Using mCRL2 for the Analysis of Software Product Lines

Maurice H. ter Beek; Erik P. de Vink

ISTI-CNR, Pisa, Italy; TU Eindhoven, The Netherlands

FormaliSE'14
Hyderabad, India
June 3rd 2014

Outline

QUANTICOL: A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours

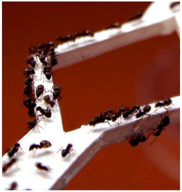
EU FP7-ICT FET-Proactive STREP: 1 April 2013 – 31 March 2017

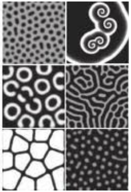
- University of Edinburgh, Scotland, Jane Hillston (Coordinator)
- CNR-ISTI, Pisa, Italy, Mieke Massink
- University of Southampton, England, Mirco Tribastone
- EPFL, Lausanne, Switzerland, Jean-Yves Le Boudec
- IMT Lucca, Italy, Rocco De Nicola

Collective Adaptive Systems (CAS)

quanticol
www.quanticol.eu

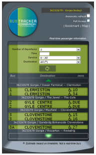
Many examples of decentralized collective adaptive behavior in nature



QUANTICOL: focus on applications arising in context of smart cities





Highly distributed systems with adaptive behavior relying on continuous feedback of vast numbers of consumers and producers

Characteristics of CAS and the project

- Coordination based on (local) decentralized interaction
- Large scale, heterogeneous agents, competing goals, open
- Capacity to smoothly adapt to changing circumstances
- Spatially inhomogeneous distribution influences global patterns
- Multiple scales in time and space, systems of systems
- Decentralized and centralized control

Vision: develop an innovative formal design framework consisting of

- mathematical (quantified) representations of the dynamic behavior of spatially inhomogeneous CAS
- a formal specification language and quantified logic for CAS
- tool-supported, scalable analysis and verification techniques
- design patterns for emergent behavior and control over spatially distributed CAS

QUANTICOL @ CNR-ISTI-FMT

quanticol
www.quanticol.eu

- Scalable verification approaches (model checkers)
- Quantitative business models and product families

Concrete case study on bike-sharing systems (BSS)

- Popular sustainable means of transportation in urban environment
- Challenging case study offering interesting runtime optimization problems and exhibiting variability in the kind of features and in their quantitative characteristics

T3.3 Relating local and global system views with variability analysis

- Study relations between (representations of) small populations and compact (family) representation of large population 'built' from them by indicating the commonalities and variabilities of single entities in their overall environment

(Software) Product Line Engineering

quanticol
www.quanticol.eu

To develop a family of products (product line) using a shared platform or architecture (commonalities) and mass customization (variabilities)

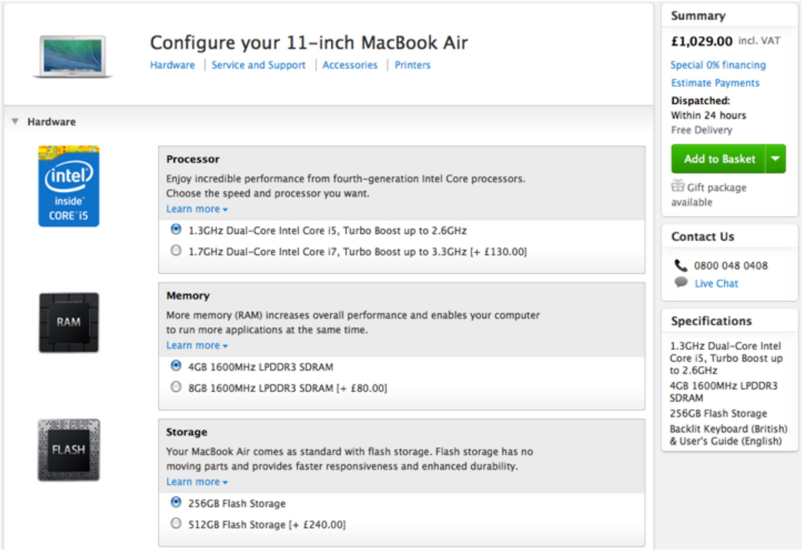
Aim: maximize commonalities whilst minimizing cost of variations (i.e., of individual products), thus specifically facilitating (software) reuse in a predictive manner

Variability in terms of features:

- End-user visible pieces of functionality that represent both commonalities (e.g., mandatory, required) and variabilities (e.g., optional, alternative)
- Only specific combinations of features concern valid products

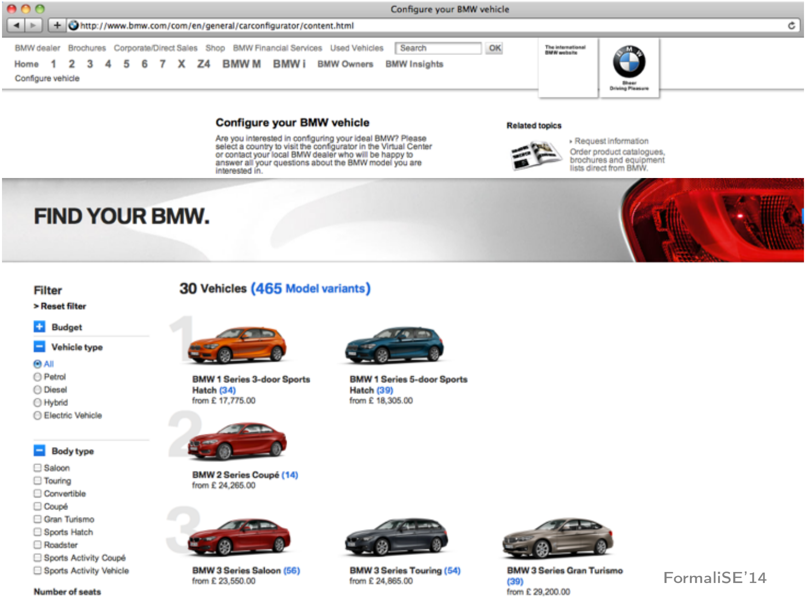
Complex: *"We always have 126,000,000 different bicycles in store! But only the parts for 1,000..."*

Example (1/2)


Configure your 11-inch MacBook Air
Hardware | Service and Support | Accessories | Printers
Hardware
intel inside CORE i5
Processor
Enjoy incredible performance from fourth-generation Intel Core processors. Choose the speed and processor you want.
Learn more
1.3GHz Dual-Core Intel Core i5, Turbo Boost up to 2.6GHz
1.7GHz Dual-Core Intel Core i7, Turbo Boost up to 3.3GHz [+ £130.00]
RAM
Memory
More memory (RAM) increases overall performance and enables your computer to run more applications at the same time.
Learn more
4GB 1600MHz LPDDR3 SDRAM
8GB 1600MHz LPDDR3 SDRAM [+ £80.00]
FLASH
Storage
Your MacBook Air comes as standard with flash storage. Flash storage has no moving parts and provides faster responsiveness and enhanced durability.
Learn more
256GB Flash Storage
512GB Flash Storage [+ £240.00]
Summary
£1,029.00 incl. VAT
Special 0% financing
Estimate Payments
Dispatched:
Within 24 hours
Free Delivery
Add to Basket
Gift package available
Contact Us
0800 048 0408
Live Chat
Specifications
1.3GHz Dual-Core Intel Core i5, Turbo Boost up to 2.6GHz
4GB 1600MHz LPDDR3 SDRAM
256GB Flash Storage
Backlit Keyboard (British) & User's Guide (English)

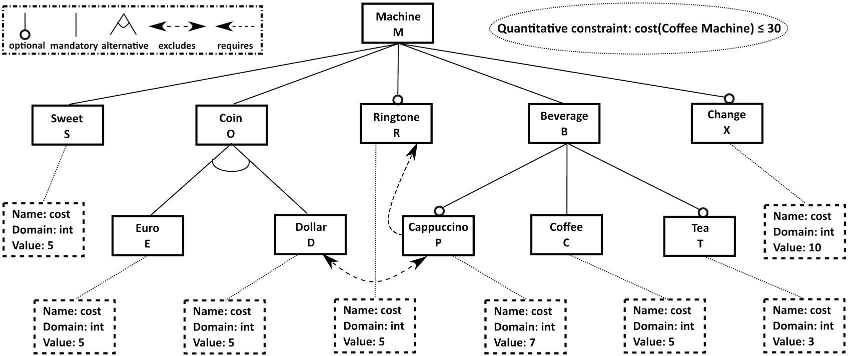
Example (2/2)

quanticol
www.quanticol.eu



Attributed feature model: A compact representation of all the family's products

quanticol
www.quanticol.eu

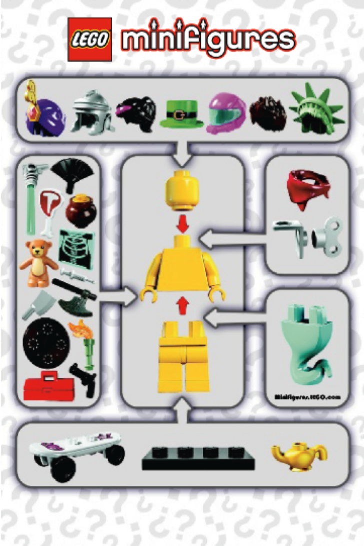


Non-functional attributes: $cost\,(product) = \sum\{\ cost\,(feature) \mid feature \in product\ \}$

From $2^{10} - 1 \xrightarrow{\text{feature diagram}} 2^5 \xrightarrow{\text{cross-tree constraints}} 20 \xrightarrow{\text{attributes}} 16$ valid products !

Family of 16 valid products
(i.e., feature combinations)

'feature model'
(allowing >16)



Formal methods in SPLE

Computer-aided analysis of variability models

- Traditionally: focus on modeling/analyzing structural constraints
- But: software systems often embedded/distributed/safety-critical
- Important: model/analyze also behavior (e.g., quality assurance)

Goal: rigorously establish critical requirements of (software) systems
⇒ lift success stories from single product/system engineering to SPLE

Recent approaches to formally model behavioral variability:

- Variants of UML diagrams (Haugen et al., Jézéquel et al.)
- Extensions of Petri nets (Clarke et al.)
- Models with LTS-like semantics: MTS (Fischbein et al., Fantechi et al.), I/O automata (Larsen et al., Lauenroth et al.), CCS/CSP (Gruler et al., Gnesi et al., ter Beek et al.), FTS (Classen et al.), FSM (Millo et al.)

Scalability is a major issue!
(slide by C. Kästner, CMU)

quanticol
www.quanticol.eu

with 33 optional, independent features

a unique product for every

person on this planet

Using mCRL2 for behavioral analysis

quanticol
www.quanticol.eu

"adopt and extend state-of-the-art analysis tools"

"examine[s] only valid product variants"

"visualize and (manually or automatically) analyze feature combinations corresponding to products of the product line"

"support (feature) modularity"

Recommendations for Improving the Usability of Formal Methods for Product Lines
(J.M. Atlee, S. Beidu, N.A. Day, F. Faghih & P. Shaker @ FormaliSE'13)

Modularization (in a feature-oriented fashion) are made concrete in ter Beek & de Vink @ ISoLA'14

(Fisler & Krishnamurthi @ ESEC/FSE'01 first recognized that most properties of interest naturally decompose around features)

mCRL2: www.mcrl2.org (open source)

quanticol
www.quanticol.eu

Formal, process-algebraic specification language for distributed and concurrent systems + associated industrial-strength toolset ($>$60)

Built on μCRL (1990), mCRL2 since 2003, now actively maintained

Up to 10^5 states per second, state spaces of size 10^9 are the norm

Symbolic exploration of 10^6 states per second, state spaces of 10^{12}

Built-in datatypes (Bool, Int, Real, Sets, Functions) + user-defined abstract datatypes to parametrize actions

Formal methods used incl. linear processes, (parametrized) Boolean equation systems, LTS, modal μ-calculus with data (incl. LTL, CTL)

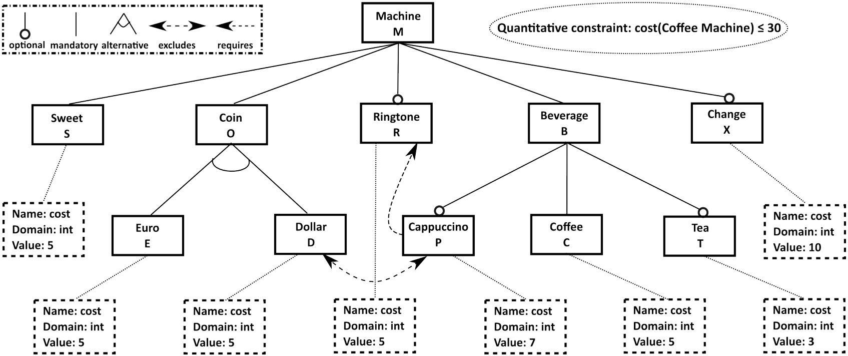
Simulation, visualization, behavioral reduction, model checking, etc.

Highly optimized, up-to-date (i.e. best-known algorithms implemented)

A family of coffee machines

- Initially, money must be inserted: either at least one euro's worth in coins, *exclusively* for European products, or at least one dollar's worth in coins, *exclusively* for Canadian products
- Input of money can be canceled via a cancel button. *Optionally*, the machine returns change after more than one euro or one dollar was inserted
- Once the machine contains at least one euro or one dollar, the user has to choose whether (s)he wants sugar, by pressing one of two buttons, after which (s)he can select a beverage
- The choice of beverage (coffee, tea, cappuccino) varies, but coffee *must* be offered by *all* products whereas cappuccino *may* be offered *solely* by European products
- *Optionally*, a ringtone *may* be rung after delivering a beverage. A ringtone *must* however be rung by *all* products offering cappuccino
- After the beverage is taken, the machine returns idle

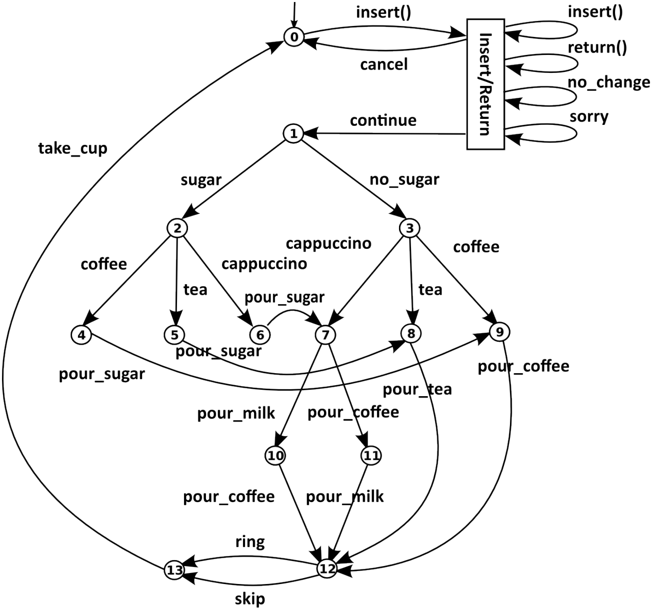
Recall: attributed feature model



Selection process: specification induced by feature diagram, CTCs, cost


```
proc Sel(st:Int,fs:FSet) =
...
(st == 1) -> ( (M in fs) -> ( set0 . Sel(2,insert(O,fs) ) ) )
+
(st == 2) -> ( (M in fs) -> ( tau . Sel(3,fs) +
                              setR . Sel(3,insert(R,fs)) ) )
+
...

(st == 8) -> ( ( (D in fs) && (P in fs) ) -> wrong_set . delta <>
               ( !(R in fs) && (P in fs) ) -> wrong_set . delta <>
               ctc_ok . Sel(9,fs) )
+

(st == 9) -> ( (tcost(fs) <= 30) ->
               set_ok(fs) . cost(tcost(fs)) . Prod(0,fs) <>
               wrong_set . delta )
;
```

LTS modeling family behavior

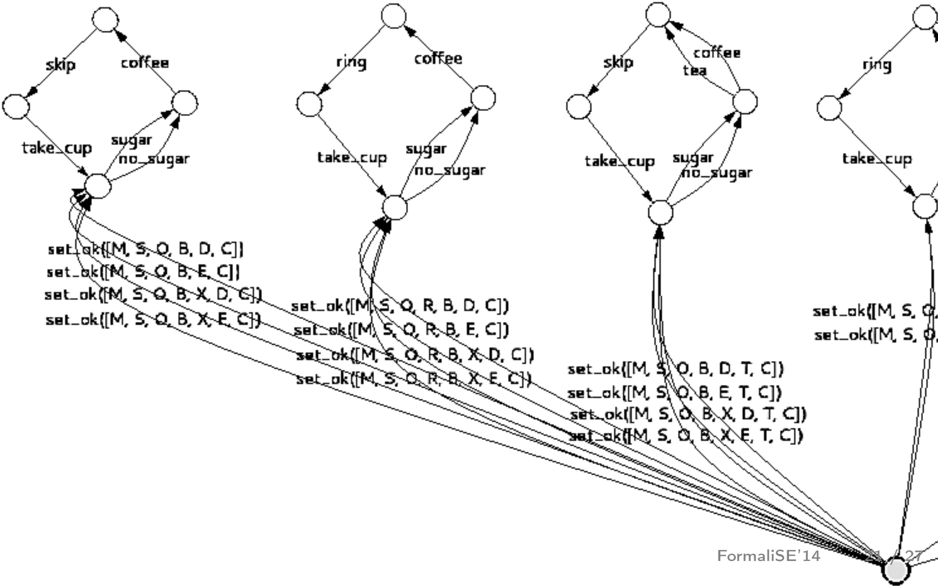


Product + Insert processes: specification of valid behavior induced by LTS

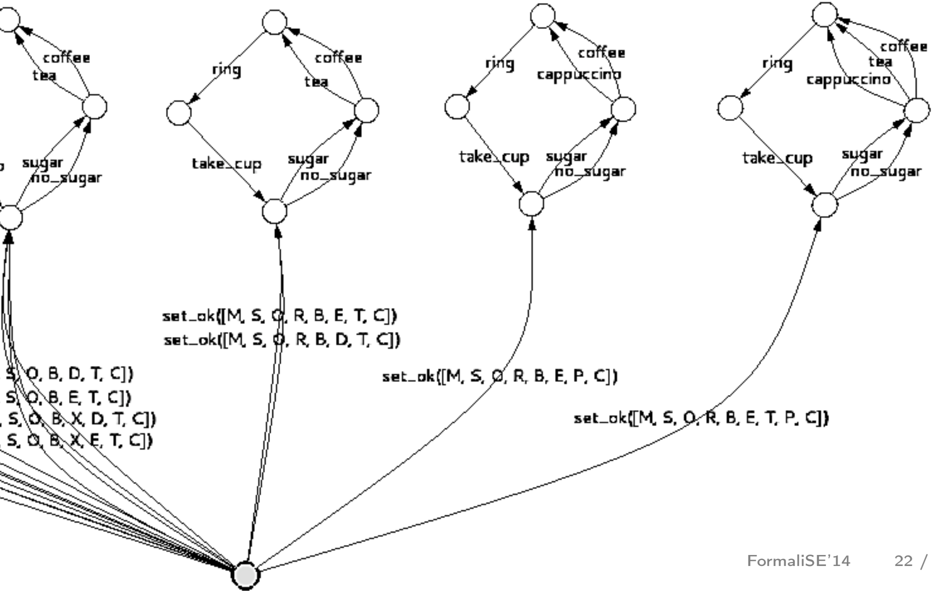

```
proc Prod(st:Int,fs:FSet) =
(st == 0) -> ( Insert(0,fs) ) +
...
(st == 2) -> ( (C in fs) -> coffee . Prod(4,fs) +
               (T in fs) -> tea . Prod(5,fs) +
               (P in fs) -> cappuccino . Prod(6,fs) ) +
...

proc Insert(bal:Nat,fs:FSet) =
(bal < 100) -> (
  (D in fs ) -> ( ... ) +
  (E in fs) -> ( insert(ct10) . Insert(bal+10,fs) +
                 insert(ct20) . Insert(bal+20,fs) +
                 insert(ct50) . Insert(bal+50,fs) +
                 insert(euro) . Insert(bal+100,fs) ) ) +
((bal > 0) && (bal < 100)) -> Return(bal,fs) . cancel . Prod(0,fs) +
(bal >= 100) ->
  ( ( ( !(X in fs) ) -> no_change . continue . Prod(1,fs) <>
  Return(Int2Nat(bal-100),fs) . continue . Prod(1,fs) ) )
;
```

Product behavior abstracting from . . .

quanticol
www.quanticol.eu



. . . configuration and payment

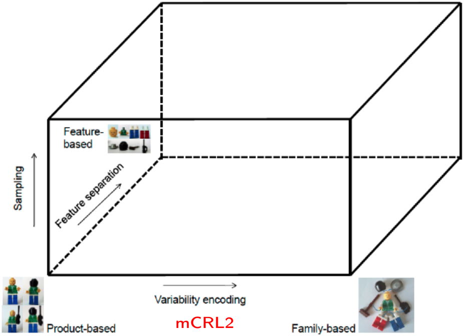


Model checking properties in modal μ-calculus with data using mCRL2

- If payment is not settled by action `continue`, no beverage is delivered:
  ```
  [(!continue)*.take_cup] false
  ```
- Once the X-feature is selected, action `no_change` will not occur:
  ```
  [true*.setX.true*.no_change] false
  ```
- If a product is configured successfully as indicated by the `set_ok` action, then it cannot be a product that accepts dollars and provides cappuccino:
  ```
  forall fs:FSet.val(isSet(fs)) && [true*. set_ok(fs)]true =>
  val((D in fs) => !(P in fs))
  ```
- From the initial state, after a finite number of steps, either action `set_ok` (with some parameter `fs`) or action `wrong_set` occurs:
  ```
  mu Y.(<exists fs:FSet.set_ok(fs)>true || <wrong_set>true ||
  [true]Y)
  ```
- After money has been inserted, in a finite number of steps, a beverage can be taken unless the transaction was canceled:
  ```
  forall c:Coin.[true*.insert(c)]
  mu Y.(<cancel || take_cup> true || [true]Y )
  ```

Position in the PLA cube (Apel et al.)

Sampling based on coverage criteria such as pairwise or *t*-wise coverage (or other heuristics)



Possible trade-off?

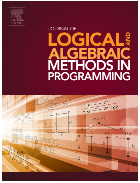
- Brute-force product-based analysis with model checkers highly optimized for single system engineering (e.g., SPIN, mCRL2)
- Highly innovative family-based analysis with model checkers developed specifically for SPL (e.g., SNIP by Classen et al.)

⇒ An evaluation of mCRL2 might lead to the desire to implement some SPL-specific features into its model-checking algorithms

Publicity (1/3): Open Call for Papers

JOURNAL OF
LOGICAL AND ALGEBRAIC METHODS IN PROGRAMMING

Special Issue on
Formal Methods in Software Product Line Engineering



Submission of papers: July 15, 2014

First review decision: December 15, 2014

Revision due: February 15, 2015

Acceptance notification: April 15, 2015

Final manuscript due: June 15, 2015

Expected publication: Summer 2015

ELSEVIER

Guest editors:

- Maurice ter Beek (ISTI-CNR, Pisa, Italy)
- Dave Clarke (U Uppsala, Sweden & KU Leuven, Belgium)
- Ina Schaefer (TU Braunschweig, Germany)

Publicity (2/3): SPLC'14 in Florence

quanticol
www.quanticol.eu

http://www.splc2014.net/

- Research track
- Industry track
- Demo/tool track
- 8 Tutorials
- 7 Workshops
- Doctoral symposium
- Hall of fame
- Panels
- ...

Organised by our Formal Methods and Tools lab of ISTI-CNR, Pisa

Publicity (3/3): deadlines in June!

http://www.splc2014.net/workshops.html

quanticol
www.quanticol.eu

- **REVE** 2nd Int. Workshop on REverse Variability Engineering
  http://www.isse.jku.at/reve2014
- **SUSPL** 1st Int. Workshop on Sustainability in Software Product Lines
  https://sites.google.com/site/susplworkshop/
- **DSPL** 8th Int. Workshop on Dynamic Software Product Lines
  http://www.lero.ie/dspl2014
- **SPLTea** 1st Int. Workshop on Software Product Line Teaching
  http://spltea.irisa.fr/
- **SPLat** A Workshop on Software Product Line Analysis Tools
  http://www.splat2014.org
- **MultiPLE** 2nd Int. Workshop on Multi Product Line Engineering
  https://sites.google.com/site/wmultiple2014/
- **SWORDS** SPES Workshop on Challenges And Deployable Solutions for Seamless Variant Management
  http://swords.in.tum.de