INF 329
Presentation of
AUTOMATED ANALYSIS OF FEATURE MODELS 20 YEARS LATER, A LITERATURE REVIEW

By Sailaja

12.03.12

Introduction

- The automated analysis of feature models can be defined as the computer–aided extraction of information from feature models that has been studied for over two decades. The literature on feature model analysis has contributed a set of techniques, tools and empirical studies. This repository aims to provide a full coverage of the publications in the literature on the automated analysis of feature models

Feature models

- Basic feature models
- Cardinality-based feature models
- Extended feature models

Basic feature models

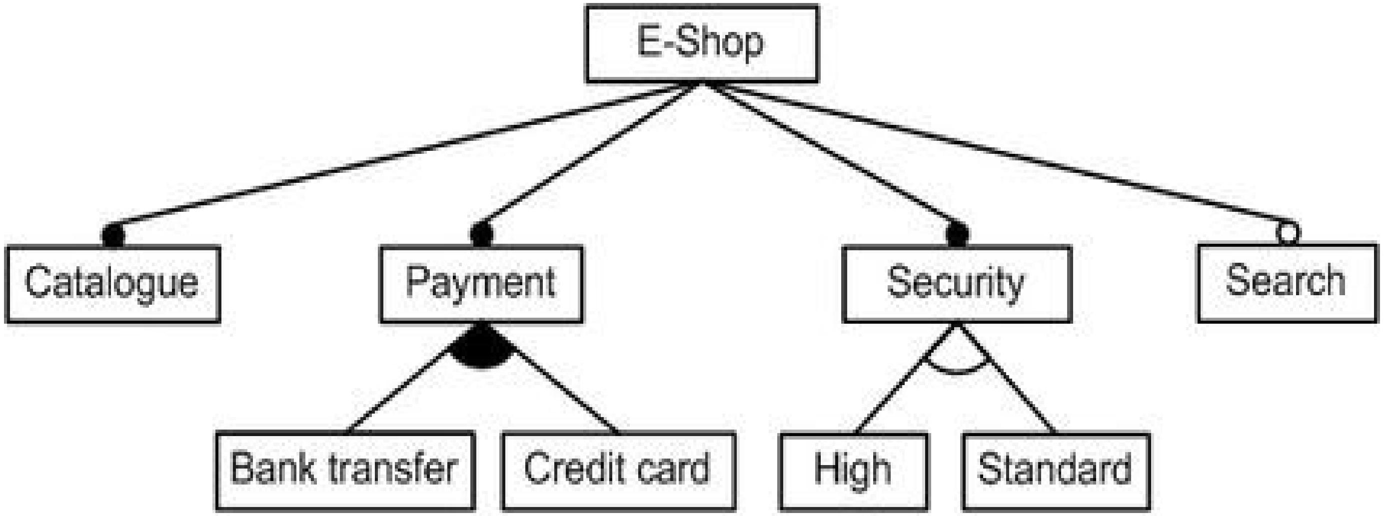
Relationships between a parent feature and its child features (or sub features) are categorized as:

- Mandatory – child feature is required.
- Optional – child feature is optional.
- Or – at least one of the sub-features must be selected.
- Alternative – one of the sub-features must be selected
- In addition to the parental relationships between features, cross-tree constraints are allowed.

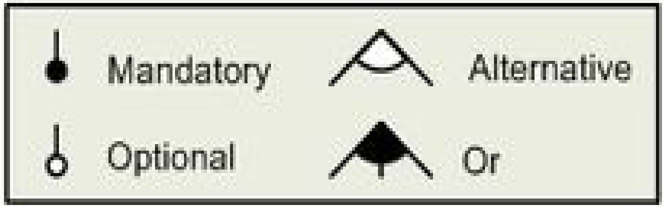
The most common are:

- A requires B – The selection of A in a product implies the selection of B.
- A excludes B – A and B cannot be part of the same product.

Example



CreditCard **implies** High



Example

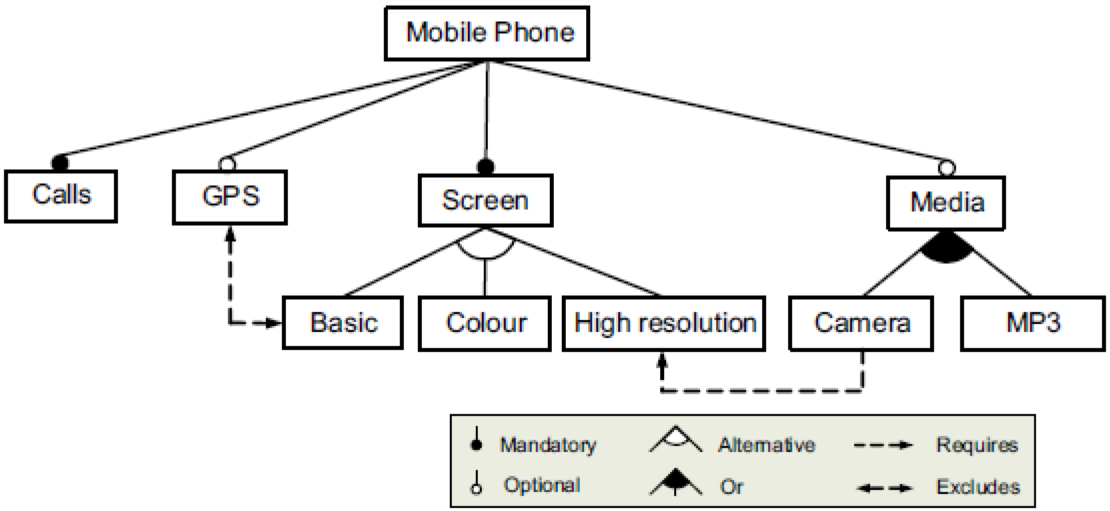


Fig. 1. A sample feature model.

Cardinality-based feature models

- Feature cardinality

  A feature cardinality is a sequence of intervals denoted [n..m] with n as lower bound and m as upper bound. These intervals determine the number of instances of the feature that can be part of a product

Group cardinality

- A group cardinality is an interval denoted <n..m> with n as lower bound and m as upper bound limiting the number of child features that can be part of a product when its parent feature is selected

Extended feature models

- Sometimes it is necessary to extend feature models to include more information about features. This information is added in terms of so-called feature attributes. This type of models where additional information is included are called extended, advanced or attributed feature models

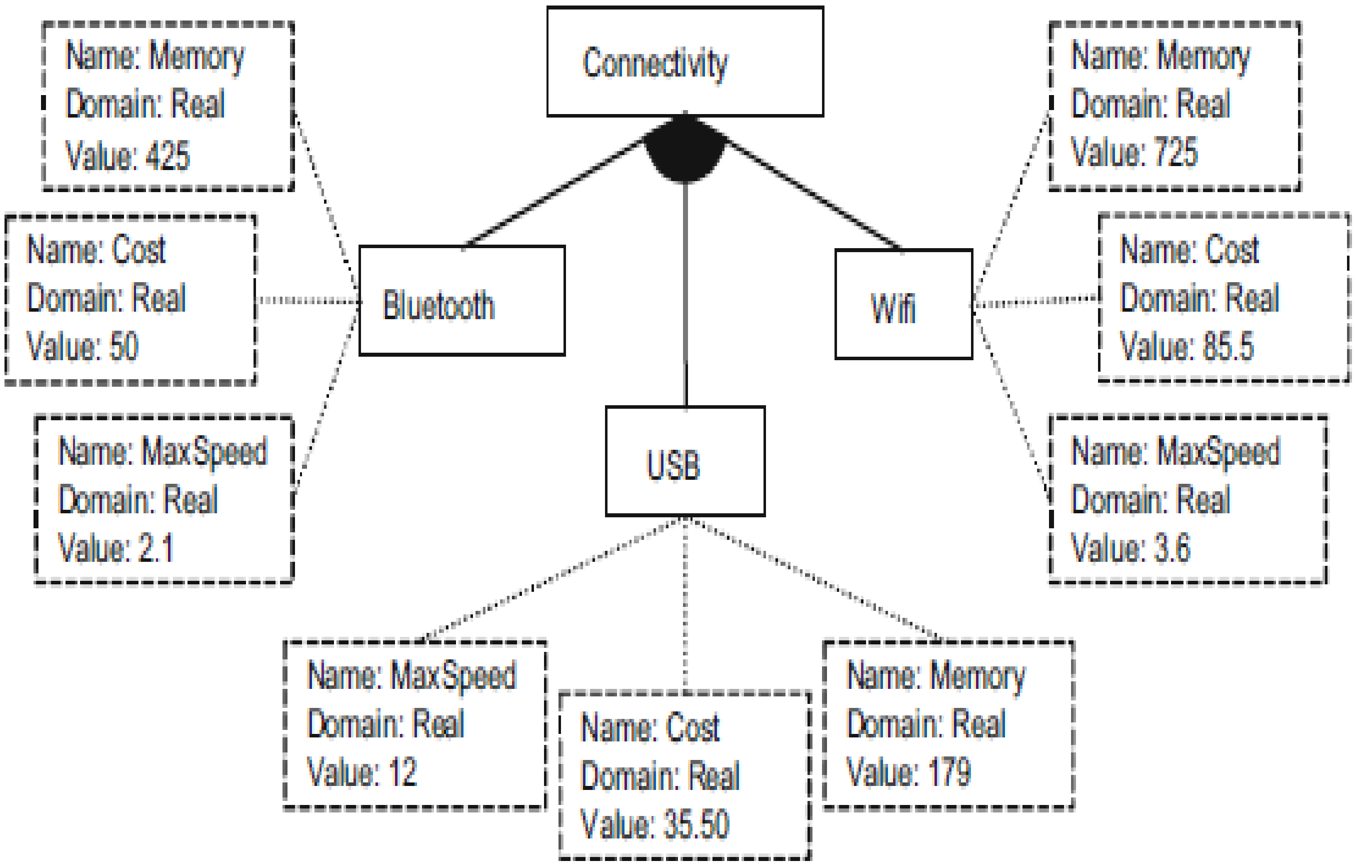


Fig. 2. A sample extended feature model.

Review method

Research questions

- What operations of analysis on feature models have been proposed?
- What kind of automated support has been proposed and how is it performed?
- What are the challenges to be faced in the future?

Inclusion and exclusion criteria

- papers proposing operations where the input feature model is modified by returning a new feature model,
- papers presenting any application of the analysis of feature models rather than proposing new analyses
- papers dealing with the analysis of other kinds of variability models

Conceptual framework

- Configuration
- Full configuration
- Partial configuration

Analysis operations on feature models

Void feature model

- A feature model is void if it represents no products.

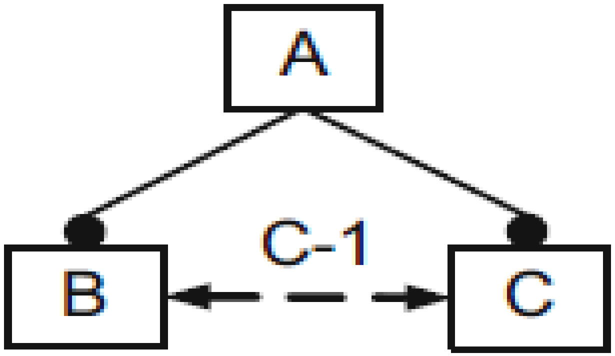


Fig. 5. A void feature model.

Valid product

- P1={MobilePhone,Screen,Colour,Media,MP3}
  P2={MobilePhone,Calls,Screen,Highresolution,GPS}
- Product P1 is not valid since it does not include the mandatory feature Calls. On the other hand, product P2 does belong to the set of products represented by the model.

Valid partial configuration

- C1=({MobilePhone,Calls,Camera}, {GPS,High resolution})
- C2=({MobilePhone,Calls,Camera}, {GPS})
- C1 is not a valid partial configuration since it selects support for the camera and removes the high resolution screen that is explicitly required by the software product line.
- C2 does not include any contradiction and therefore could still be extended to a valid full configuration

Filter

P1={MobilePhone,Calls,Screen,Highresolution,GPS}

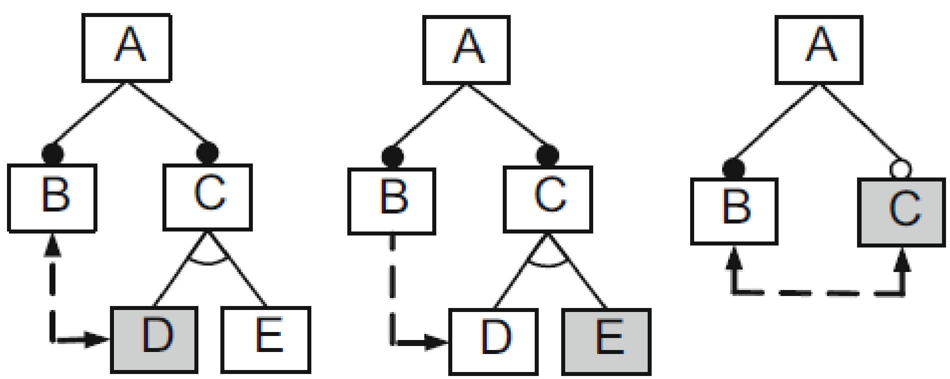
P2={MobilePhone,Calls,Screen,Highresolution,Media, MP3,GPS}

Filtering may be helpful to assist users during the configuration process .Firstly, users can filter the set of products according to their key requirements.Then ,the list of resultant products can be inspected to select the desired solution

Anomalies detection

Dead features

- A feature is dead if it cannot appear in any of the products of the software product line .Dead features are caused by a wrong usage of cross-tree constraints.



. **6.** Common cases of dead features. Grey features are dead.

Conditionally dead features

- A feature is conditionally dead if it becomes dead under certain circumstances (e.g. when selecting another feature) . Both unconditional and conditional dead features are often referred to in the literature as“contradictions” or “inconsistencies

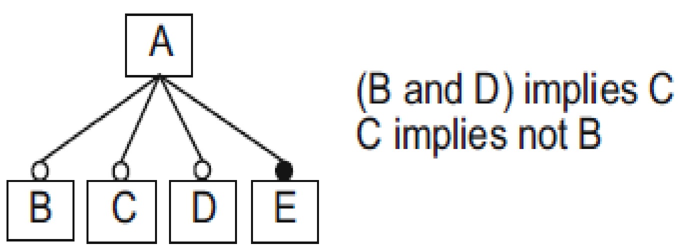


Fig. 7. An example of a conditionally dead feature.

False optional features

- A feature is false optional if it is included in all the products of the product line despite not being modeled as mandatory

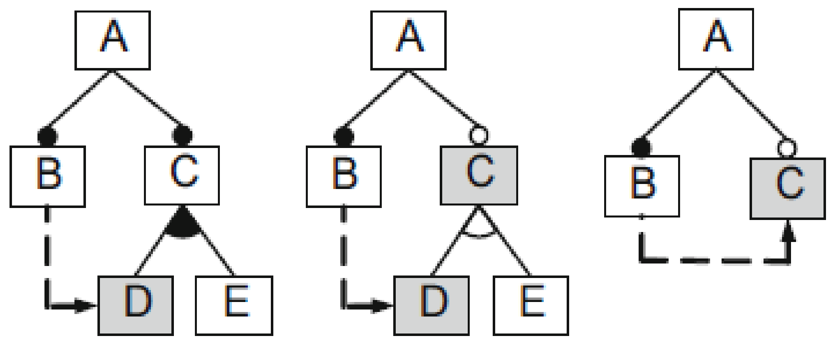


Fig. 8. Some examples of false optional features. Grey features are false optional.

Wrong cardinalities

- Group cardinality is wrong if it cannot be instantiated. These appear in cardinality- based feature models where cross-tree constraints are involved.

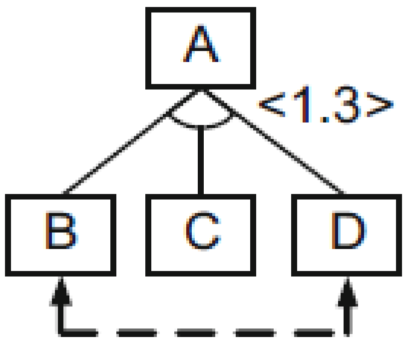


Fig. 9. An example of wrong cardinality.

Redundancies

- A feature model contains redundancies when some semantic information is modeled in multiple ways . Generally, this is regarded as a negative aspect since it may decrease the maintainability of the model

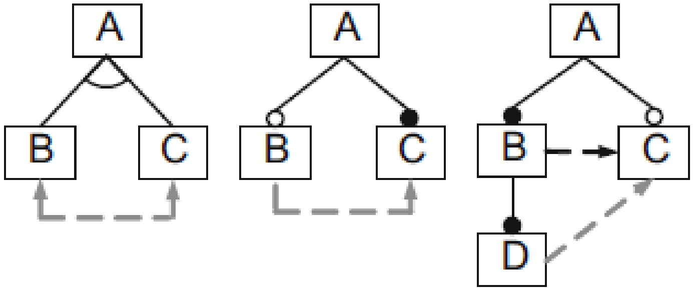
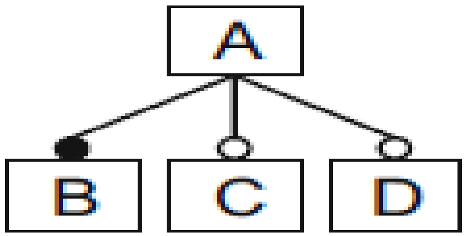


Fig. 10. Some examples of redundancies. Grey constraints are redundant.

Refactoring

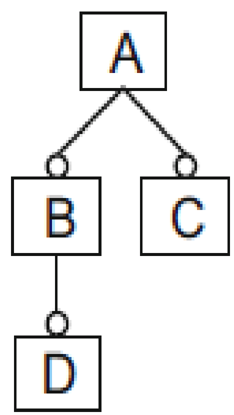
- A feature model is a refactoring of another one if they represent the same set of products while having a different structure



(b) Refactoring

Generalization

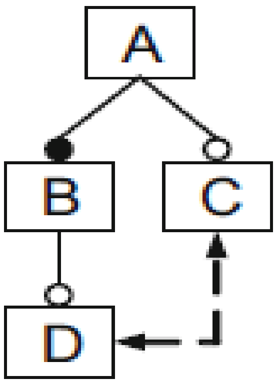
- A feature model, F, is a generalization of another one, G, if the set of products of F maintains and extends the set of products of G



(c) Generalization

Specialization

- A feature model, F, is a specialization of another one, G, if the set of products of F is a subset of the set of products of G



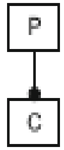
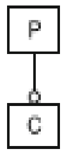
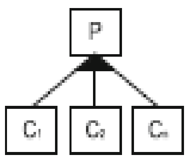
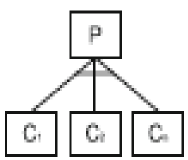
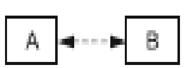
(d) Specialization

Atomic sets

An atomic set is a group of features (at least one) that can be treated as a unit when performing certain analyses .The intuitive idea behind atomic sets is that mandatory features and their parent features always appear together in products and therefore can be grouped without altering the result of certain operations.

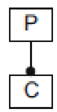
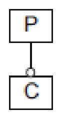
Propositional logic based analyses

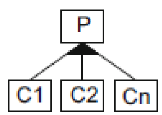
A propositional formula consists of a set of primitive symbols or variables and a set of logical connectives constraining the values of the variables

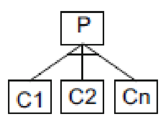
Relationship		PL Mapping	Mobile Phone Example
MANDATORY		$P \leftrightarrow C$	$MobilePhone \leftrightarrow Calls$ $MobilePhone \leftrightarrow Screen$
OPTIONAL		$C \rightarrow P$	$GPS \rightarrow MobilePhone$ $Media \rightarrow MobilePhone$
OR		$P \leftrightarrow (C_1 \vee C_2 \vee ... \vee C_n)$	$Media \leftrightarrow (Camera \vee MP3)$
ALTERNATIVE		$(C_1 \leftrightarrow (\neg C_2 \wedge ... \wedge \neg C_n \wedge P)) \wedge$ $(C_2 \leftrightarrow (\neg C_1 \wedge ... \wedge \neg C_n \wedge P)) \wedge$ $(C_n \leftrightarrow (\neg C_1 \wedge \neg C_2 \wedge ... \wedge \neg C_{n-1} \wedge P))$	$(Basic \leftrightarrow (\neg Color \wedge \neg Highresolution \wedge Screen)) \wedge$ $(Color \leftrightarrow (\neg Basic \wedge \neg Highresolution \wedge Screen)) \wedge$ $(Highresolution \leftrightarrow (\neg Basic \wedge \neg Color \wedge Screen))$
IMPLIES		$A \rightarrow B$	$Camera \rightarrow Highresolution$
EXCLUDES		$\neg(A \wedge B)$	$\neg(GPS \wedge Basic)$

Constraint programming based analyses

- A constraint satisfaction problem (CSP) consists of a set of variables, a set of finite domains for those variables and a set of constraints restricting the values of the variables.

Relationship		CSP Mapping	Mobile Phone Example
MANDATORY		P = C	Mobilephone = Calls Mobilephone = Screen
OPTIONAL		if (P = 0) C = 0	if (Mobilephone = 0) GPS = 0 if (Mobilephone = 0) Media = 0

OR		if (P > 0)
		Sum (C1, C2,...Cn) in {1..n}
		else
		C1 = 0, C2 = 0,...., Cn = 0
			if (Media > 0)
			Sum (Camera, MP3) in {1..2}
			else
			Camera = 0, MP3 = 0

ALTERNATIVE		if (P > 0)
		Sum (C1 ,C2,...Cn) in {1..1}
		else
		C1 = 0, C2 = 0,...., Cn = 0
			if (Screen > 0)
			Sum (Basic, Colour, High resolution) in {1..1}
			else
			Basic = 0, Colour = 0, High resolution = 0

REQUIRES		if (A > 0) B>0	if (Camera > 0) High resolution > 0
EXCLUDES		if (A > 0) B = 0	if (GPS > 0) Basic = 0

Fig. 16. Mapping from feature model to CSP.

Discussions and challenges

- RQ 3: What are the challenges to be faced in the future?
- Challenge 1: Formally describe all the operations of analysis and provide a formal framework for defining new operations.
- Challenge 2: Include feature attribute relationships for analyses on feature models and propose new operations of analysis leveraging extended feature models
- Challenge 3: Further studies about computational complexity of analysis.
- Challenge 4: Develop standard benchmarks for analysis operations.
- Challenge 5: Study how propositional logic and description logic-based solvers can be used to add attributes on feature models.
- Challenge 6: Compare in depth description logic- based solvers with respect to analysis operations and other solvers.
- Challenge 7: Characterize feature models, analysis operations and solvers to select the best choice in each case

Conclusions

The analysis of feature models is maturing with an increasing number of contributions, operations, tools and empirical works. We also identified a number of challenges for future research mainly related to the formalization and computational complexity of the operations, performance comparison of the approaches and the support of extended feature models