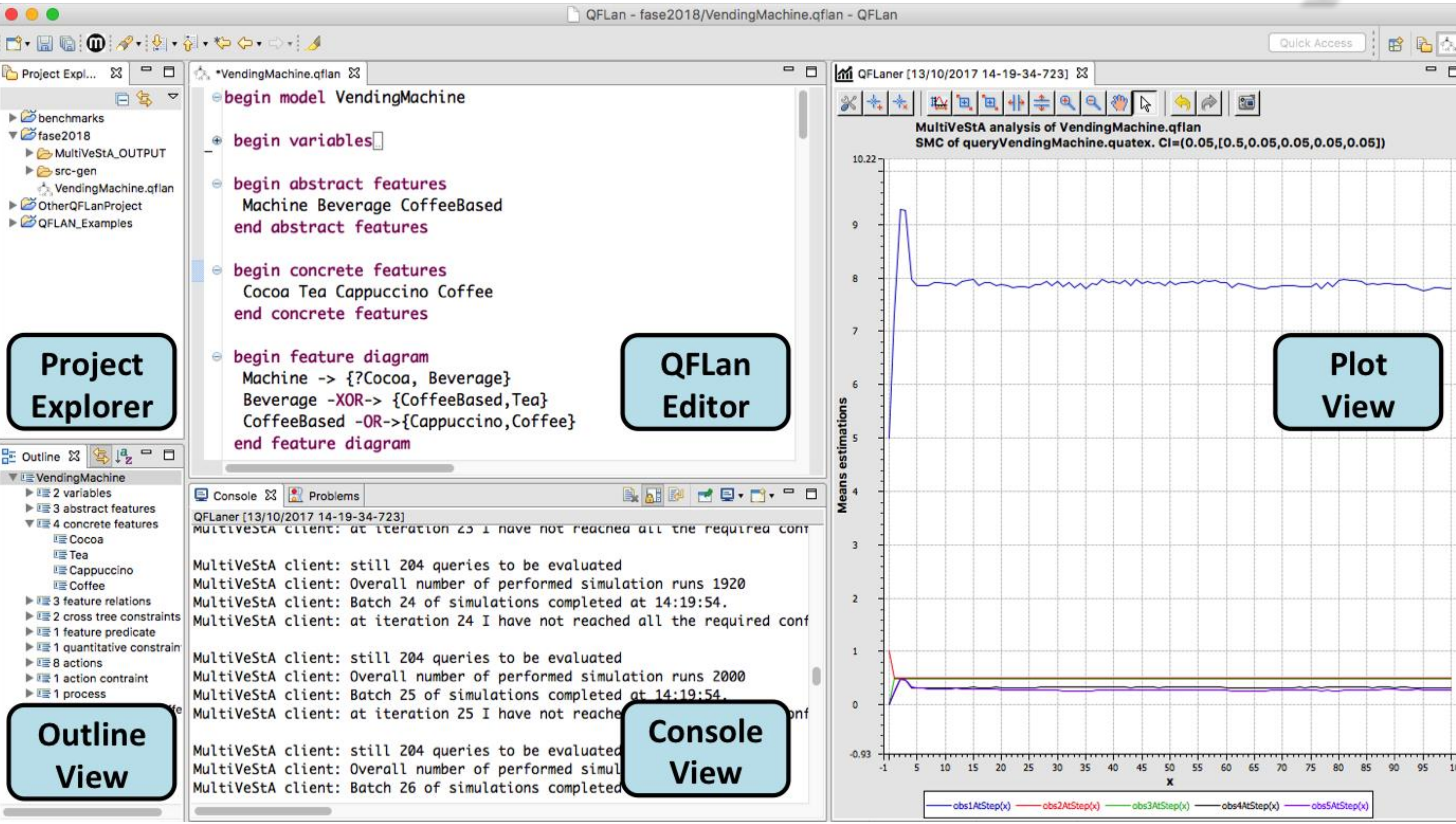
VMC offers both product-based and family-based analyses



http://fmt.isti.cnr.it/vmc/

Quantitative Modelling and Analysis of Highly (re)Configurable Systems

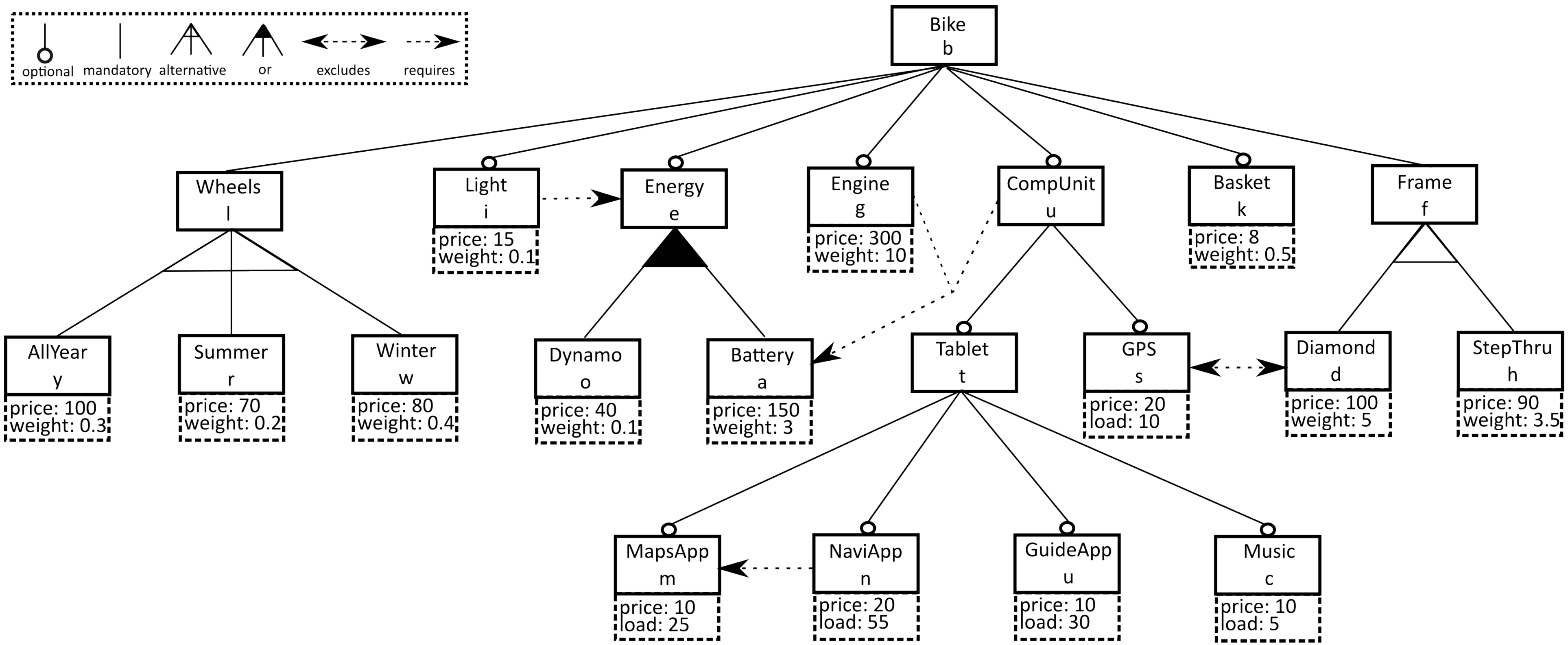




https://github.com/qflanTeam/QFLan/

TSE, FM, SPLC

A Smart Bike Product Line's Structural Constraints



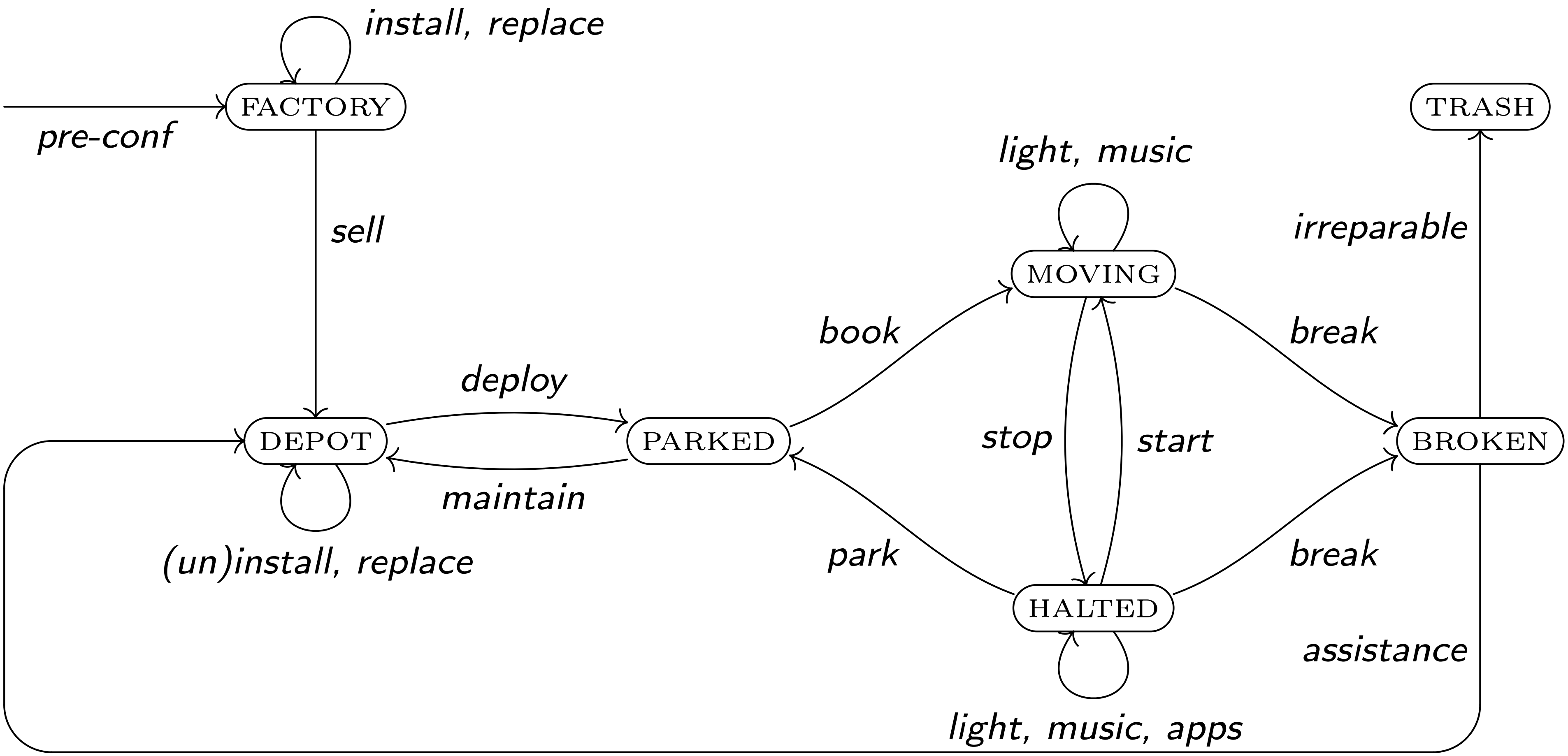
Additional feature constraints:

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

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

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

A Smart Bike Product Line's Behavioural Constraints



Additional action constraints:

$$do(sell) \;\rightarrow\; \sum_{f \in \mathcal{P}_{\mathcal{F}}} price(f) \geq 250$$
$$do(irreparable) \;\rightarrow\; \sum_{f \in \mathcal{P}_{\mathcal{F}}} price(f) \leq 400$$

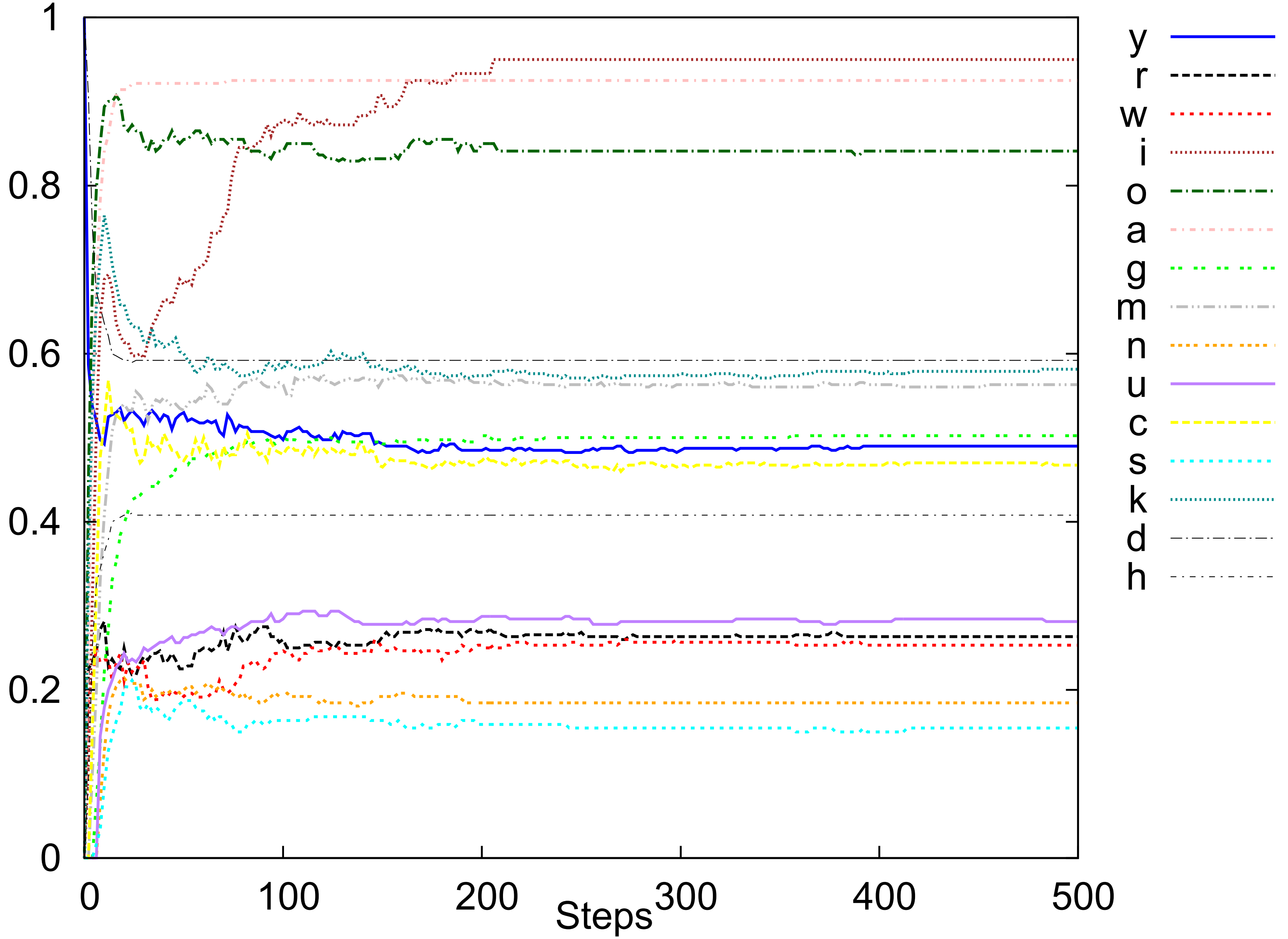
Statistical Model Checking with QFLan

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 mCRL2 analysing system behaviour [Groote et al.]

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

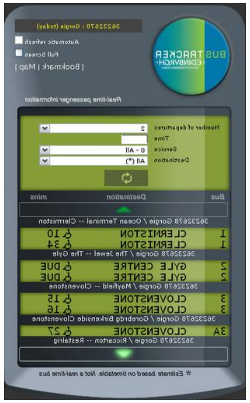
COOPERATION

FUNDAMENTALS OF COLLECTIVE ADAPTIVE SYSTEMS

FOCAS

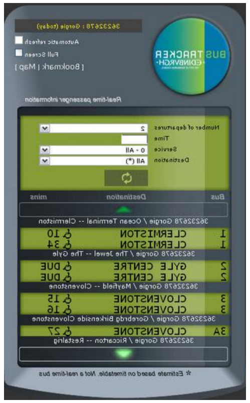
quanticol

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:

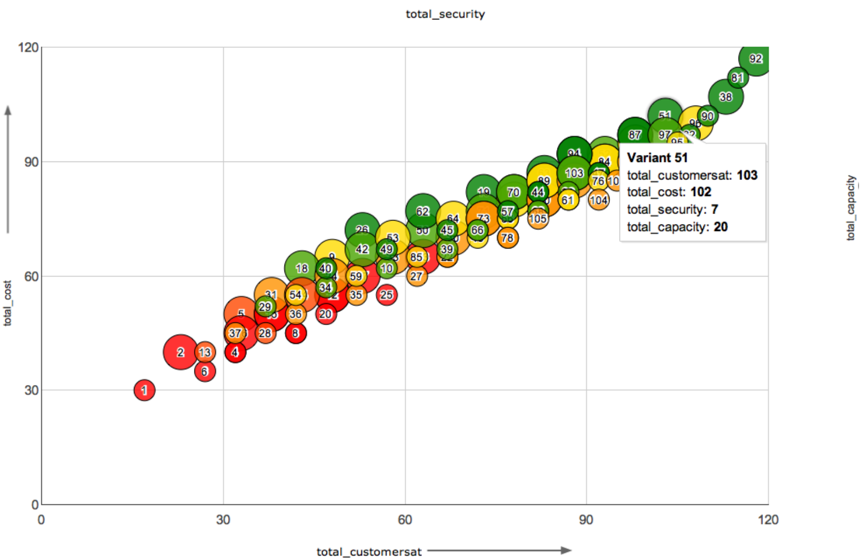
- Scalable formal verification approaches for CAS
- Quantitative business models and product lines

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



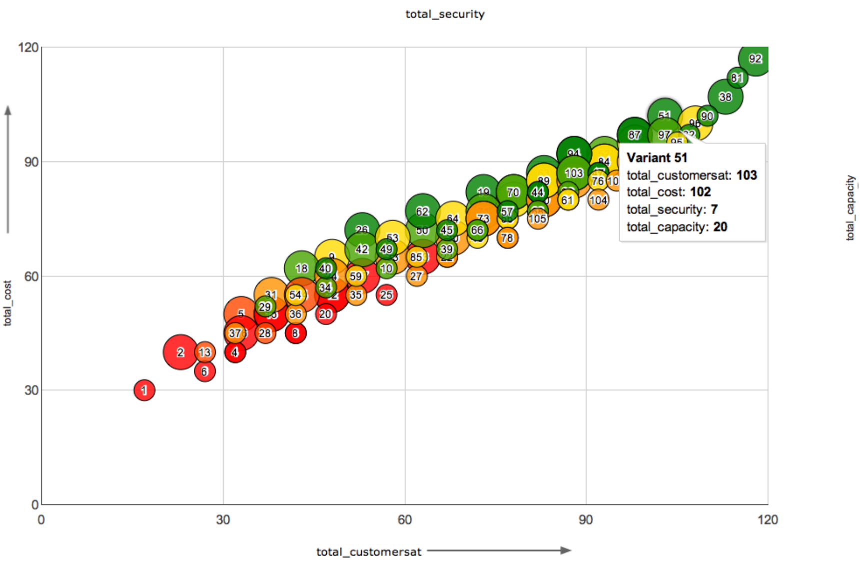
MOO attributed feature models with Clafer [Czarnecki et al.]:

- 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

SEFM, SPLC

Variability Analysis of Pisa's Bike-sharing System

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



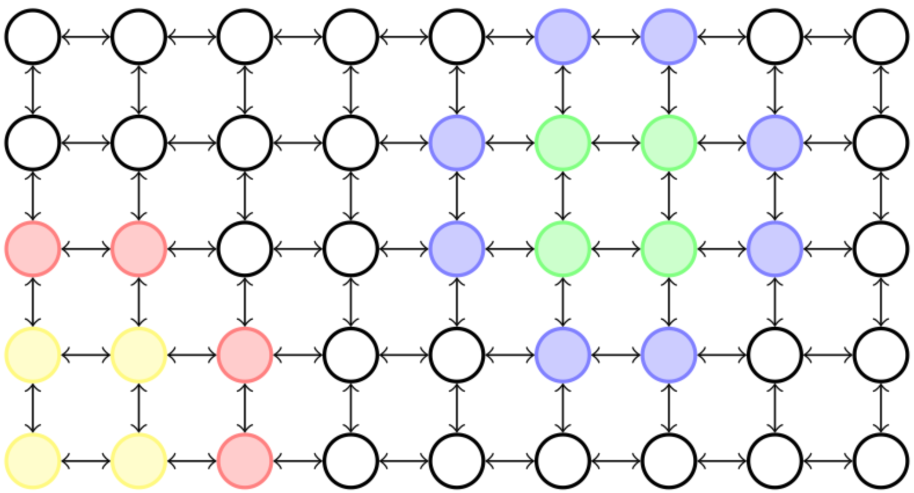
Mean field model checking with FlyFast:

- 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?

SEFM, SPLC

Statistical Spatio-temporal Model Checking

Reachability properties in graphs (discretised physical space)



$$
\begin{array}{llll}
\Phi & ::= & p & [\textsc{Atomic proposition}] \\
 & | & \top & [\textsc{True}] \\
 & | & \neg\Phi & [\textsc{Not}] \\
 & | & \Phi \wedge \Phi & [\textsc{And}] \\
 & | & \mathcal{N}\Phi & [\textsc{Near}] \\
 & | & \Phi\,\mathcal{S}\,\Phi & [\textsc{Surrounded}]
\end{array}
$$

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

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

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

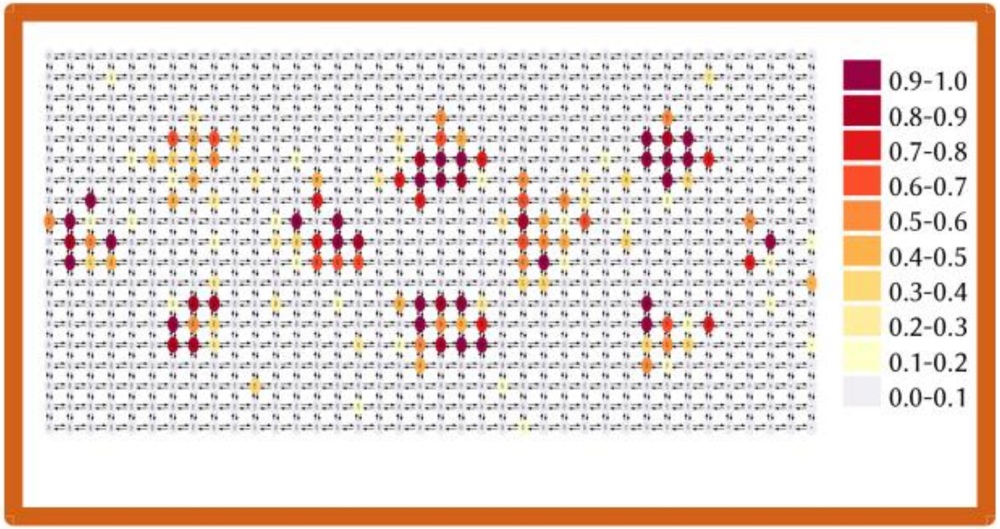
No points satisfy: $\mathcal{I}\;green$

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

LMCS, STTT, ISoLA

Statistical Spatio-temporal Model Checking

Reachability properties in graphs (discretised physical space)



$$
\begin{array}{lcll}
\Phi & ::= & p & [\text{ATOMIC PROPOSITION}] \\
& \mid & \top & [\text{TRUE}] \\
& \mid & \neg\Phi & [\text{NOT}] \\
& \mid & \Phi \wedge \Phi & [\text{AND}] \\
& \mid & \mathcal{N}\Phi & [\text{NEAR}] \\
& \mid & \Phi\,\mathcal{S}\,\Phi & [\text{SURROUNDED}]
\end{array}
$$

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

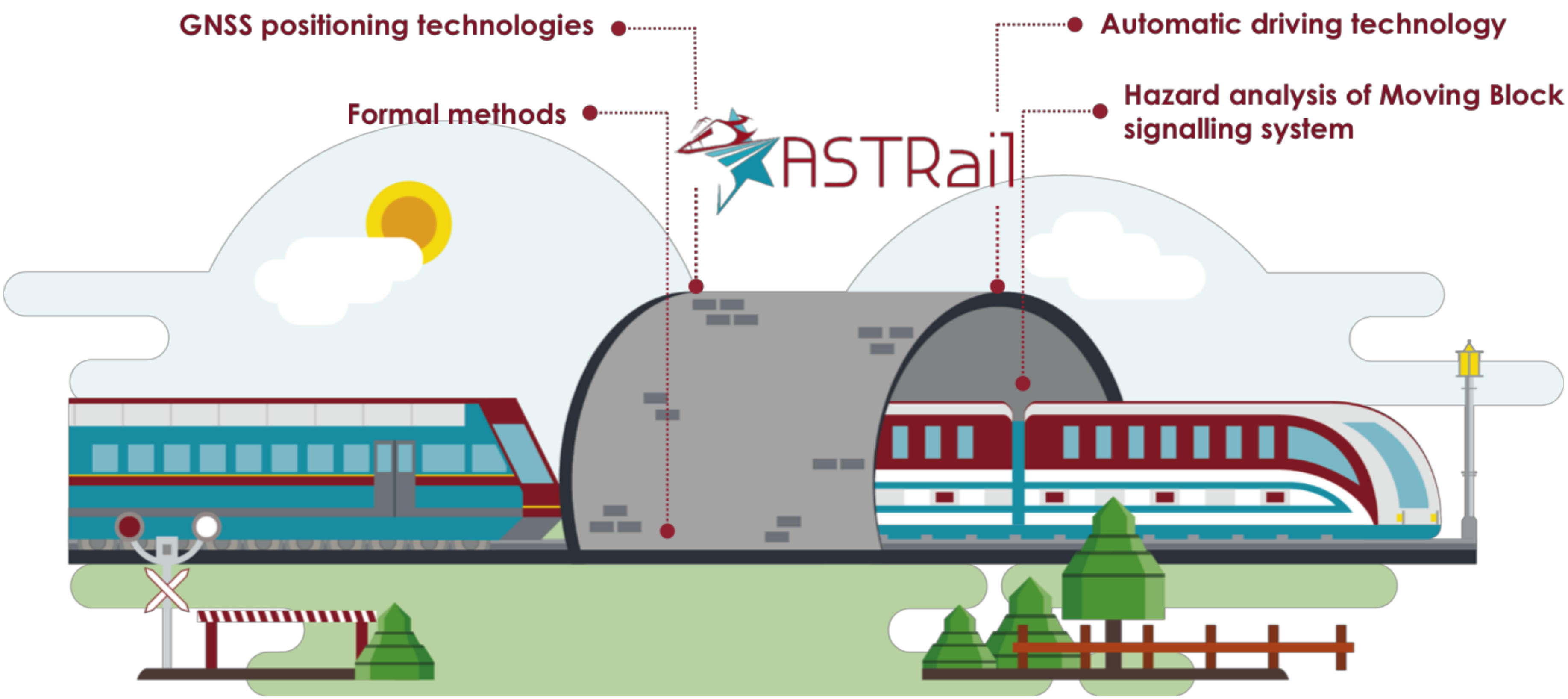
LMCS, STTT, ISoLA

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”

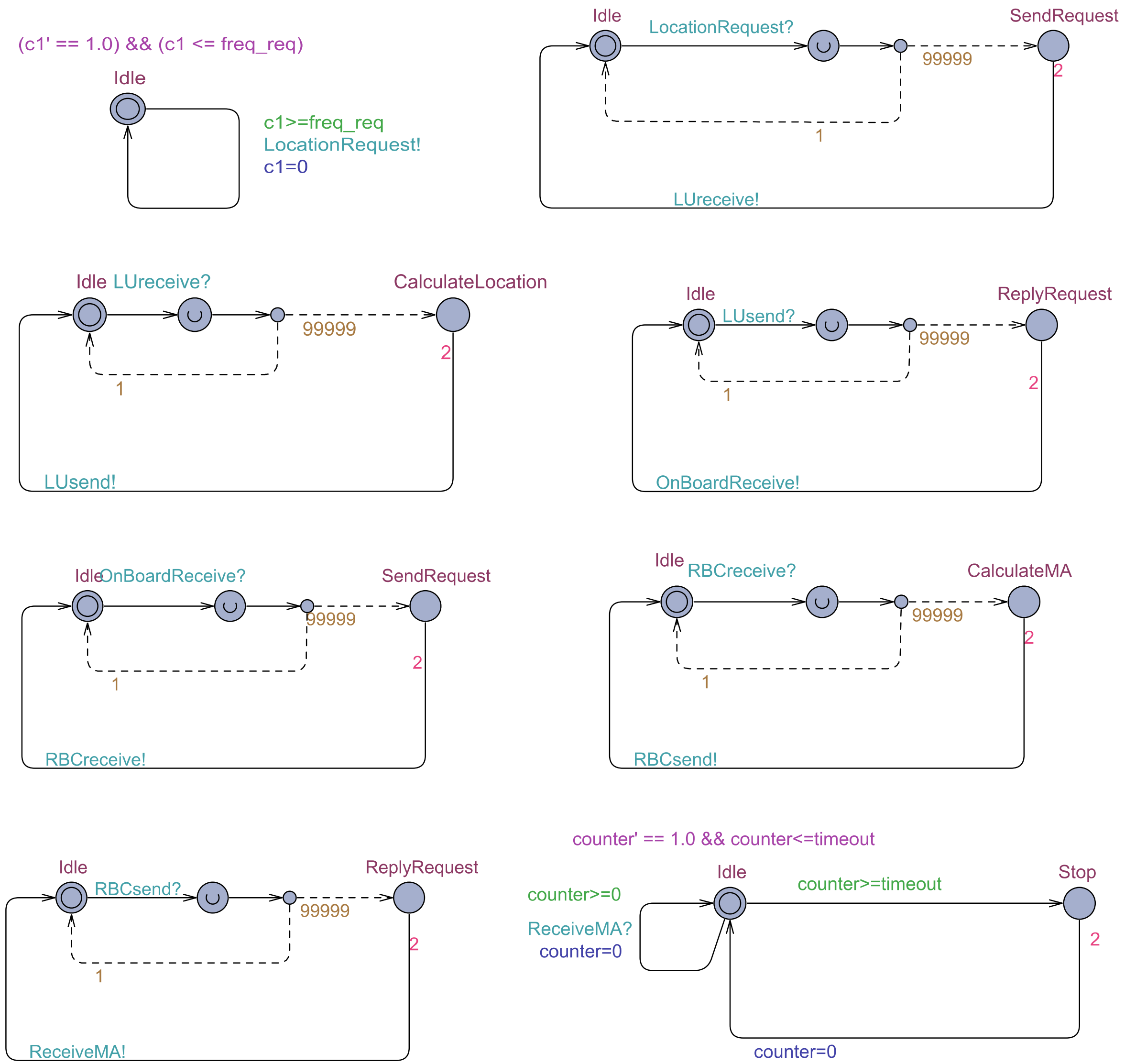
H2020 Shift2Rail Initiative: 920M€ (2014–20)

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

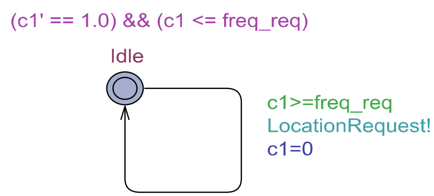


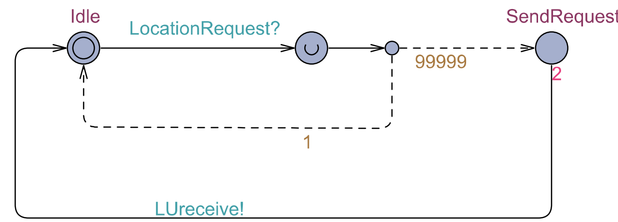
http://www.astrail.eu/

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

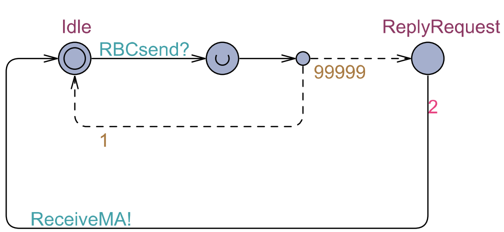


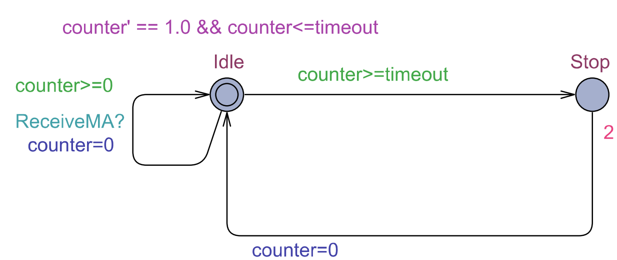


Probability that train enters safe state Stop upon timeout:

$$\mathbb{P}_M(\Diamond_{\leq(\texttt{timeout})}\ \texttt{Controlling.Stop})$$

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

2014 2020 POR CreO
PROGRAMMA OPERATIVO REGIONALE
CRESCITA e OCCUPAZIONE

StingRay

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

Unione Europea
Fondo Europeo di Sviluppo Regionale

Regione Toscana

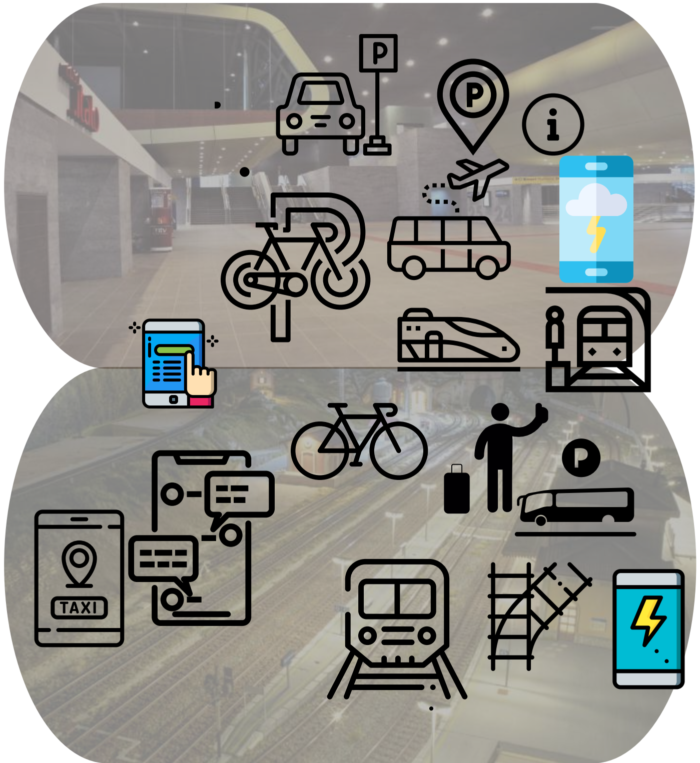
2014 2020 POR CreO
PROGRAMMA OPERATIVO REGIONALE
CRESCITA e OCCUPAZIONE

StingRay

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



Thanks for your attention!

Hope to meet some of you again in Porto later this year