Feature Interactions:
the Good, the Bad, and the Ugly

Jo Atlee • U Waterloo • Dec 2013

Sandy Beidu
Shoham Ben-David
Cecylia Bocovich
Jonathan Hay
Pourya Shaker

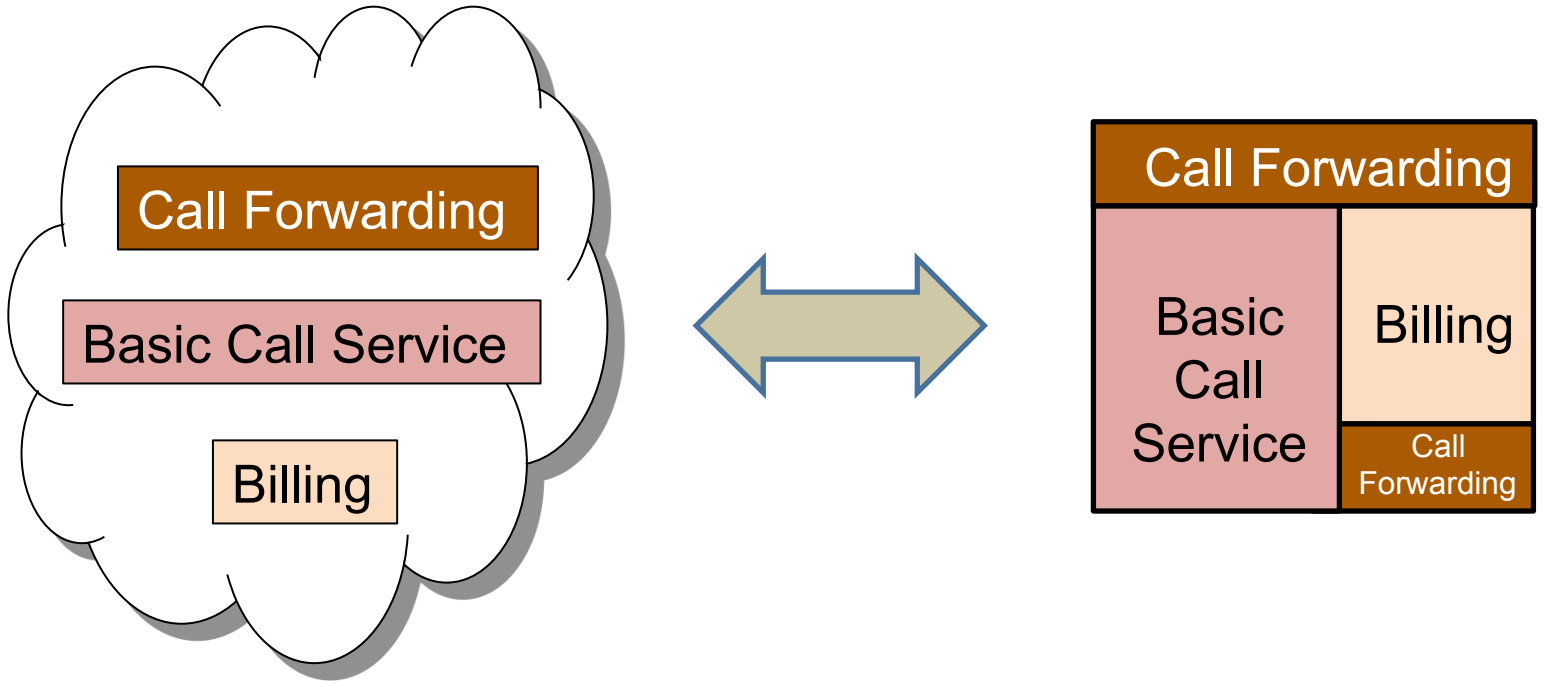
WATFORM

David R. Cheriton School of Computer Science
University of Waterloo

nesis

feature-oriented software development

feature : **a unit of *functionality* or *added value* in the product**

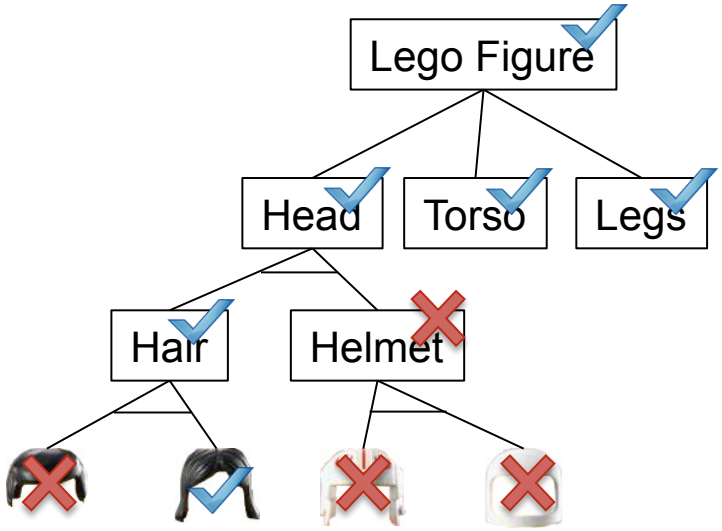


stakeholders' mental model of system

feature-oriented software system

product lines

feature model = valid configurations



mapping

reusable implementations

products

[Example from Sven Apel]

product lines

feature interactions

feature interaction: features influence each other in defining overall system behaviour [Zave]

- conflicts over shared context
- violations of global correctness properties
- emergent behaviours

feature interaction problem: the number of potential interactions is exponential in the number of features

what this talk is about

modelling feature requirements

- feature modularity
- modelling intended interactions

analyzing feature combinations

- to detect interactions

resolution strategies

- strategies that avoid classes of interactions

the good

not all interactions are bad!

intended interactions

- advanced cruise-control variants override basic cruise control
- prohibit navigation overrides navigation
- prohibit-navigation override overrides prohibit-navigation

unintended but harmless interactions

- call screening prevents activation of caller id

(planned) resolutions to conflicts

- brake override overrides (acceleration $\oplus$ braking)

feature-oriented requirements modelling language (FORML)

Shaker, Atlee, Wang, RE'12

a notation for modelling the requirements of a product line (PL)

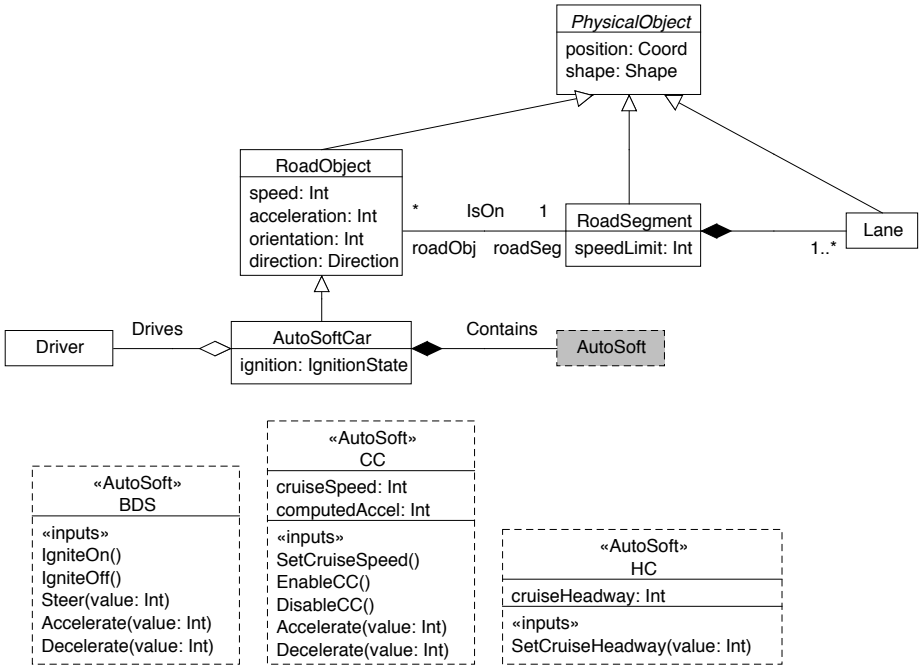
- supports feature modularity
- provides language constructs for expressing intended interactions explicitly
- composes features into a product line

req models decomposed by feature

world model

a conceptual model of the problem world

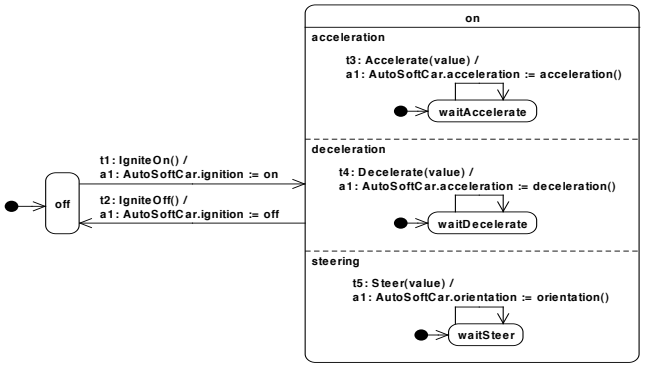
- defines possible world states
- including feature phenomena



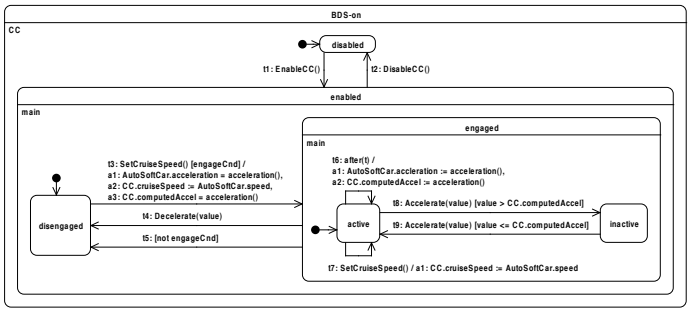
behaviour model

state-machine models (of features)

- whose events, conditions and actions are expressions over world phenomena
- and over feature phenomena



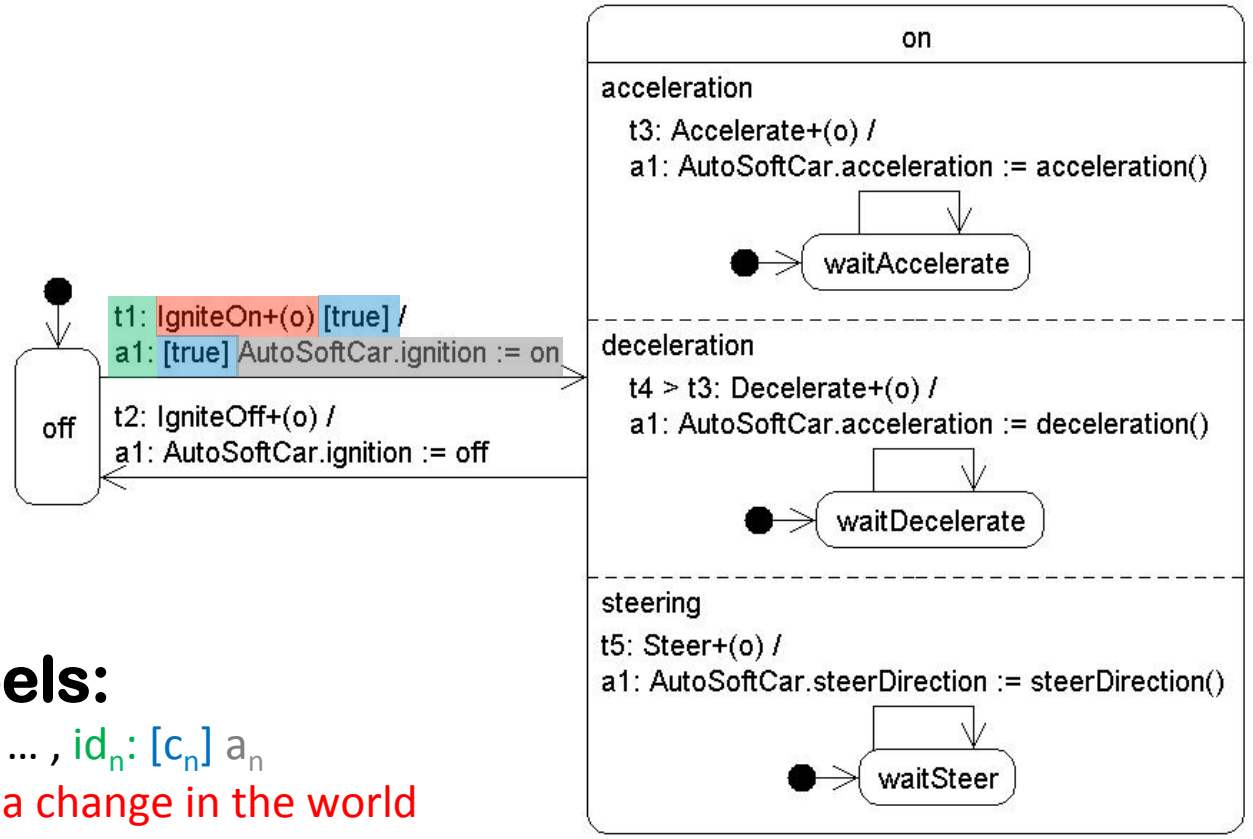
feature



feature

modelling features

features are modelled as hierarchical state machines that sense and control the world



transition labels:

id: e [c] / id_1: [c_1] a_1, … , id_n: [c_n] a_n

- triggering event: a change in the world
- guard condition: predicate over the world
- action: a prescribed change to the world
- transition or action name

a new feature may…

introduce behaviours
- **via:** new machines

eliminate behaviours
- **via:** new or stronger enabling conditions on existing actions or transitions

substitute behaviours
- **via:** new *pre-empting* actions or transitions

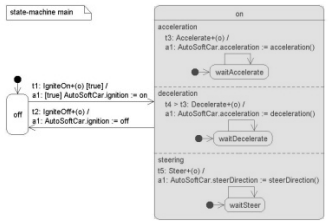
can also be expressed as extensions to existing features:
new regions, new states,
new transitions,
weakened enabling conditions

intended interactions:
modelled as structural extensions at extension points in existing features

adding behaviours

BDS

Cruise Control (CC)



extends BDS state

new region

state-machine extension

transition BDS{t3}: [strengthen with c: not inState(main.enabled.main.engaged.main.active) or driverOverride()]

BDS{main.on}

main

enabled

main

engaged

main

t6: after(t()) /
a1: AutoSoftCar.accleration := acceleration(),
a2: CC.goalAccel := acceleration()

disabled

t1: EnableCC+(o)

t2: DisableCC+(o)

disengaged

t3: SetCruiseSpeed+(o) [engageCnd()] /
a1: CC.cruiseSpeed := AutoSoftCar.speed,
a2: CC.goalAccel = 0

t4: Decelerate+(o)

t5: [not engageCnd()]

active

t8: Accelerate+(o) [driverOverride()]

t9: Accelerate+(o) [not driverOverride()]

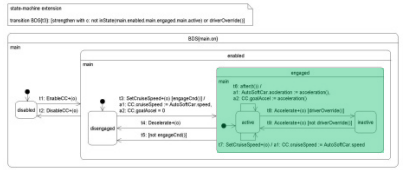
inactive

t7: SetCruiseSpeed+(o) / a1: CC.cruiseSpeed := AutoSoftCar.speed

replacing behaviours

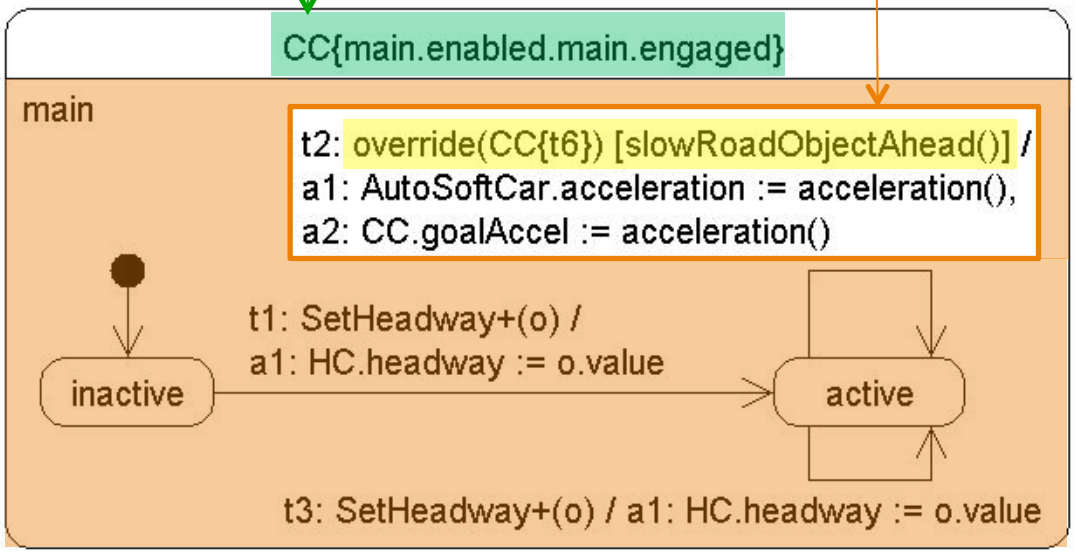
Headway Control (HC)

CC



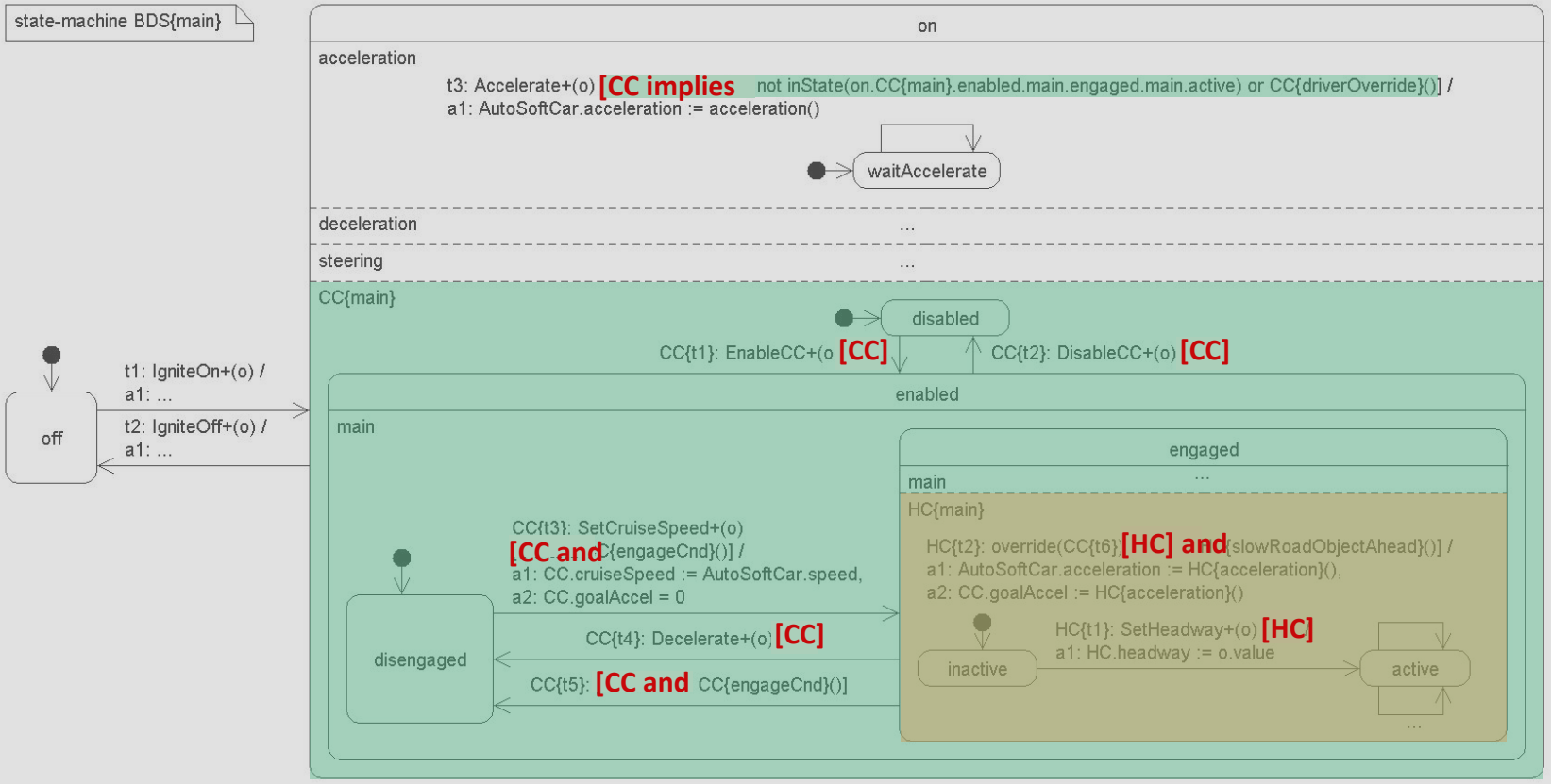
extends CC state

new region includes pre-empting transition:
models HC intentionally prohibiting CC



composition is a product line

transitions, actions, clauses are guarded by presence conditions (of their declaring feature)



product line = {BDS, BDS + CC, BDS + CC + HC}

summary of FORML

- **precise modular modelling of features**
- **new features extend existing features**
  - with added, removed, and replaced behaviours
- **explicit modelling of intended feature interactions**
- **result of feature composition is a product line**

the bad

hybrid brakes ⊕ anti-lock breaking

2010 Toyota Prius

hybrid brake system

- (normal) hydraulic brake system
- regenerative braking system
  - converts loss of vehicle momentum into electrical energy
  - stored in on-board batteries

anti-lock brake system (ABS)

- maintains stability, steerability during panic braking

interaction

- braking force after ABS actuation reduced
- vehicle stopping distance is increased
- 62 reported crashes, 12 injuries

cruise control ⊕ traction control

cruise control

- vehicle set to maintain driver-specified speed

traction control

- brake fluid applied when wheels slip

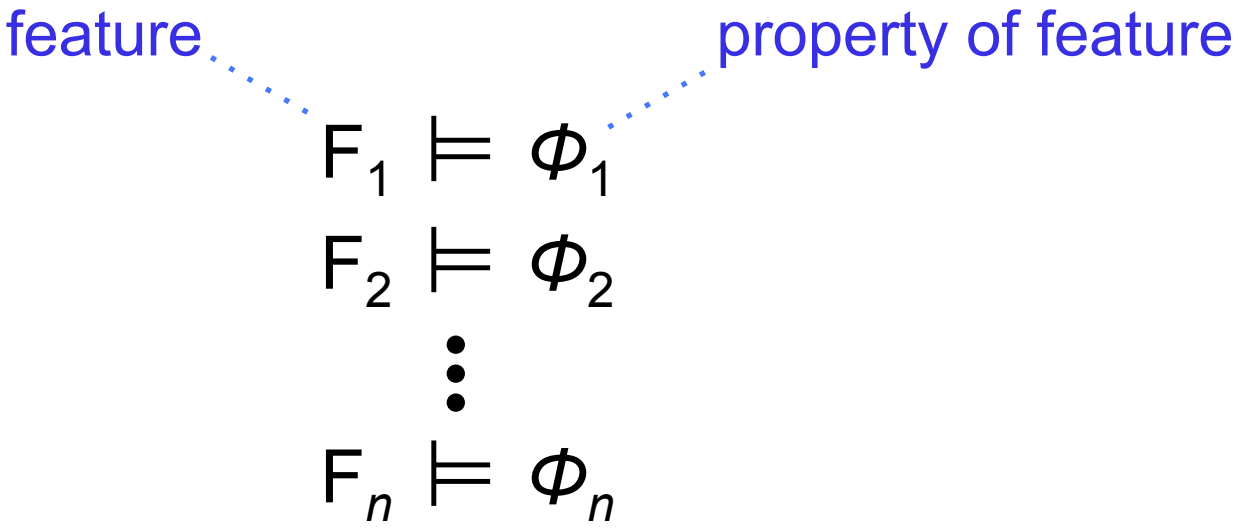
interaction

- engine power is increased (to maintain speed)
- driver senses “sudden acceleration”
  - vehicle becomes difficult to control

resolution

- advise drivers not to use cruise control on slippery roads

feature interaction

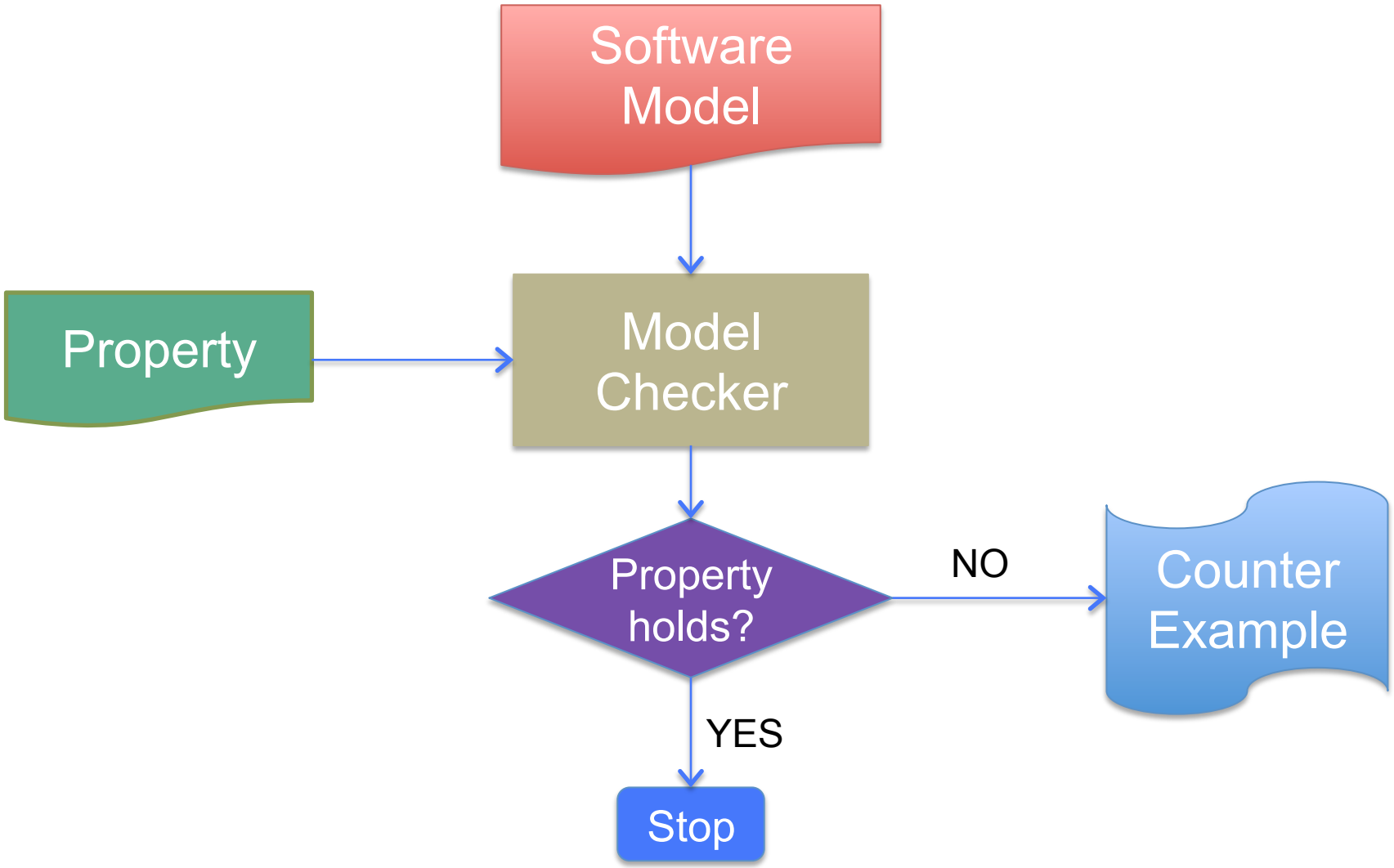


$$F_1 \oplus F_2 \oplus \cdots \oplus F_n \not\models \Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_n$$

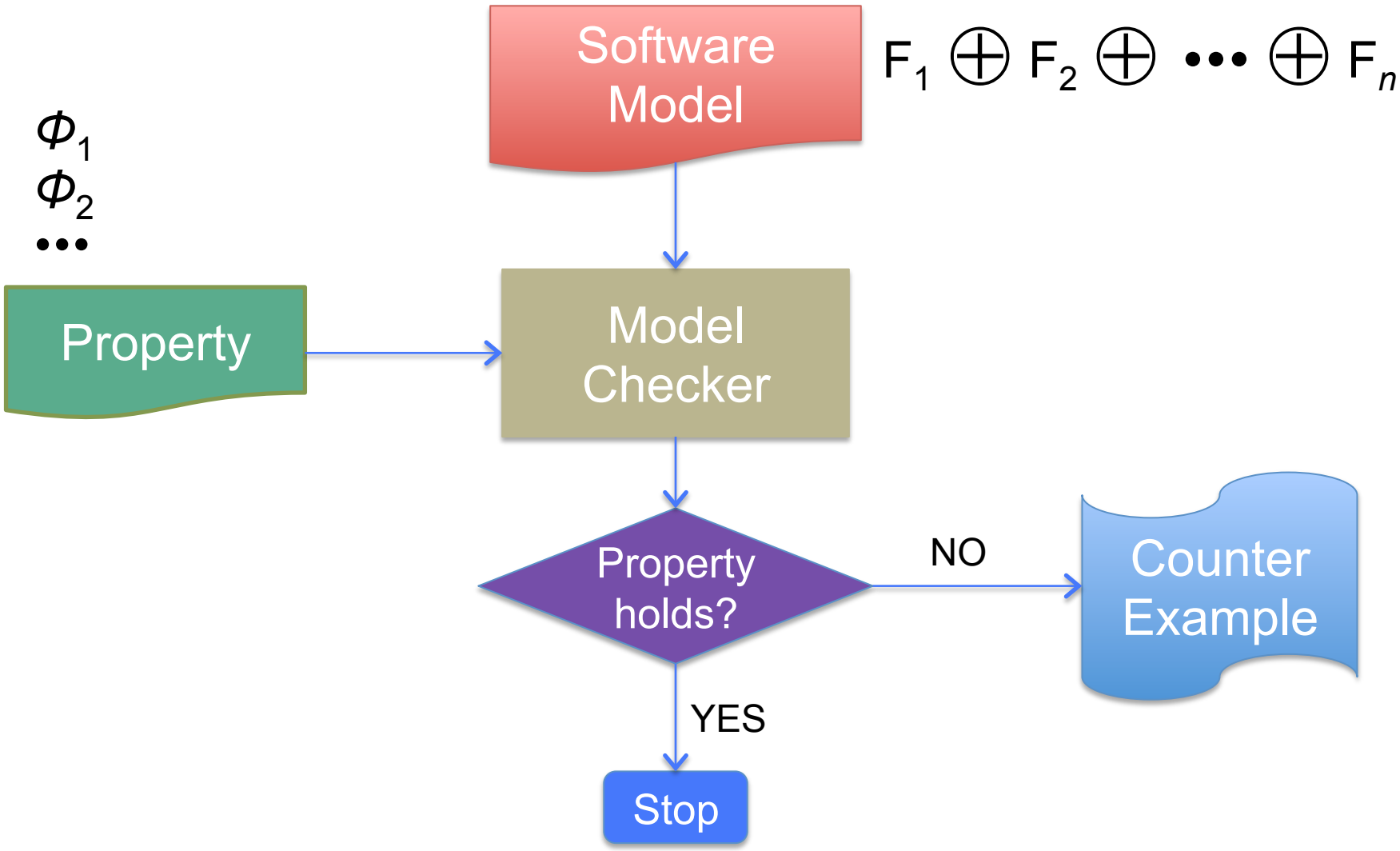
feature composition (= product)

model checking

Clarke, Emerson '81, Queille, Sifakis '82

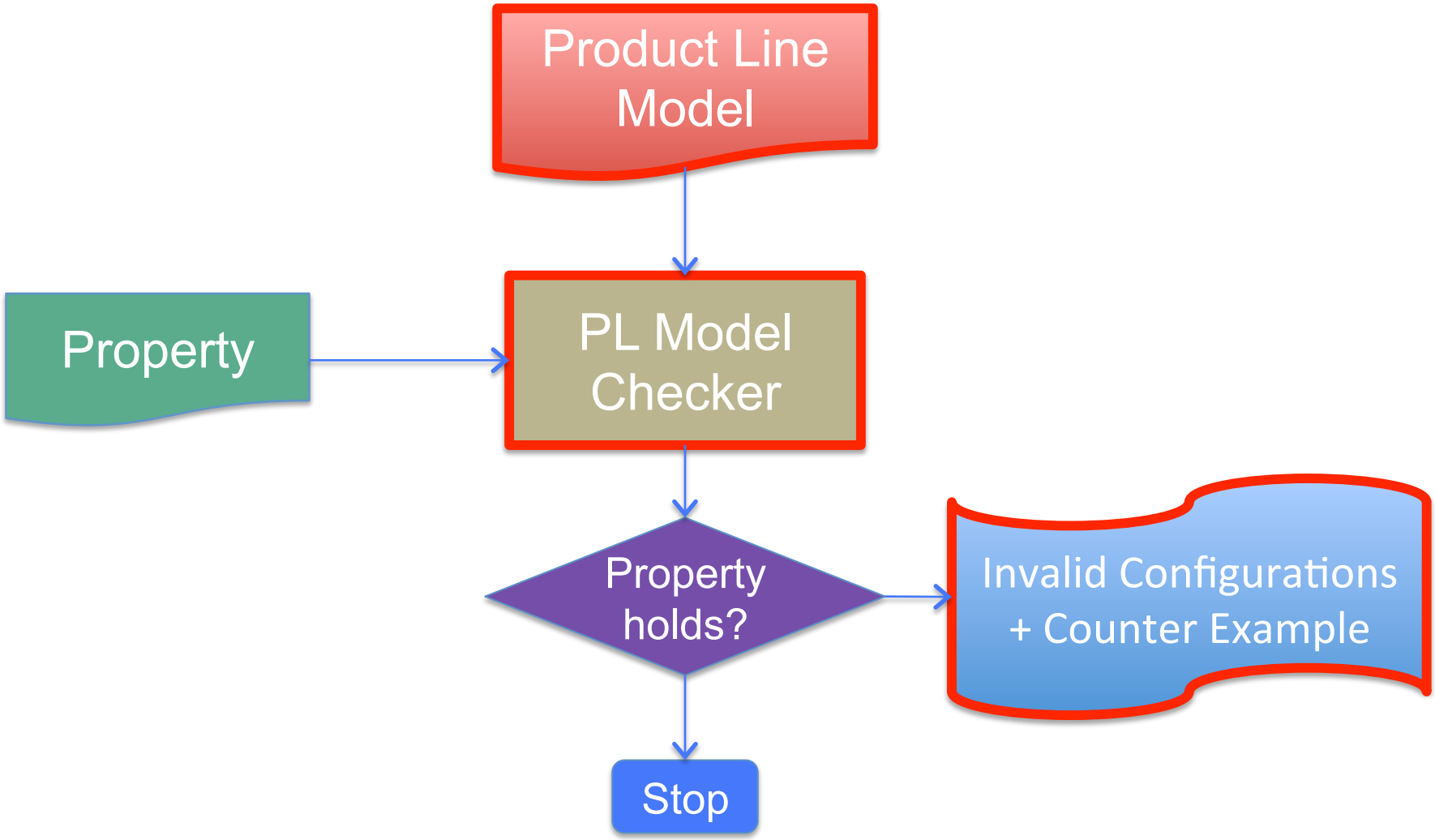


detecting feature interactions



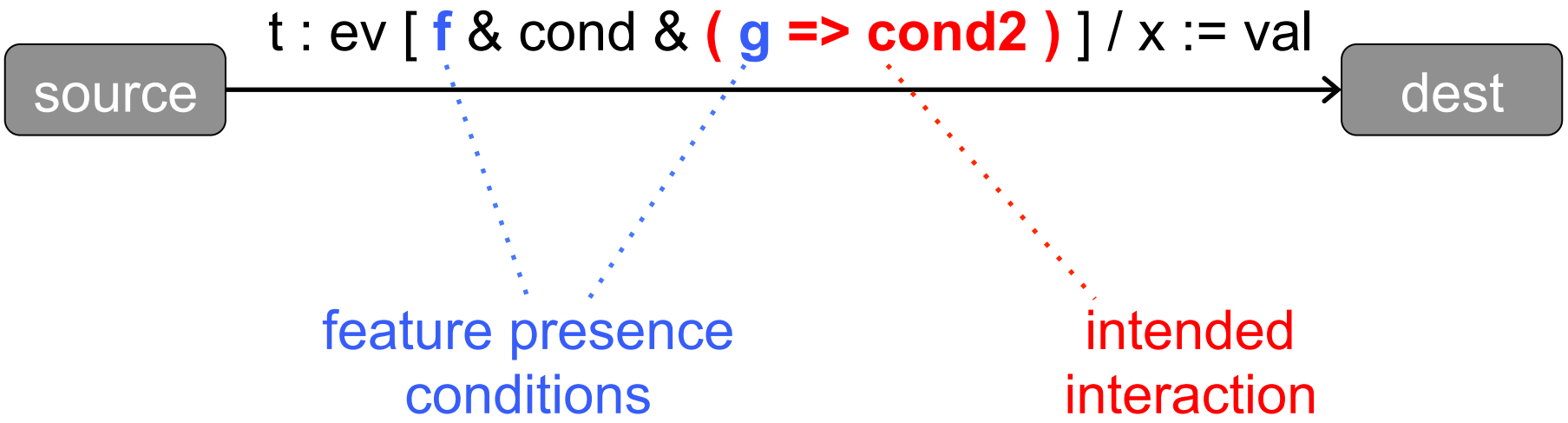
product-line model checking

Classen, Heymans, Schobbens, Legay, Raskin, ICSE'10



properties should…

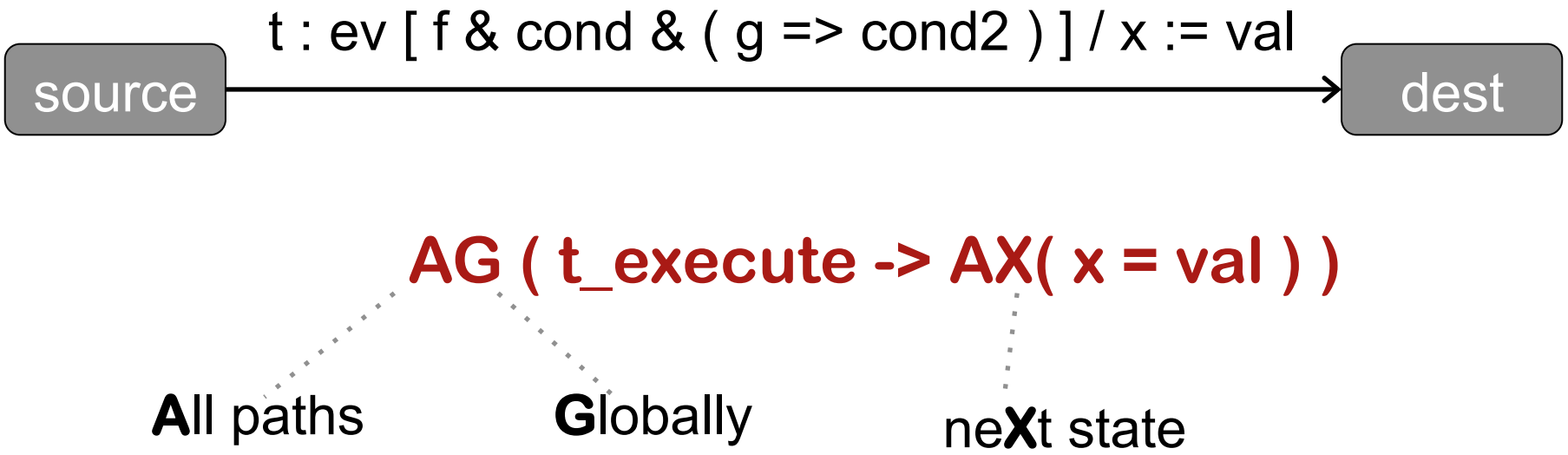
- **reflect each feature's desired behaviour**
- **be conditional on whether a feature is present**
- **accommodate intended interactions**
  - which affect whether a transition executes



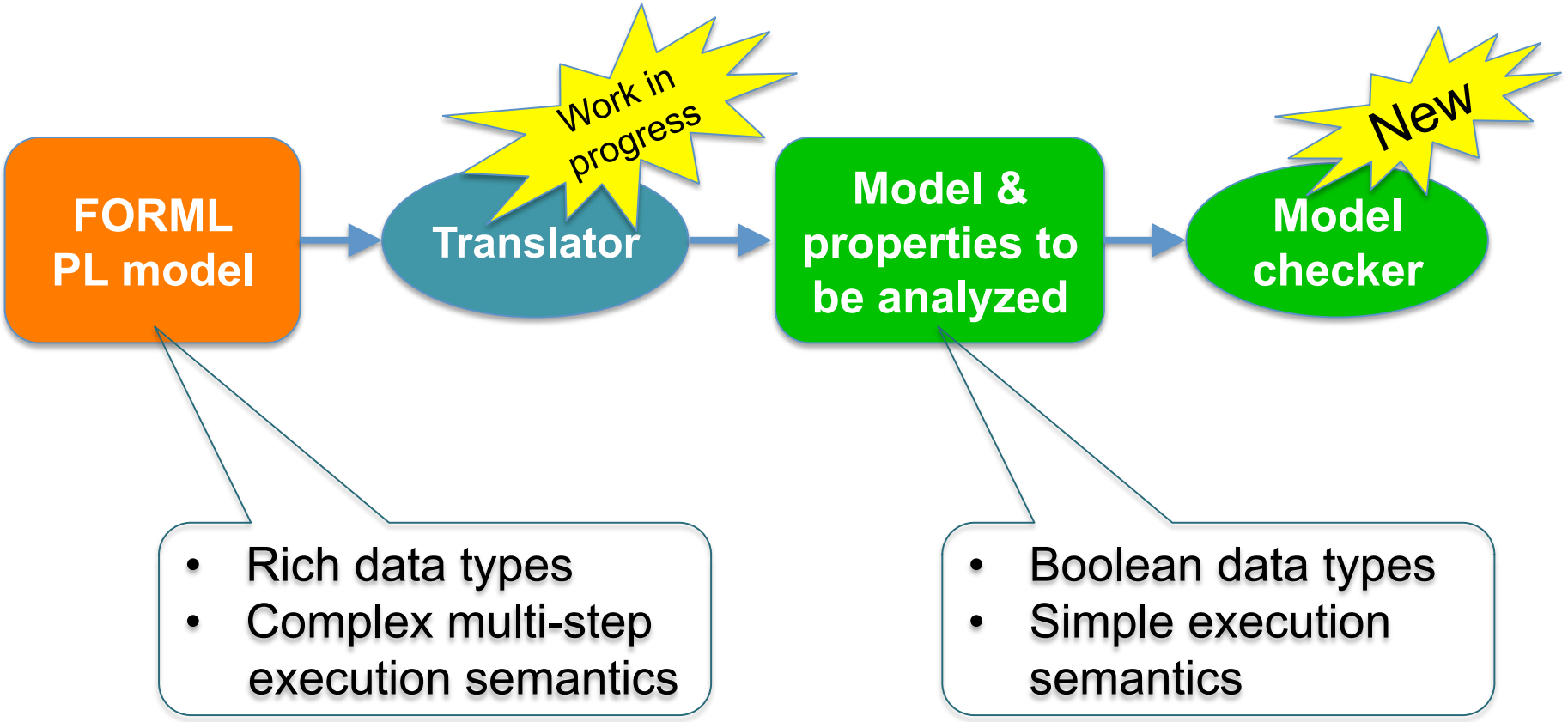
properties

a property for each transition in the PL model:

- if transition executes, the effects of its actions are realized
- can be generated automatically from PL model



progress


FORML
PL model
Work in progress
Translator
Model & properties to be analyzed
New
Model checker
• Rich data types
• Complex multi-step execution semantics
• Boolean data types
• Simple execution semantics

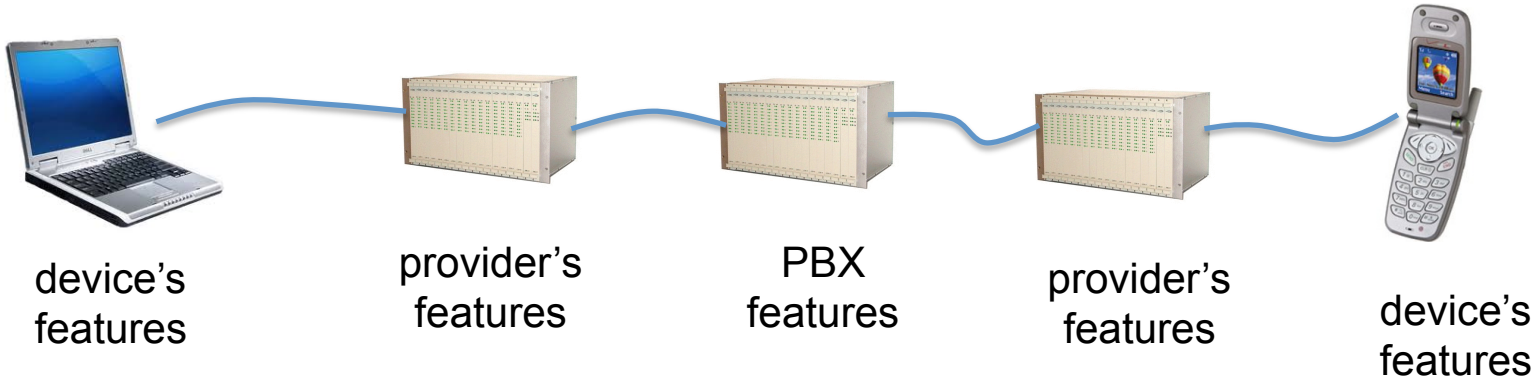
summary of interaction detection

- **properties can be generated automatically from the PL model**
- **analyzer checks every property in all behaviours of all products in product line**
- **analyzer identifies, for each property, all products in which the property can be violated**
- **only unintended interactions will be reported**

the ugly: scalability

lots of features

e.g., telephony has 1000+ features per system

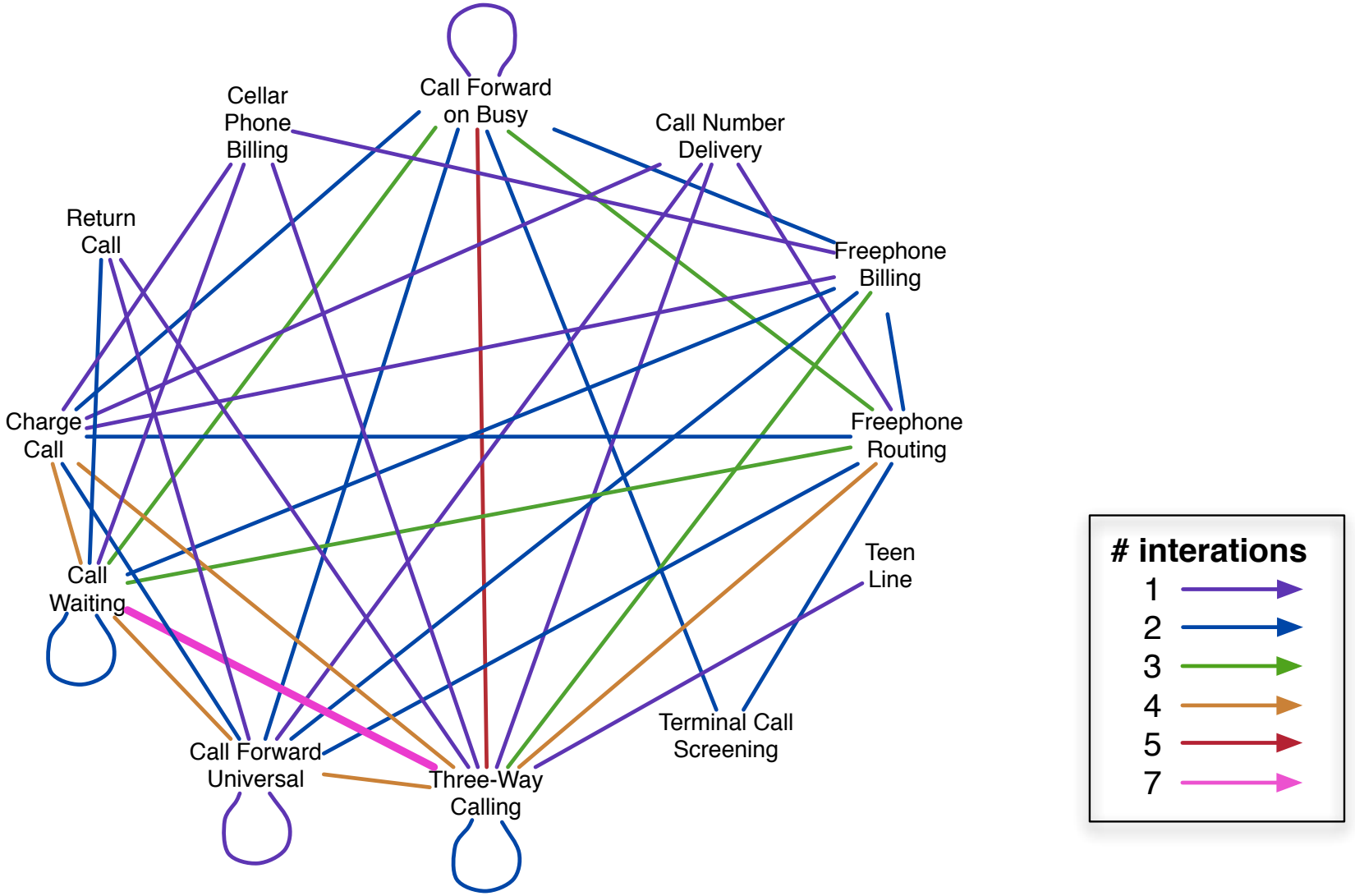


a system of feature-rich systems

- features from multiple providers
- multiple active versions of the same feature

lots of interactions

results of the second feature interaction contest



lots of types of interactions

control-flow
one feature affects the flow of control in another feature

data-flow
one feature affects (deletes, alters) a message destined for another feature

data modification
shared data read by one feature is modified by another feature

data conflict
two features modify the same data

control conflicts
two features issue conflicting actions

assertion violation
one feature violates another feature's assertions or invariants

resource contention
the supply of resources is inadequate, given the set of competing features

introduced in several phases

Bowen, SETSS'89

[req] understanding / specifying how features ought to interact

[req] the number of interactions (and resolutions) to consider grows exponentially with the number of features

[design] more interactions introduced during design due to sharing of resources, I/O devices, protocol signals, etc.

[imp] near-commonalities among features leads to questions about how to effectively reuse software components

[test] the sheer number of interactions and resolutions to be tested lengthens the testing phase

wicked problem

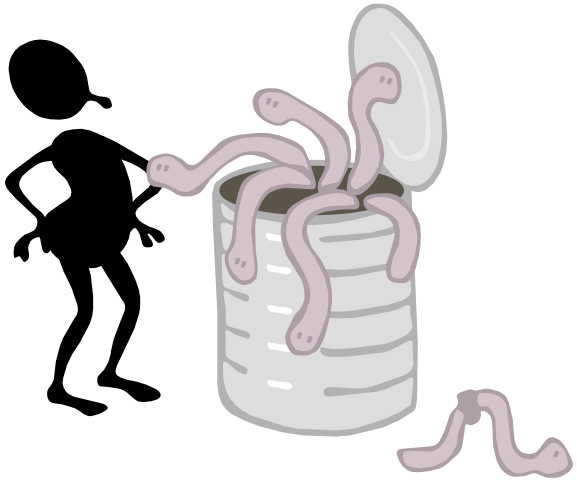
lots of features

lots of interactions

multiple types of interaction

lots of resolutions

introduced in several phases



resolve interactions through feature composition

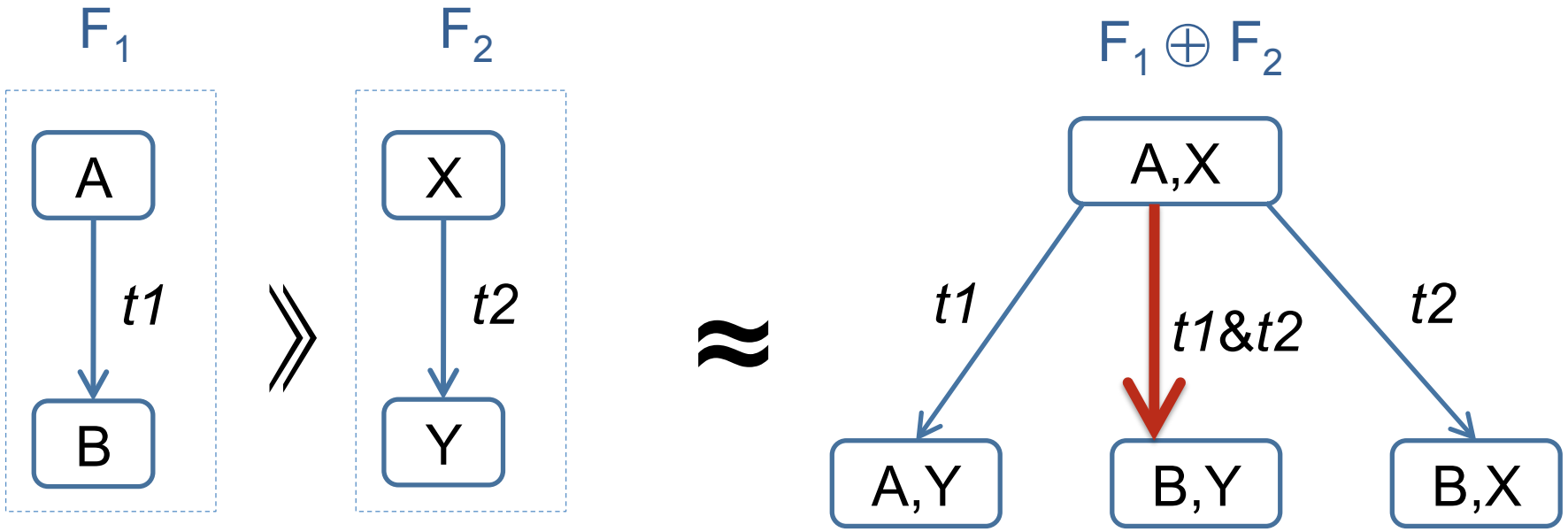
- compose features into products (or product lines)
- composition algorithm resolves entire classes of interactions

conflict-free composition

Hay, Atlee, FSE'00

resolution strategy: maximal subset of enabled transitions with *nonconflicting actions*

› uses feature priority to resolve conflicts

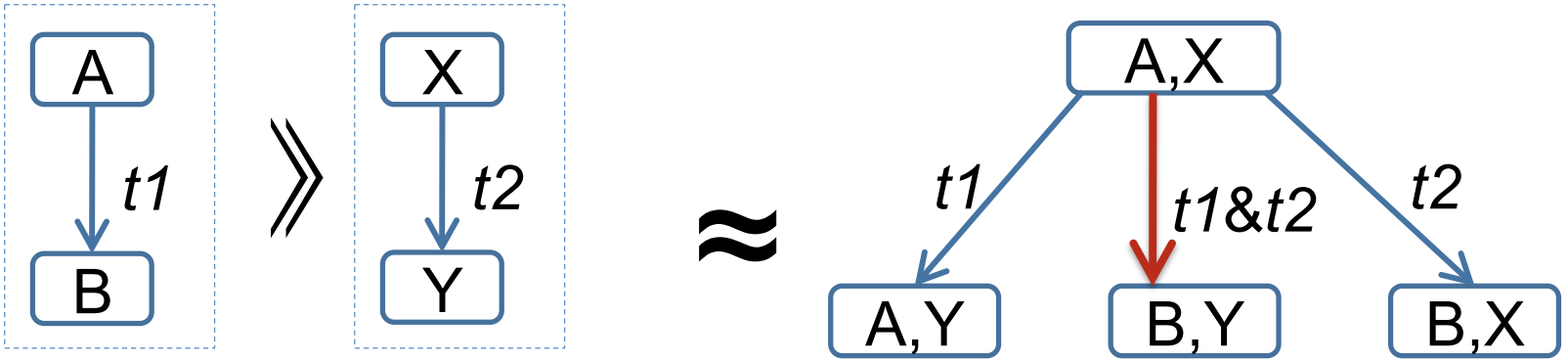


t1&t2 ≫ *t1* ≫ *t2*

violation-free composition

Hay, Atlee, FSE'00

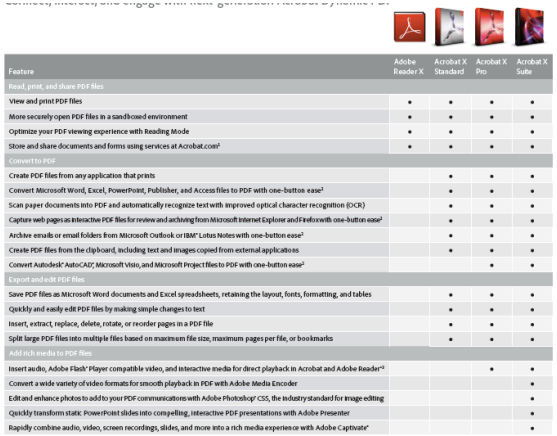
resolution strategy: maximal subset of enabled transitions with *nonconflicting and nonviolating* actions



4 classes of interactions

	resolution
› actions conflict	resolve by priority
› actions violate assertions	resolve by priority
› new assertions not satisfied	apply transition
› new assertions conflict	apply transition

feature coordination



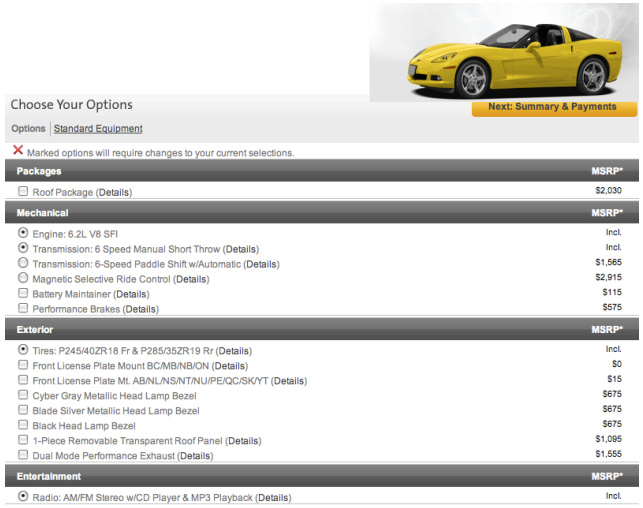
Feature	Adobe Reader X	Acrobat X Standard	Acrobat X Pro	Acrobat X Suite
Read, print, and share PDF files				
View and print PDF files	•	•	•	•
More securely open PDF files in a sandboxed environment	•	•	•	•
Optimize your PDF viewing experience with Reading Mode	•	•	•	•
Store and share documents and forms using services at Acrobat.com[2]	•	•	•	•
Convert to PDF				
Create PDF files from any application that prints		•	•	•
Convert Microsoft Word, Excel, PowerPoint, Publisher, and Access files to PDF with one-button ease[3]		•	•	•
Scan paper documents into PDF and automatically recognize text with improved optical character recognition (OCR)		•	•	•
Capture web pages as interactive PDF files for review and archiving from Microsoft Internet Explorer and Firefox with one-button ease[3]		•	•	•
Archive emails or email folders from Microsoft Outlook or IBM® Lotus Notes with one-button ease[3]		•	•	•
Create PDF files from the clipboard, including text and images copied from external applications		•	•	•
Convert Autodesk® AutoCAD®, Microsoft Visio, and Microsoft Project files to PDF with one-button ease[3]			•	•
Export and edit PDF files				
Save PDF files as Microsoft Word documents and Excel spreadsheets, retaining the layout, fonts, formatting, and tables		•	•	•
Quickly and easily edit PDF files by making simple changes to text		•	•	•
Insert, extract, replace, delete, rotate, or reorder pages in a PDF file		•	•	•
Split large PDF files into multiple files based on maximum file size, maximum pages per file, or bookmarks		•	•	•
Add rich media to PDF files				
Insert audio, Adobe Flash® Player compatible video, and interactive media for direct playback in Acrobat and Adobe Reader[4]			•	•
Convert a wide variety of video formats for smooth playback in PDF with Adobe Media Encoder				•
Edit and enhance photos to add to your PDF communications with Adobe Photoshop® CS5, the industry standard for image editing				•
Quickly transform static PowerPoint slides into compelling, interactive PDF presentations with Adobe Presenter				•
Rapidly combine audio, video, screen recordings, slides, and more into a rich media experience with Adobe Captivate®				•

› fixed set of features

› pre-determined selection of features

› static integration

› perfect coordination possible



Choose Your Options

Next: Summary & Payments

Options | Standard Equipment

Marked options will require changes to your current selections.

Packages	MSRP*
Roof Package (Details)	$2,030
Mechanical	**MSRP***
Engine: 6.2L V8 SFI	Incl.
Transmission: 6 Speed Manual Short Throw (Details)	Incl.
Transmission: 6-Speed Paddle Shift w/Automatic (Details)	$1,565
Magnetic Selective Ride Control (Details)	$2,915
Battery Maintainer (Details)	$115
Performance Brakes (Details)	$575
Exterior	**MSRP***
Tires: P245/40ZR18 Fr & P285/35ZR19 Rr (Details)	Incl.
Front License Plate Mount BC/MB/NB/ON (Details)	$0
Front License Plate Mt. AB/NL/NS/NT/NU/PE/QC/SK/YT (Details)	$15
Cyber Gray Metallic Head Lamp Bezel	$675
Blade Silver Metallic Head Lamp Bezel	$675
Black Head Lamp Bezel	$675
1-Piece Removable Transparent Roof Panel (Details)	$1,095
Dual Mode Performance Exhaust (Details)	$1,555
Entertainment	**MSRP***
Radio: AM/FM Stereo w/CD Player & MP3 Playback (Details)	Incl.

› fixed set of features

› semi-configurable selection of features

› set of static integrations

› perfect coordination possible, but impractical

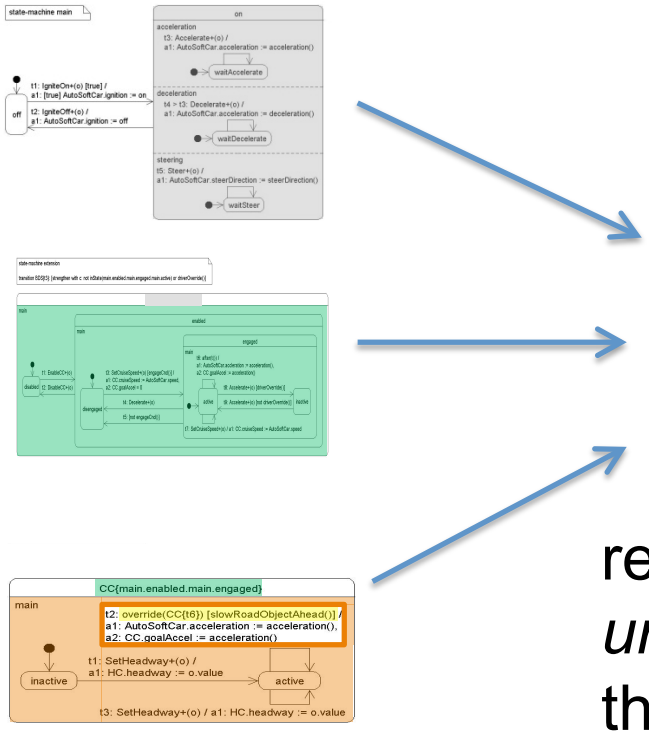
› unlimited features

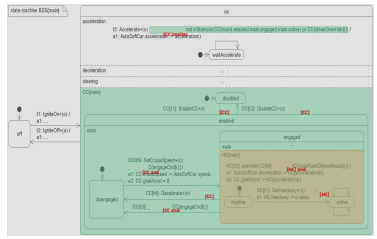
› user-defined selection of features

› dynamic integration

› loose coordination

summary





Properties

PL Model Checker

Invalid Configurations + Counter Example

resolve classes of *undesired* interactions through *composition*

model features modularly with *intended* interactions

detect remaining *unintended* interactions