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 (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
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 m²
  >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

Techniques and Tools → Application Domains

- Formal Methods
- Code Test
- Requirements Design
- Natural Language Processing
- Model Checking
- Software Development Process
Formal Methods and Tools: Techniques


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 (autl) information. 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...

1 11. 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 center
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

QuARS - version 4.1

QuARS Dictionaries: [Syntactic] implicitness

<table>
<thead>
<tr>
<th>Implicitity</th>
<th>Multiplicity</th>
<th>Underspecification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Above</td>
<td>Below</td>
<td>Following</td>
</tr>
<tr>
<td>Previous</td>
<td>Them</td>
<td>These</td>
</tr>
<tr>
<td>This</td>
<td>Those</td>
<td></td>
</tr>
</tbody>
</table>

QuARS Output

----------- QuARS [Syntactic] implicitness ANALYSIS -----------

The line number:
2. 222: each pp claim shall identify security objectives and if security requirements statements contain an implicit sentence: implicit object

Syntax Parser

Lexical Parser

Indicators Detector

QuARS

Domain dictionaries

Views derivation

Graphics

Indicator related dictionaries

QuARS Sentences Input file: requirements.txt

1 1. Requirements for the functions
2 222: Each PP claim shall identify security objectives and if security requirements statements contain an implicit sentence: implicit object
3 417: The system shall provide cap...
4 451: The system shall provide cap...
5 469: The system shall provide the ...
6 479: The system shall maintain the ...
7 499: The system shall provide a co...

http://quars.isti.cnr.it/
# QuARS at Work: Defect Examples

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Responsibility</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Assist Troubleshooting</td>
<td>Schmalfeldt</td>
<td>TDS</td>
</tr>
</tbody>
</table>

**FREQ1177**

For each error message information concerning **possible** corrective measures and consequences shall be delivered.

---

## Lexical Analysis

<table>
<thead>
<tr>
<th>Defects</th>
<th>Optionality</th>
<th>Subjectivity</th>
<th>Vagueness</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>--</td>
<td>--</td>
<td><strong>possible</strong></td>
<td>--</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Responsibility</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Converter cooling</td>
<td>Baddam</td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Defects</th>
<th>Optionality</th>
<th>Subjectivity</th>
<th>Vagueness</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>--</td>
<td>--</td>
<td><strong>sufficient</strong></td>
<td>--</td>
</tr>
</tbody>
</table>

---

CSSE, ASE, RE
## QuARS at Work: Defect Examples

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Responsibility</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Assist Troubleshooting</td>
<td>Schmalfeldt</td>
<td>TDS</td>
</tr>
</tbody>
</table>

**FREQ1177**

For each error message information concerning **possible** corrective measures and consequences shall be delivered.

*Lexical Analysis*

<table>
<thead>
<tr>
<th>Defects</th>
<th>Optionality</th>
<th>Subjectivity</th>
<th>Vagueness</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>--</td>
<td>--</td>
<td><strong>possible</strong></td>
<td>--</td>
</tr>
</tbody>
</table>

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

---

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Responsibility</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Converter cooling</td>
<td>Baddam</td>
<td></td>
</tr>
</tbody>
</table>

**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*

<table>
<thead>
<tr>
<th>Defects</th>
<th>Optionality</th>
<th>Subjectivity</th>
<th>Vagueness</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>--</td>
<td>--</td>
<td><strong>sufficient</strong></td>
<td>--</td>
</tr>
</tbody>
</table>

CSSE, ASE, RE
# QuARS at Work: Defect Examples

## Table 1: Defect Examples

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Responsibility</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Assist Troubleshooting</td>
<td>Schmalfeldt</td>
<td>TDS</td>
</tr>
</tbody>
</table>

**FREQ1177**

For each error message information concerning *possible* corrective measures and consequences shall be delivered.

---

## Lexical Analysis

<table>
<thead>
<tr>
<th>Defects</th>
<th>Optionality</th>
<th>Subjectivity</th>
<th>Vagueness</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>--</td>
<td>--</td>
<td><em>possible</em></td>
<td>--</td>
</tr>
</tbody>
</table>

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

---

## Table 2: Defect Examples

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
<th>Responsibility</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Converter cooling</td>
<td>Baddam</td>
<td></td>
</tr>
</tbody>
</table>

**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 input sensors, as fan speed, water/air temperatures, water flux, water conductivity, shall be presented. If air cooling is applied, a filter with less maintenance effort shall be used.

---

## Lexical Analysis

<table>
<thead>
<tr>
<th>Defects</th>
<th>Optionality</th>
<th>Subjectivity</th>
<th>Vagueness</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>--</td>
<td>--</td>
<td><em>sufficient</em></td>
<td>--</td>
</tr>
</tbody>
</table>

- This usage of the vague word *sufficient* might be an underspecification.
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, ...
Kan&ISTI

Family of model checkers developed by FMT for >2 decades

CMC  |  FMC  |  VMC

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!

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

Family-Based Analysis

Feature-Based Analysis
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
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

http://fmt.isti.cnr.it/vmc/
Quantitative Modelling and Analysis of Highly (re)Configurable Systems

https://github.com/qflanTeam/QFLan/
A Smart Bike Product Line’s Structural Constraints

Additional feature constraints:

\[ \sum_{f \in P_F} \text{price}(f) \leq 600: \text{a bike may cost at most 600 } € \]
\[ \sum_{f \in P_F} \text{weight}(f) \leq 15: \text{a bike may weigh up to 15 kg} \]
\[ \sum_{f \in P_F} \text{load}(f) \leq 100\%: \text{a bike’s total computational load may not exceed 100\%} \]
A Smart Bike Product Line’s Behavioural Constraints

Additional action constraints:

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

\[
\text{do(irreparable)} \rightarrow \sum_{f \in \mathcal{P}_F} \text{price}(f) \leq 400
\]
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 \(\text{mCRL2}\) [Groote et al.]

<table>
<thead>
<tr>
<th>(\varphi)</th>
<th>property</th>
<th>result</th>
<th>one-by-one</th>
<th>all-in-one</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\varphi_1)</td>
<td>Absence of deadlock</td>
<td>128/0</td>
<td>10.02</td>
<td>2.07</td>
</tr>
<tr>
<td>(\varphi_3)</td>
<td>The controller cannot fairly receive each of the three message types</td>
<td>0/128</td>
<td>24.33</td>
<td>0.25</td>
</tr>
<tr>
<td>(\varphi_5)</td>
<td>The system cannot be in a situation in which the pump runs indefinitely in the presence of methane</td>
<td>96/32</td>
<td>17.26</td>
<td>0.86</td>
</tr>
<tr>
<td>(\varphi_6)</td>
<td>Assuming fairness ((\varphi_3)), the system cannot be in a situation in which the pump runs indefinitely in the presence of methane ((\varphi_5))</td>
<td>112/16</td>
<td>27.32</td>
<td>3.67</td>
</tr>
<tr>
<td>(\varphi_7)</td>
<td>The controller can always eventually receive/read a message, i.e. it can return to its initial state from any state (CTL/LTL formula)</td>
<td>128/0</td>
<td>18.36</td>
<td>2.40</td>
</tr>
<tr>
<td>(\varphi_9)</td>
<td>Invariantly, when the level of methane rises, it inevitably decreases</td>
<td>0/128</td>
<td>20.47</td>
<td>0.21</td>
</tr>
<tr>
<td>(\varphi_{11})</td>
<td>Products with feature Ct can always switch on the pump</td>
<td>28/100</td>
<td>21.11</td>
<td>2.32</td>
</tr>
<tr>
<td>(\varphi_{12})</td>
<td>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)</td>
<td>128/0</td>
<td>13.35</td>
<td>3.36</td>
</tr>
</tbody>
</table>

Runtime improvement w.r.t. product-based model checking: average speed up of \(\pm 31\) - ranging from \(\pm 4\) (\(\varphi_{12}\)) to \(> 97\) (\(\varphi_3\))
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)

\[ \Phi ::= \rho \quad \text{[ATOMIC PROPOSITION]} \]
\[ \top \quad \text{[TRUE]} \]
\[ \neg \Phi \quad \text{[NOT]} \]
\[ \Phi \land \Phi \quad \text{[AND]} \]
\[ \land \Phi \quad \text{[NEAR]} \]
\[ \Phi \land \Phi \quad \text{[SURROUNDED]} \]

Derived operators, like interior: \( \mathcal{I} \Phi = \neg \land \neg \Phi \)

All red and yellow points satisfy: \( \land \text{yellow} \)
One yellow point satisfies: \( \mathcal{I} \text{yellow} \)
No points satisfy: \( \mathcal{I} \text{green} \)
Green points satisfy: \( \text{green} \land \text{blue} \)
Spot congestion in bike-sharing system:

full = [vacantPlaces == 0]
cluster = I full
eventuallyCluster = EF cluster
“Formal methods are fundamental for safe and reliable technological advances to increase the competitiveness of the European rail industry”

“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
Probability that train enters safe state Stop upon timeout:

\[ P_M(\Diamond_{\leq \text{timeout}} \text{Controlling.Stop}) \]

Uppaal SMC [Larsen et al.] reports that this probability is in the interval \([0, 9.99994\times10^{-5}]\), with confidence 0.995, obtained from 59912 runs in ±5 minutes (\(M\) is the model)
Renew the role of railway stations in the future’s smart cities
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.