

VMC: A Tool for Product Variability Analysis

Maurice H. ter Beek
ISTI-CNR, Pisa, Italy

joint work with fellow FMT lab members:
P. Asirelli, A. Fantechi, S. Gnesi, F. Mazzanti & A. Sulova

FM 2012

Paris, France
29 August 2012

Outline

- 1 Background and aim of our research activity
- 2 Running example: a family of coffee machines
- 3 Demo of the Variability Model Checker VMC
- 4 Discussion and future work

(Software) Product Line Engineering

Paradigm

To develop a family of products (product line) using a common platform and mass customization

Aim

To lower production costs of the individual products by

- letting them share an overall reference model of the product family
- allowing them to differ w.r.t. particular characteristics to serve, e.g., different markets

Product variants can be derived from a product family, thus allowing for reuse and differentiation

Production process

Maximize commonalities of product whilst minimizing cost of variations

(Software) Product Line Engineering

Paradigm

To develop a family of products (product line) using a common platform and mass customization

Aim

To lower production costs of the individual products by

- letting them share an overall reference model of the product family
- allowing them to differ w.r.t. particular characteristics to serve, e.g., different markets

Product variants can be derived from a product family, thus allowing for reuse and differentiation

Production process

Maximize commonalities of product whilst minimizing cost of variations

(Software) Product Line Engineering

Paradigm

To develop a family of products (product line) using a common platform and mass customization

Aim

To lower production costs of the individual products by

- letting them share an overall reference model of the product family
- allowing them to differ w.r.t. particular characteristics to serve, e.g., different markets

Product variants can be derived from a product family, thus allowing for reuse and differentiation

Production process

Maximize commonalities of product whilst minimizing cost of variations

Feature modeling

Provide compact representations of all the products of a product family in terms of their *features* (pieces of functionality)

Variability modelling

How to explicitly define **optional**, **alternative**, **mandatory**, **required**, or **excluded** features of a product family as variation points

Managing variability with formal methods

Show that a certain product belongs to a product family or—instead—derive a product from a family by properly selecting features
Formally prove characteristics of products and families alike

Feature modeling

Provide compact representations of all the products of a product family in terms of their *features* (pieces of functionality)

Variability modelling

How to explicitly define **optional**, **alternative**, **mandatory**, **required**, or **excluded** features of a product family as variation points

Managing variability with formal methods

Show that a certain product belongs to a product family or—instead—derive a product from a family by properly selecting features

Formally prove characteristics of products and families alike

Variability management/modeling

Feature modeling

Provide compact representations of all the products of a product family in terms of their *features* (pieces of functionality)

Variability modelling

How to explicitly define **optional**, **alternative**, **mandatory**, **required**, or **excluded** features of a product family as variation points

Managing variability with formal methods

Show that a certain product belongs to a product family or—instead—derive a product from a family by properly selecting features
Formally prove characteristics of products and families alike

Aim of our research activity at large

Aim

- One formal framework to express both feature-based constraints over the products of a family and constraints over their behavior
- Tool support for product derivation and for the formal verification (by model checking) of properties over products and families alike

Outcome : iFM'10, ACOTA @ ASE'10, PLEASE @ ICSE'11, FMOODS'11, SEW-34 @ FM'11, SPLC'11, iFM'12, SPLC'12

- MTS: Modal Transition Systems (Larsen et al.)
- MHML: CTL-like action- and state-based branching-time temporal logic (a.k.a. VACTL)
- VMC: Variability Model Checker

Aim of our research activity at large

Aim

- One formal framework to express both feature-based constraints over the products of a family and constraints over their behavior
- Tool support for product derivation and for the formal verification (by model checking) of properties over products and families alike

Outcome : iFM'10, ACOTA @ ASE'10, PLEASE @ ICSE'11, FMOODS'11, SEW-34 @ FM'11, SPLC'11, iFM'12, SPLC'12

- MTS: Modal Transition Systems (Larsen et al.)
- MHML: CTL-like action- and state-based branching-time temporal logic (a.k.a. VACTL)
- VMC: Variability Model Checker

Feature-based constraints

- The only accepted coins are 1€, exclusively for European products, and 1\$, exclusively for Canadian products (**alternative** features)
- All products must offer coffee (**mandatory** feature); only European products may offer cappuccino (**excludes** relation among features)
- A ringtone must be rung in products offering cappuccino (**requires** relation among features), while it may be rung in other products (**optional** feature)

Behavioral constraints

- After coin insertion, user must press a button to choose whether (s)he wants sugar, after which (s)he may select a beverage
- The optional ringtone is rung after delivering a beverage

Running example: family of coffee machines

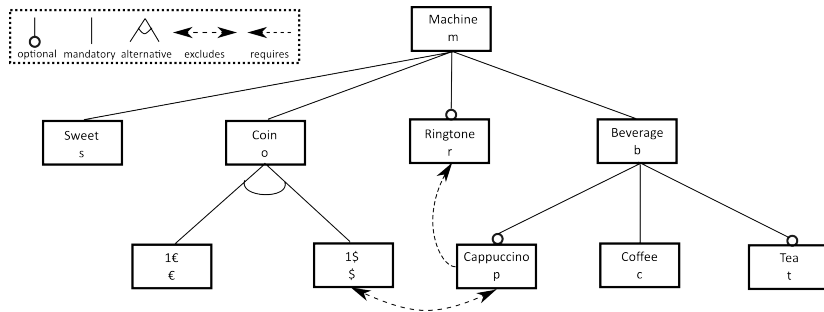
Feature-based constraints

- The only accepted coins are 1€, exclusively for European products, and 1\$, exclusively for Canadian products (**alternative** features)
- All products must offer coffee (**mandatory** feature); only European products may offer cappuccino (**excludes** relation among features)
- A ringtone must be rung in products offering cappuccino (**requires** relation among features), while it may be rung in other products (**optional** feature)

Behavioral constraints

- After coin insertion, user must press a button to choose whether (s)he wants sugar, after which (s)he may select a beverage
- The optional ringtone is rung after delivering a beverage

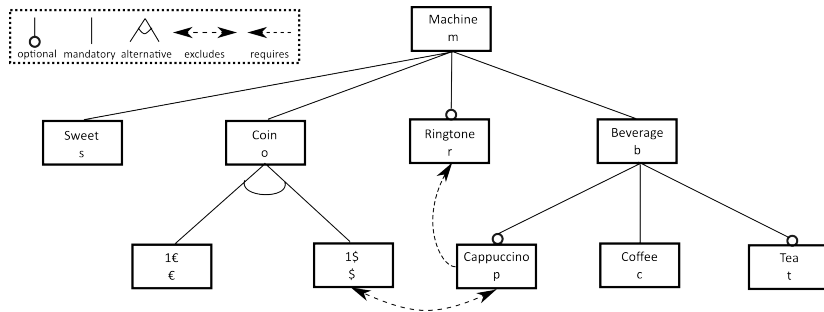
Coffee machine family: Feature model



10 different valid products (coffee machines defined by features)

$\{m, s, o, b, c, \epsilon\}$, $\{m, s, o, b, c, \epsilon, r\}$, $\{m, s, o, b, c, \epsilon, t\}$,
 $\{m, s, o, b, c, \$\}$, $\{m, s, o, b, c, \$, r\}$, $\{m, s, o, b, c, \$, t\}$,
 $\{m, s, o, b, c, \epsilon, t, r\}$, $\{m, s, o, b, c, \epsilon, p, r\}$,
 $\{m, s, o, b, c, \$, t, r\}$, $\{m, s, o, b, c, \epsilon, p, r, t\}$

Coffee machine family: Feature model



10 different valid products (coffee machines defined by features)

$\{m, s, o, b, c, \epsilon\}$, $\{m, s, o, b, c, \epsilon, r\}$, $\{m, s, o, b, c, \epsilon, t\}$,
 $\{m, s, o, b, c, \$\}$, $\{m, s, o, b, c, \$, r\}$, $\{m, s, o, b, c, \$, t\}$,
 $\{m, s, o, b, c, \epsilon, t, r\}$, $\{m, s, o, b, c, \epsilon, p, r\}$,
 $\{m, s, o, b, c, \$, t, r\}$, $\{m, s, o, b, c, \epsilon, p, r, t\}$

Modal Transition Systems (MTSs)

Use for behavioral modeling of SPLs recognized by Uchitel et al.

An MTS is an LTS distinguishing **optional** (may) and **mandatory** (must) transitions to formalize a product family's

- *underlying behavior*, shared among all products, and
- *variation points*, differentiating between products

A product (LTS) is derived by including all (reachable) must transitions and a subset of the (reachable) may transitions

MTS however cannot model variability constraints regarding **alternative** features nor regarding the **requires** and **excludes** inter-feature relations

Our solution: add a set of variability constraints (hiding our logic) to the MTS to define which derivable products must be considered valid ones (we defined an algorithm to derive only—and possibly all—valid ones)

Modal Transition Systems (MTSs)

Use for behavioral modeling of SPLs recognized by Uchitel et al.

An MTS is an LTS distinguishing **optional** (may) and **mandatory** (must) transitions to formalize a product family's

- *underlying behavior*, shared among all products, and
- *variation points*, differentiating between products

A product (LTS) is derived by including all (reachable) must transitions and a subset of the (reachable) may transitions

MTS however cannot model variability constraints regarding **alternative** features nor regarding the **requires** and **excludes** inter-feature relations

Our solution: add a set of variability constraints (hiding our logic) to the MTS to define which derivable products must be considered valid ones (we defined an algorithm to derive only—and possibly all—valid ones)

Modal Transition Systems (MTSs)

Use for behavioral modeling of SPLs recognized by Uchitel et al.

An MTS is an LTS distinguishing **optional** (may) and **mandatory** (must) transitions to formalize a product family's

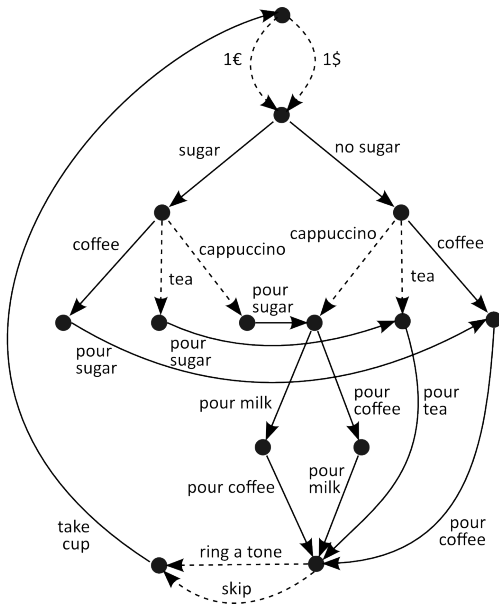
- *underlying behavior*, shared among all products, and
- *variation points*, differentiating between products

A product (LTS) is derived by including all (reachable) must transitions and a subset of the (reachable) may transitions

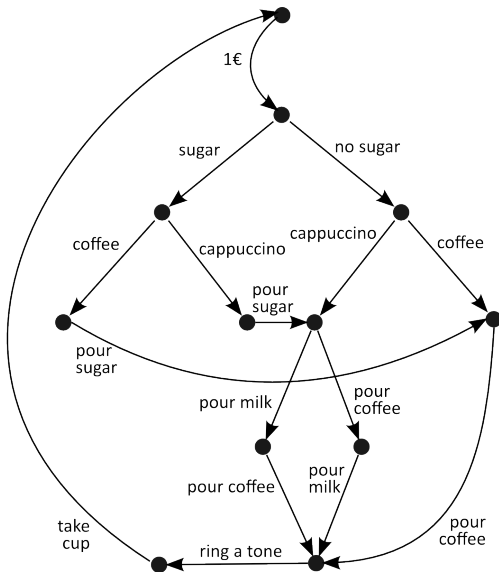
MTS however cannot model variability constraints regarding **alternative** features nor regarding the **requires** and **excludes** inter-feature relations

Our solution: add a set of variability constraints (hiding our logic) to the MTS to define which derivable products must be considered valid ones (we defined an algorithm to derive only—and possibly all—valid ones)

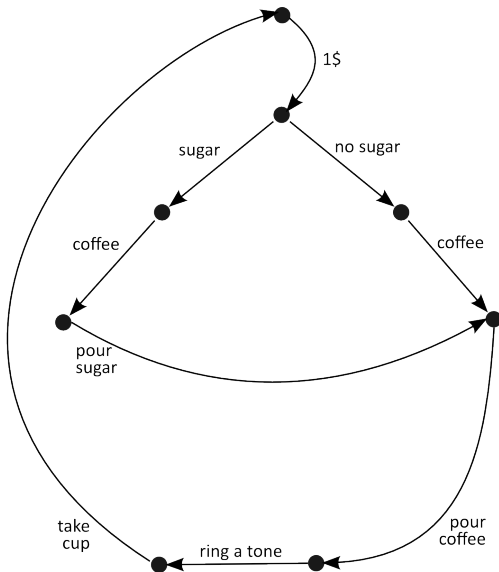
Coffee machine family: MTS



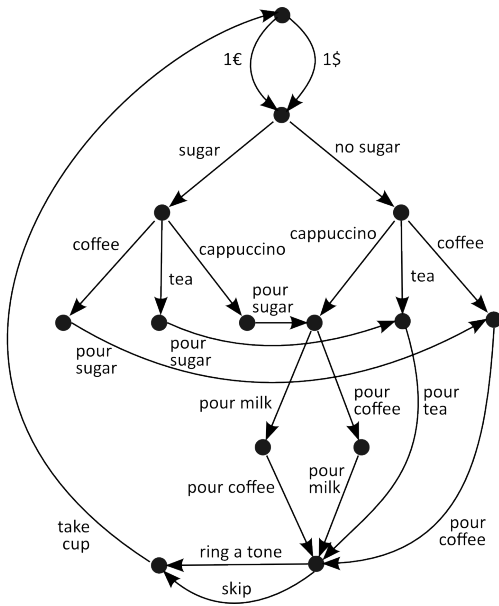
A valid European coffee machine ($\{m, s, o, b, c, \epsilon, p, r\}$)



A valid Canadian coffee machine ($\{m, s, o, b, c, \$, r\}$)



A correct but not a valid product LTS of MTS



VMC: Variability Model Checker

A tool for modeling and analysis of behavioral variability in SPLs

Given a textual encoding of an MTS and a set of variability constraints:

- interactively explore the MTS
- derive and explore (all) the family's valid products (LTSs)
- visualize the family/products graphically as MTS/LTSs
- verify branching-time temporal logic properties over family/products
- interactively explain why a product does (not) satisfy a property

Freely usable online: <http://fmtlab.isti.cnr.it/vmc/>

Verification engine and advanced explanation techniques are features of the highly optimized family of on-the-fly model checkers developed at our FMT lab over the last decades for verifying formulae in CTL-like action- and state-based branching-time temporal logics (e.g. FMC, UMC, CMC)

VMC: Variability Model Checker

A tool for modeling and analysis of behavioral variability in SPLs

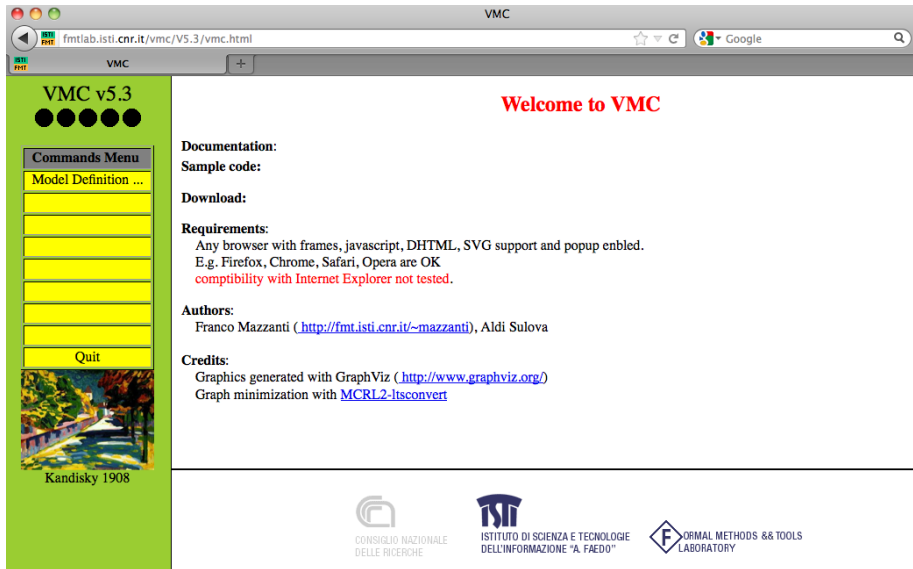
Given a textual encoding of an MTS and a set of variability constraints:

- interactively explore the MTS
- derive and explore (all) the family's valid products (LTSS)
- visualize the family/products graphically as MTS/LTSS
- verify branching-time temporal logic properties over family/products
- interactively explain why a product does (not) satisfy a property

Freely usable online: <http://fmtlab.isti.cnr.it/vmc/>

Verification engine and advanced explanation techniques are features of the highly optimized family of on-the-fly model checkers developed at our FMT lab over the last decades for verifying formulae in CTL-like action- and state-based branching-time temporal logics (e.g. FMC, UMC, CMC)

VMC's web interface



The screenshot shows a web browser window with the address bar containing `fmtlab.isti.cnr.it/vmc/V5.3/vmc.html`. The page title is "VMC". The main content area displays "Welcome to VMC" in red text. On the left, there is a green sidebar with a "Commands Menu" containing "Model Definition ...", "Quit", and a small image of a painting labeled "Kandisky 1908". The main content area lists several sections: "Documentation:", "Sample code:", "Download:", "Requirements:" (with a note about browser compatibility), "Authors:" (listing Franco Mazzanti and Aldi Sulova), and "Credits:" (listing GraphViz and MCRL2-Itsconvert).

VMC v5.3

Commands Menu

Model Definition ...

Quit

Kandisky 1908

Welcome to VMC

Documentation:

Sample code:

Download:

Requirements:
Any browser with frames, javascript, DHTML, SVG support and popup enabled.
E.g. Firefox, Chrome, Safari, Opera are OK
comptibility with Internet Explorer not tested.

Authors:
Franco Mazzanti (<http://fmt.isti.cnr.it/~mazzanti>), Aldi Sulova

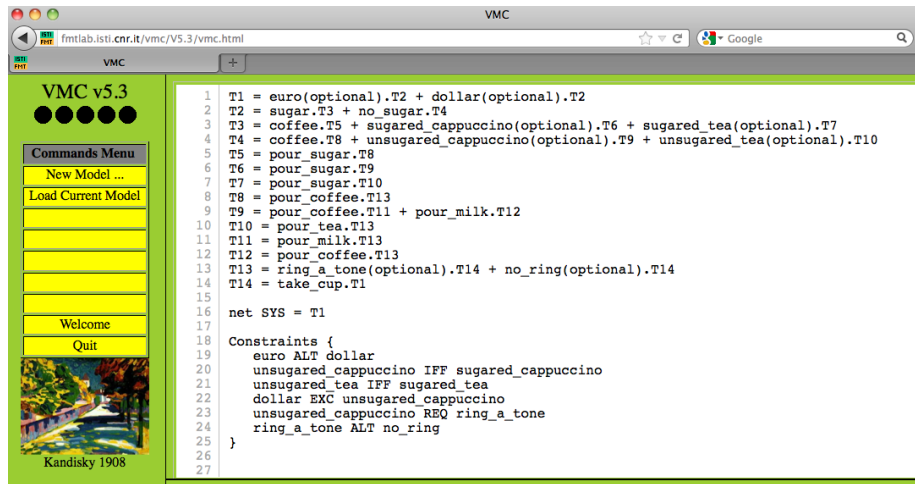
Credits:
Graphics generated with GraphViz (<http://www.graphviz.org/>)
Graph minimization with [MCRL2-Itsconvert](#)

CONSIGLIO NAZIONALE DELLE RICERCHE

ISTI
ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

FORMAL METHODS & TOOLS LABORATORY

Family of coffee machines specified in VMC



The screenshot shows a web browser window with the URL `fmlab.isti.cnr.it/vmc/V5.3/vmc.html`. The browser title is "VMC". The page content is divided into a left sidebar and a main text area.

VMC v5.3

Commands Menu

- New Model ...
- Load Current Model
- Welcome
- Quit

Kandisky 1908

```
1 T1 = euro(optional).T2 + dollar(optional).T2
2 T2 = sugar.T3 + no_sugar.T4
3 T3 = coffee.T5 + sugared_cappuccino(optional).T6 + sugared_tea(optional).T7
4 T4 = coffee.T8 + unsugared_cappuccino(optional).T9 + unsugared_tea(optional).T10
5 T5 = pour_sugar.T8
6 T6 = pour_sugar.T9
7 T7 = pour_sugar.T10
8 T8 = pour_coffee.T13
9 T9 = pour_coffee.T11 + pour_milk.T12
10 T10 = pour_tea.T13
11 T11 = pour_milk.T13
12 T12 = pour_coffee.T13
13 T13 = ring_a_tone(optional).T14 + no_ring(optional).T14
14 T14 = take_cup.T1
15
16 net SYS = T1
17
18 Constraints {
19   euro ALT dollar
20   unsugared_cappuccino IFF sugared_cappuccino
21   unsugared_tea IFF sugared_tea
22   dollar EXC unsugared_cappuccino
23   unsugared_cappuccino REQ ring_a_tone
24   ring_a_tone ALT no_ring
25 }
26
27
```

Permitted variability constraints ALternative, EXCludes, REQUIRES, and IFF (shorthand for bilateral REQs) hide the logic formalization from user

Family/MTS of coffee machines visualized by VMC

VMC v5.3

Commands Menu

- New Model ...
- Edit Current Model
- Explore the MTS
- Modelcheck MTS ...
- View Current Model
- View Family MTS
- Generate Products
- Welcome
- Quit

Kandisky 1919

Family Model Evolutions Chart (MTS)

Zoom Out Zoom In

View the graph in [DOT](#) format or as a [PDF](#) pdf picture or as plain [SVG](#) data.

The above graph shows the MTS family model evolutions. Dotted edges denote "may" transitions, full edges denote "must" transitions.

Family/MTS of coffee machines explored in VMC

VMC v5.3

Commands Menu

- New Model ...
- Edit Current Model
- Explore the MTS
- Modelcheck MTS ...
- View Current Model
- View Family MTS
- Generate Products
- Welcome
- Quit

Kandinsky 1908

VMC

The Path from The Initial Configuration to Configuration C6 is:

- [C1](#) -> C2 {dollar(optional)} /* ... */
- [C2](#) -> C4 {no_sugar} /* ... */
- [C4](#) -> C6 {unsugared_cappuccino(optional)} /* ... */

The Current Configuration is **C6** ([show details ...](#))

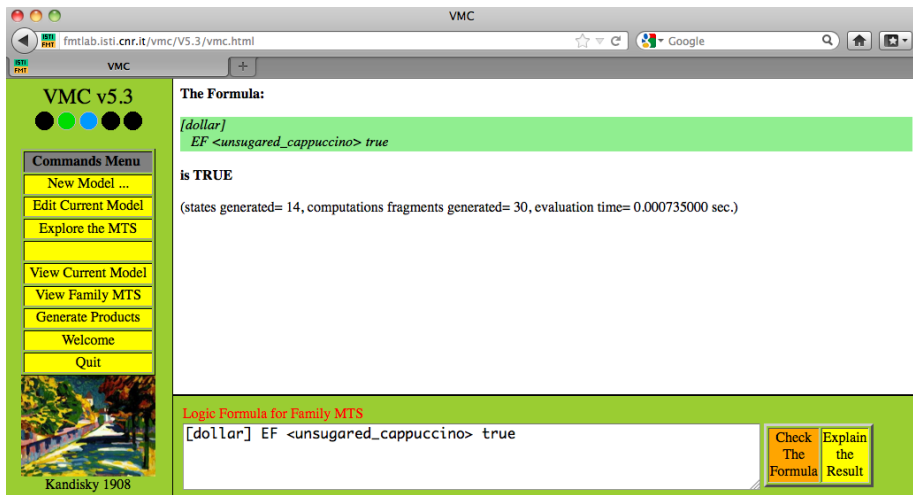
The Abstract State Labels of Configuration C6 are:
T9

The Possible Evolutions From Configuration C6 are:

- 1) [C6](#) -> [C8](#) { pour_coffee } /* ... */
- 2) [C6](#) -> [C9](#) { pour_milk } /* ... */

MTS model of coffee machine family actually permits a user to buy a cappuccino with a dollar, something which is forbidden for its products by the variability constraint `dollar EXC unsugared_cappuccino`

Outcome of a property verified over a family with VMC



The screenshot shows the VMC v5.3 web interface. The browser address bar displays `fmlab.isti.cnr.it/vmc/V5.3/vmc.html`. The page title is "VMC". On the left, there is a "Commands Menu" with buttons for "New Model ...", "Edit Current Model", "Explore the MTS", "View Current Model", "View Family MTS", "Generate Products", "Welcome", and "Quit". Below the menu is a small image of a landscape painting titled "Kandisky 1908".

The main content area displays the following information:

The Formula:

```
[dollar]  
EF <unsugared_cappuccino> true
```

is TRUE

(states generated= 14, computations fragments generated= 30, evaluation time= 0.000735000 sec.)

At the bottom, there is a section titled "Logic Formula for Family MTS" with a text input field containing the formula `[dollar] EF <unsugared_cappuccino> true`. To the right of the input field are two buttons: "Check The Formula" and "Explain the Result".

The formula expresses that every path through the MTS that starts with the insertion of a dollar, eventually leads to an unsugared cappuccino

Outcome of a property explained by VMC

The screenshot shows the VMC v5.3 web interface. On the left is a sidebar with a 'Commands Menu' containing options like 'New Model ...', 'Edit Current Model', 'Explore the MTS', 'View Current Model', 'View Family MTS', 'Generate Products', 'Welcome', and 'Quit'. Below the menu is a small image of a bridge and the text 'Kandisky 1908'. The main content area displays the results of a logic formula check. It shows three instances of a formula being checked and found to be 'FOUND_TRUE' in different states (C1, C2, C4). Each instance includes a 'This happens because' section with transition rules and a 'Logic Formula for Family MTS' section with a text input field containing the formula $[dollar] EF \langle unsugared_cappuccino \rangle true$. At the bottom right, there are two buttons: 'Check The Formula' and 'Explain the Result'.

VMC v5.3

Commands Menu

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

Kandisky 1908

⊗ The formula:
 $[dollar]$
 $EF \langle unsugared_cappuccino \rangle true$
is **FOUND_TRUE** in State C1

This happens because
 $C1 \rightarrow C2 \{euro(optional)\}$
 $C1 \rightarrow C2 \{dollar(optional)\}$
And the evolutions which satisfy the action formula $dollar$
also satisfy the subformula:
 $EF \langle unsugared_cappuccino \rangle true$

In particular:
In state C2 the subformula:
 $EF \langle unsugared_cappuccino \rangle true$ **Is Satisfied** .

⊗ The formula:
 $EF \langle unsugared_cappuccino \rangle true$
is **FOUND_TRUE** in State C2

This happens because
 $C2 \rightarrow C4 \{no_sugar\} /* \dots */$
and the subformula:
 $\langle unsugared_cappuccino \rangle true$
is **Satisfied** in State C4

⊗ The formula:
 $\langle unsugared_cappuccino \rangle true$
is **FOUND_TRUE** in State C4

This happens because
 $C4 \rightarrow C11 \{unsugared_cappuccino(optional)\}$
the transition label satisfies the action expression $unsugared_cappuccino$
and in State C11 the subformula:
 $true$ **Is Satisfied** .

Logic Formula for Family MTS
 $[dollar] EF \langle unsugared_cappuccino \rangle true$

Check The Formula Explain the Result

Products of family of coffee machines derived by VMC



The screenshot shows a web browser window with the URL `fmlab.isti.cnr.it/vmc/V5.3/vmc.html`. The page title is "VMC v5.3". On the left side, there is a "Commands Menu" with buttons for "New Model ...", "Edit Current Model", "Explore the MTS", "Modelcheck Products", "View Current Model", "View Family MTS", "Generate Products", "Welcome", and "Quit". Below the menu is a small image of a bridge over a stream, labeled "Kandisky 1908". The main content area is titled "Valid Products of the Family" and lists 10 product identifiers, each followed by a list of variability constraints in blue text:

- [product101-dollar-sugared tea-unsugared tea-ring a tone](#)
- [product102-dollar-sugared tea-unsugared tea-no ring](#)
- [product11-euro-ring a tone](#)
- [product12-euro-no ring](#)
- [product20-dollar-ring a tone](#)
- [product21-dollar-no ring](#)
- [product71-euro-sugared cappuccino-unsugared cappuccino-ring a tone](#)
- [product83-euro-sugared tea-unsugared tea-ring a tone](#)
- [product84-euro-sugared tea-unsugared tea-no ring](#)
- [product95-euro-sugared cappuccino-sugared tea-unsugared cappuccino-unsugared tea-ring a tone](#)

VMC indeed generates all 10 valid products/LTSs that are derivable from the family/MTS if the set of variability constraints is considered

Outcomes of a property verified over products with VMC

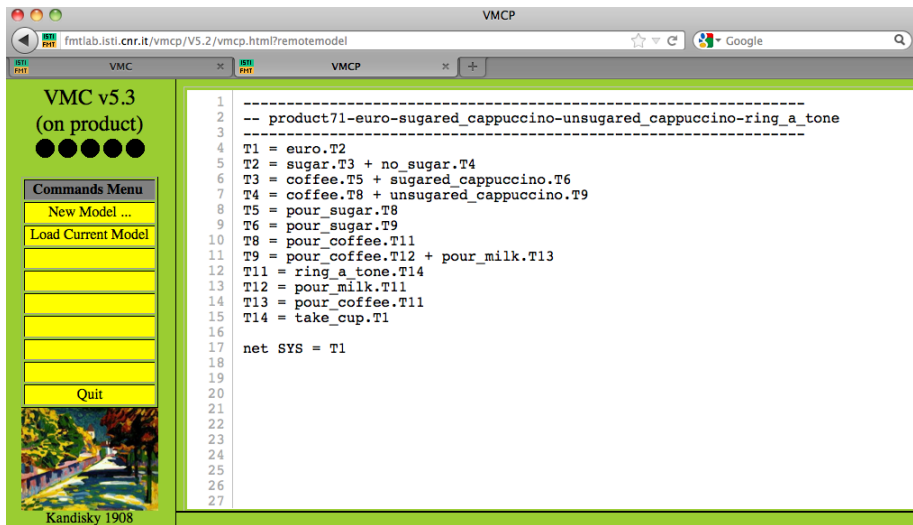
The screenshot displays the VMC web application interface. The browser address bar shows the URL `fmlab.isti.cnr.it/vmc/V5.3/vmc.html`. The page title is "VMC". The main content area is titled "Evaluation of the formula "[dollar] EF true" on all family products". Below this title, a list of product configurations is shown, each followed by the result of the formula evaluation. The results are color-coded: FALSE in red and TRUE in green.

Product Configuration	Formula evaluates
product101-dollar-sugared_tea-unsugared_tea-ring_a_tone	FALSE
product102-dollar-sugared_tea-unsugared_tea-no_ring	FALSE
product11-euro-ring_a_tone	TRUE
product12-euro-no_ring	TRUE
product20-dollar-ring_a_tone	FALSE
product21-dollar-no_ring	FALSE
product71-euro-sugared_cappuccino-unsugared_cappuccino-ring_a_tone	TRUE
product83-euro-sugared_tea-unsugared_tea-ring_a_tone	TRUE
product84-euro-sugared_tea-unsugared_tea-no_ring	TRUE
product95-euro-sugared_cappuccino-sugared_tea-unsugared_cappuccino-unsugared_tea-ring_a_tone	TRUE

At the bottom of the interface, there is a section for the "Logic Formula for all Products" with the input `[dollar] EF <unsugared_cappuccino> true`. To the right of the input field are two buttons: "Check The Formula" and "Explain the Result".

As required, no valid product (i.e. coffee machine) can deliver an (unsugared) cappuccino upon the insertion of a dollar!

Specification of one of the products derived by VMC



The screenshot shows a web browser window titled "VMCP" with the URL `fmtlab.isti.cnr.it/vmcp/V5.2/vmcp.html?remotemodel`. The interface is split into two main sections. On the left, there is a green sidebar with the text "VMC v5.3 (on product)" and five black circles. Below this is a "Commands Menu" with buttons for "New Model ...", "Load Current Model", and "Quit". At the bottom of the sidebar is a small image of a street scene with the caption "Kandisky 1908". The main area on the right displays a textual encoding of a product specification, numbered 1 through 27. The specification is enclosed in dashed lines and lists transitions T1 through T14, along with a net SYS = T1.

```
1 -----
2 -- product71-euro-sugared_cappuccino-unsugared_cappuccino-ring_a_tone
3 -----
4 T1 = euro.T2
5 T2 = sugar.T3 + no_sugar.T4
6 T3 = coffee.T5 + sugared_cappuccino.T6
7 T4 = coffee.T8 + unsugared_cappuccino.T9
8 T5 = pour_sugar.T8
9 T6 = pour_sugar.T9
10 T8 = pour_coffee.T11
11 T9 = pour_coffee.T12 + pour_milk.T13
12 T11 = ring_a_tone.T14
13 T12 = pour_milk.T11
14 T13 = pour_coffee.T11
15 T14 = take_cup.T1
16
17 net SYS = T1
18
19
20
21
22
23
24
25
26
27
```

Clicking on a product, VMC opens a window with its textual encoding

Product/LTS on previous slide visualized by VMC

VMCP

fmtilab.isti.cnr.it/vmcp/v5.2/vmcp.html?remotemodel


VMCP

VMC v5.3 (on product)

● ● ● ● ● ● ● ●

Commands Menu

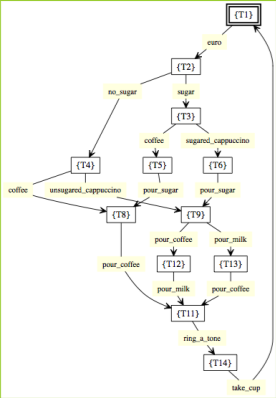
- New Model ...
- Edit Current Model
- Explore the LTS
- View Current Model
- View the LTS Graph
- Quit



Kandisky 1919

Product Model Evolutions Chart (LTS)

Zoom Out Zoom In

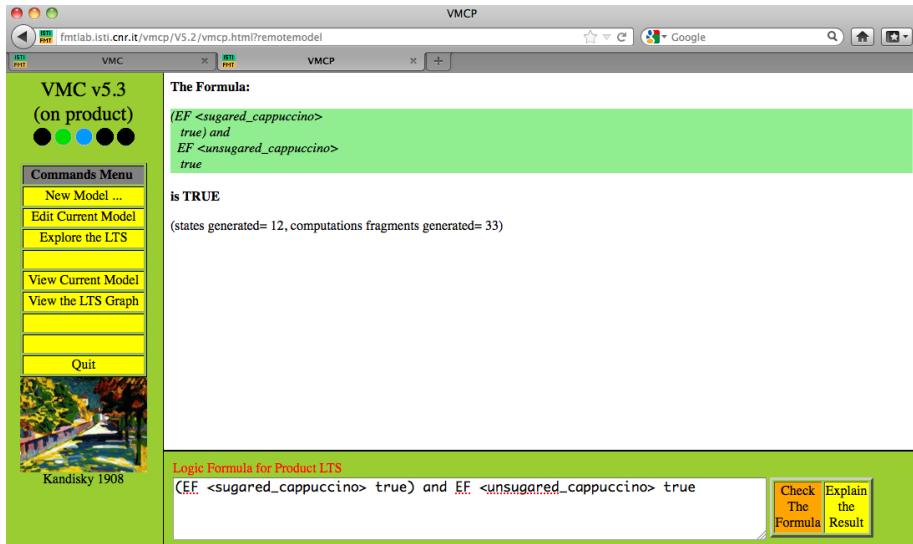


The diagram illustrates the LTS product model evolutions. It shows a sequence of states represented by nodes {T1} through {T14}. Transitions between these states are labeled with actions such as 'no_sugar', 'sugar', 'coffee', 'sugared_cappuccino', 'pour_sugar', 'pour_coffee', 'pour_milk', 'ring_a_tone', and 'take_cup'. The graph starts at state {T1} and branches out to various intermediate states before converging back to {T11} and finally reaching {T14}.

View the graph in [DOT](#) format or as a [PDF](#) pdf picture or as plain [SVG](#) data.

The above graph shows the LTS product model evolutions, which by definition contains only full edges.

Outcome of a property verified over a product with VMC



The screenshot shows a web browser window titled "VMCP" with the URL `fmltab.isti.cnr.it/vmcp/V5.2/vmcp.html?remotemodel`. The browser has two tabs: "VMC" and "VMCP".

On the left side, there is a sidebar for "VMC v5.3 (on product)" with a "Commands Menu" containing buttons for "New Model ...", "Edit Current Model", "Explore the LTS", "View Current Model", "View the LTS Graph", and "Quit". Below the menu is a small image of a bridge and the text "Kandisky 1908".

The main content area displays the following information:

- The Formula:**
 $(EF \langle \text{sugared_cappuccino} \rangle \text{true}) \text{ and } EF \langle \text{unsugared_cappuccino} \rangle \text{true}$
- is TRUE**
(states generated= 12, computations fragments generated= 33)

At the bottom, there is a section titled "Logic Formula for Product LTS" with a text input field containing the formula $(EF \langle \text{sugared_cappuccino} \rangle \text{true}) \text{ and } EF \langle \text{unsugared_cappuccino} \rangle \text{true}$. To the right of the input field are two buttons: "Check The Formula" and "Explain the Result".

The formula expresses that in this particular LTS, there exists both a path to a sugared cappuccino and a path to an unsugared cappuccino

Outcome property on previous slide explained by VMC

VMC v5.3
(on product)

Commands Menu
New Model ...
Edit Current Model
Explore the LTS
View Current Model
View the LTS Graph
Quit

Kandisky 1908

VMCP

fmlab.isti.cnr.it/vmcp/V5.2/vmcp.html?remotemodel

VMC VMCP

⊗ The formula:
(EF <sugared_cappuccino> true) and EF <unsugared_cappuccino> true
is **FOUND_TRUE** in State C1

This happens because the subformula:
EF <sugared_cappuccino> true
is **Satisfied** in State C1

And because the subformula:
EF <unsugared_cappuccino> true
is **Satisfied** in State C1

⊗ The formula:
EF <sugared_cappuccino> true
is **FOUND_TRUE** in State C1

This happens because
C1 -> C2 {euro} / ... */*
C2 -> C3 {sugar} / ... */*
and the subformula:
<sugared_cappuccino> true
is **Satisfied** in State C3

⊗ The formula:
EF <unsugared_cappuccino> true
is **FOUND_TRUE** in State C1

This happens because
C1 -> C2 {euro} / ... */*
C2 -> C4 {no_sugar} / ... */*
and the subformula:
<unsugared_cappuccino> true
is **Satisfied** in State C4

Logic Formula for Product LTS
(EF <sugared_cappuccino> true) and EF <unsugared_cappuccino> true

Check The Formula Explain the Result

Discussion and future work

VMC can also be used to specify and analyze only specific subsets of a product family's valid products by applying restrictions via constraints

Add constraint `coffee EXC dollar` to the family specification

Only European coffee machines will be derived as valid products, which can then be analyzed both as a subset and individually

Future work required before possible application in industry

- A high-level language hiding all semantic details (investigate the relation between features and actions)
- A predefined taxonomy for exemplary logical properties (e.g. the specification patterns repository for LTL, (A)CTL, etc.)
- Scale to large, industrial-size product families (with many variation points and many features)

Discussion and future work

VMC can also be used to specify and analyze only specific subsets of a product family's valid products by applying restrictions via constraints

Add constraint `coffee EXC dollar` to the family specification

Only European coffee machines will be derived as valid products, which can then be analyzed both as a subset and individually

Future work required before possible application in industry

- A high-level language hiding all semantic details
(investigate the relation between features and actions)
- A predefined taxonomy for exemplary logical properties
(e.g. the specification patterns repository for LTL, (A)CTL, etc.)
- Scale to large, industrial-size product families
(with many variation points and many features)

Discussion and future work

VMC can also be used to specify and analyze only specific subsets of a product family's valid products by applying restrictions via constraints

Add constraint `coffee EXC dollar` to the family specification

Only European coffee machines will be derived as valid products, which can then be analyzed both as a subset and individually

Future work required before possible application in industry

- A high-level language hiding all semantic details (investigate the relation between features and actions)
- A predefined taxonomy for exemplary logical properties (e.g. the specification patterns repository for LTL, (A)CTL, etc.)
- Scale to large, industrial-size product families (with many variation points and many features)

Publicity: consider submitting to VaMoS 2013 in Pisa

Variability Modelling of Software-Intensive Systems (VaMoS'13)
7th International Workshop

Pisa, Italy, January 23–25, 2013

⇒ <http://www.vamos-workshop.net>

Submission deadline: November 4, 2012

PC chairs:

- Philippe Collet (Université Nice Sophia Antipolis, France)
- Klaus Schmid (Stiftung Universität Hildesheim, Germany)

Organized by our FMT lab at the CNR research area in Pisa