Detecting and Resolving Software Errors

Jo Atlee • University Research Talk • Jan 2017

Waterloo Formal Methods Group (WatForm)
David R. Cheriton School of Computer Science
University of Waterloo

60 YEARS OF INNOVATION
UNIVERSITY OF WATERLOO

WATFORM

WatCAR

GM
CANADA

IBM

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School
of Computer Science

ICR Institute for
Computer Research

Malina
SOFTWARE CORP.

buggy software

Quest
Microsoft patches holes in Outlook
Royal Bank
Nasdaq Trading Software G
'glitch' that sold flight packages for 90% off
pany's Shares to Soar Nearly 30%
JobMine
TD hit by smartphone bug
Air Canada not hon
Canada recal
Canada Softw
Waterloo LEARN
Nationwide Resolved
Air Canada Software Glitch Affecting Services
pgrade Goes Awry
ter Errors
nked to AECL
FORE
in Diebold Voting Software Caused Lost Ballots
Fatal Error – Radiation D
Concur
Bombardier delays C Series due to software
Claim HealthCare.gov is Still Flawed

who is to blame?

are the software developers incompetent or negligent?

OR

is it really so difficult to build error-free software?

who is to blame?

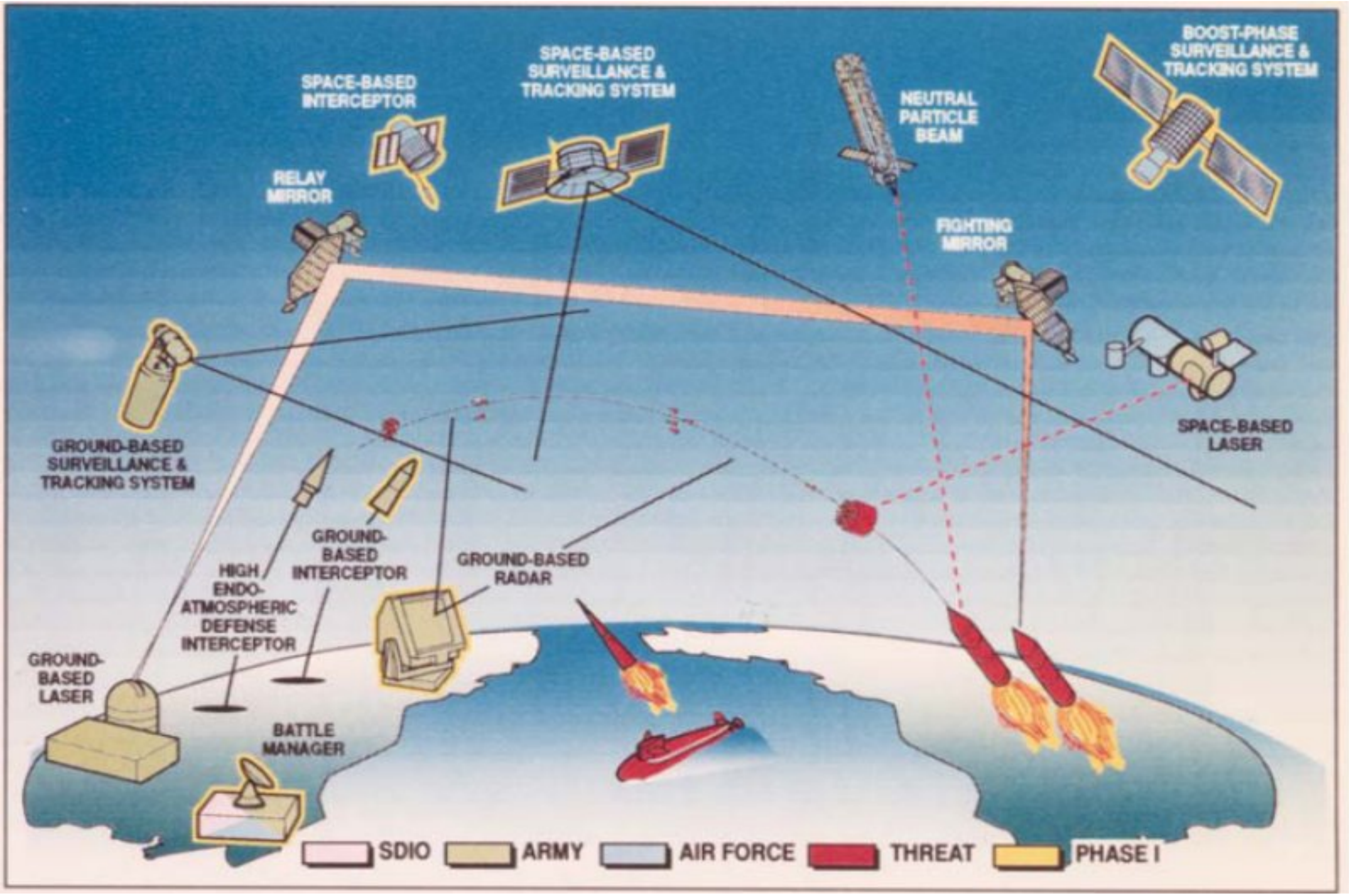
are the software developers incompetent or negligent?

Sometimes

is it really so difficult to build error-free software?

YES

limits on software correctness

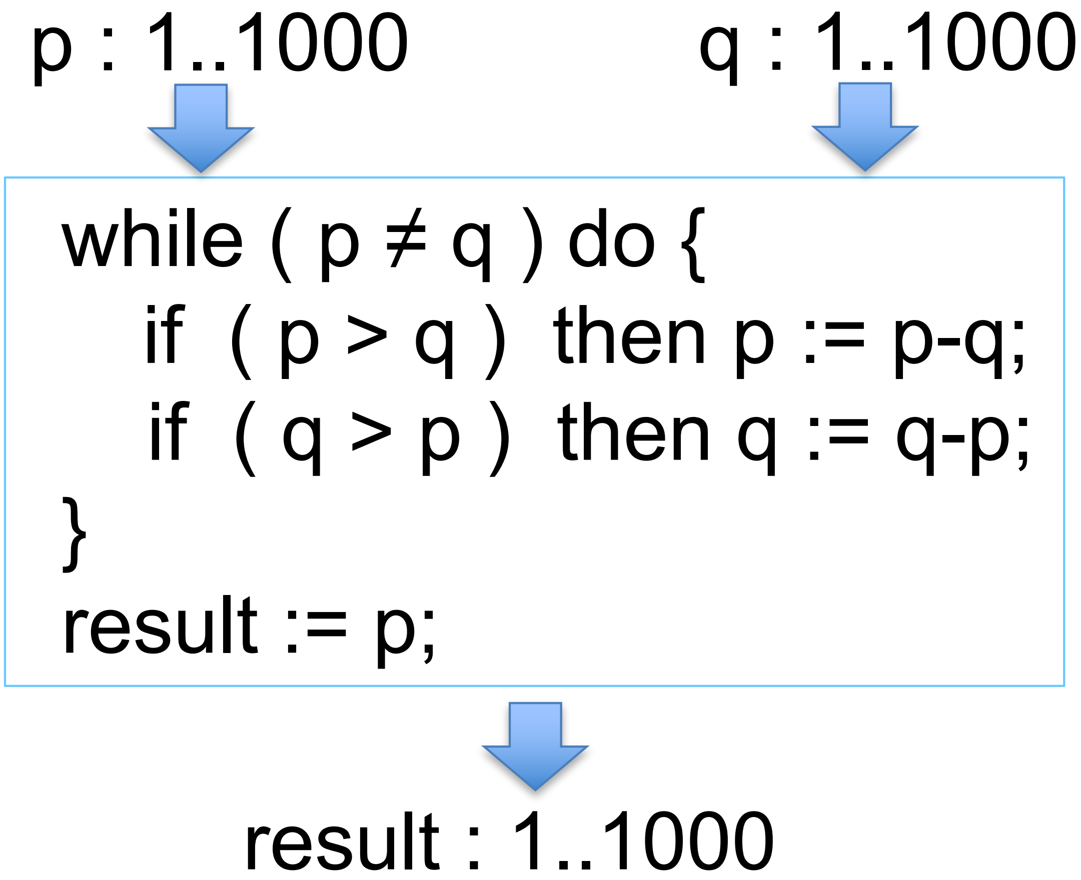


Strategic Defense Initiative (“Star Wars”)

incomplete testing

it is generally impossible to exhaustively test all possible inputs.

GREATEST COMMON DIVISOR



p : 1..1000

q : 1..1000

```
while ( p ≠ q ) do {
    if  ( p > q )  then p := p-q;
    if  ( q > p )  then q := q-p;
}
result := p;
```

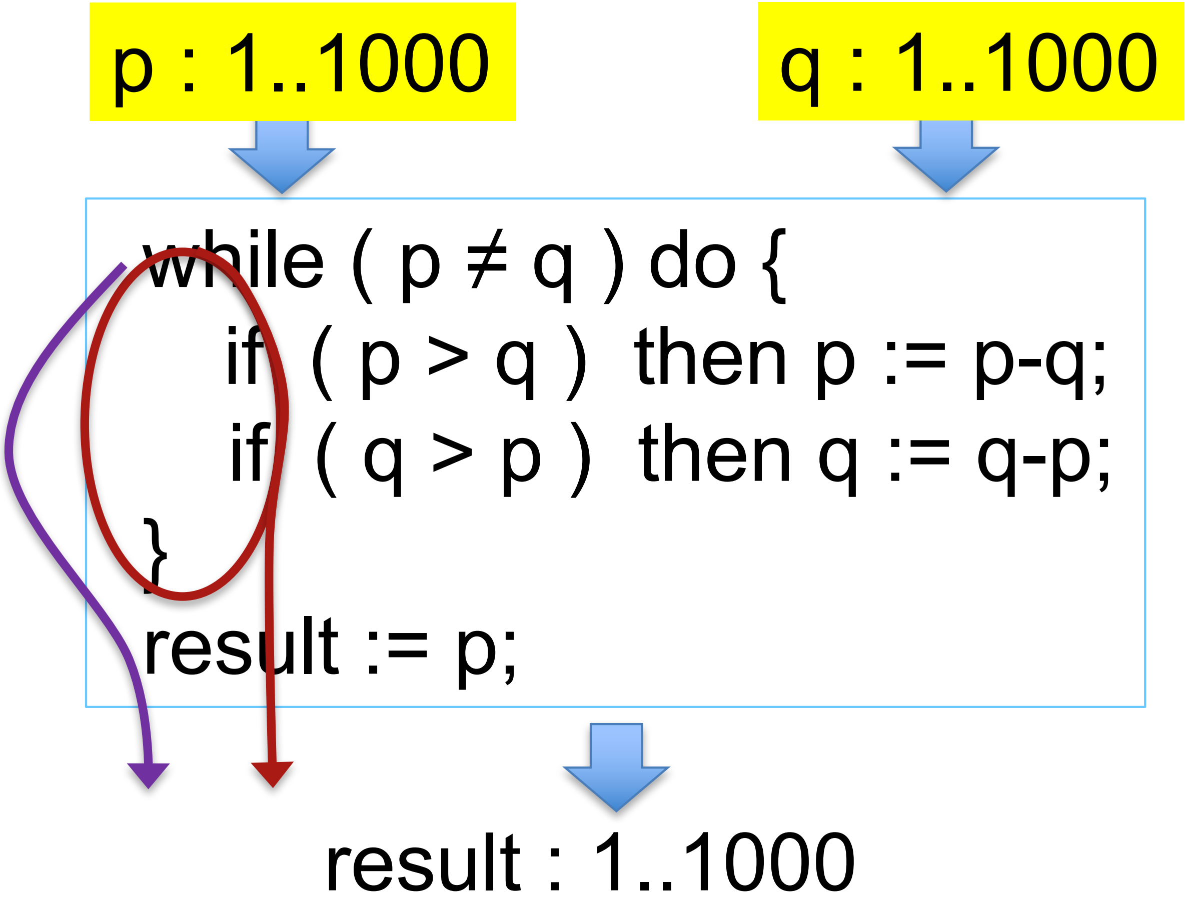
result : 1..1000

1,000,000 possible inputs

discontinuous behaviour

cannot simply test software on a sample of input values and consider the software thoroughly tested.

GREATEST COMMON DIVISOR



p : 1..1000

q : 1..1000

```
while ( p ≠ q ) do {
    if ( p > q )  then p := p-q;
    if ( q > p )  then q := q-p;
}
result := p;
```

result : 1..1000

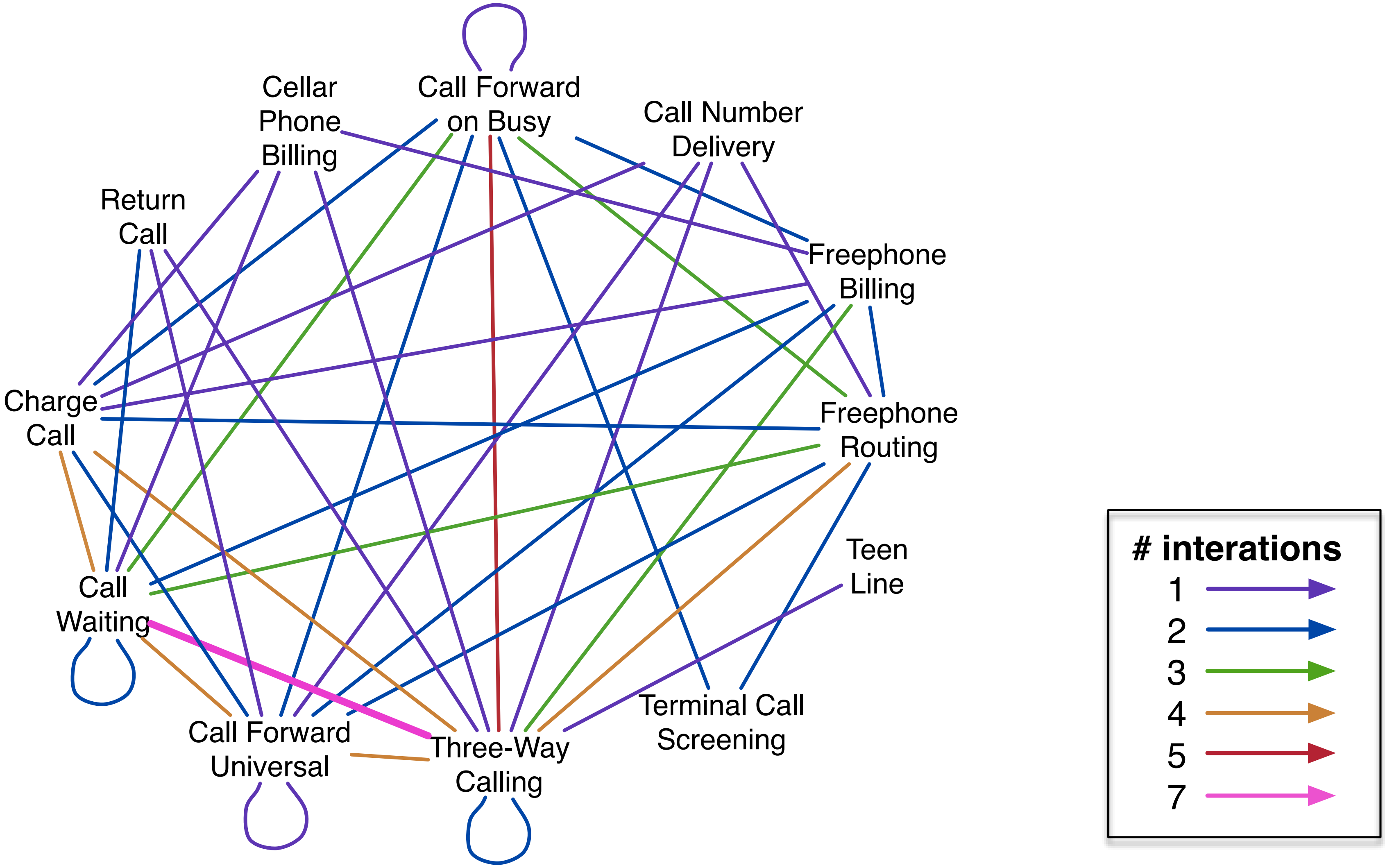
p = 100, q = 100
result = 100

vs.

p = 100, q = 101
result = 1

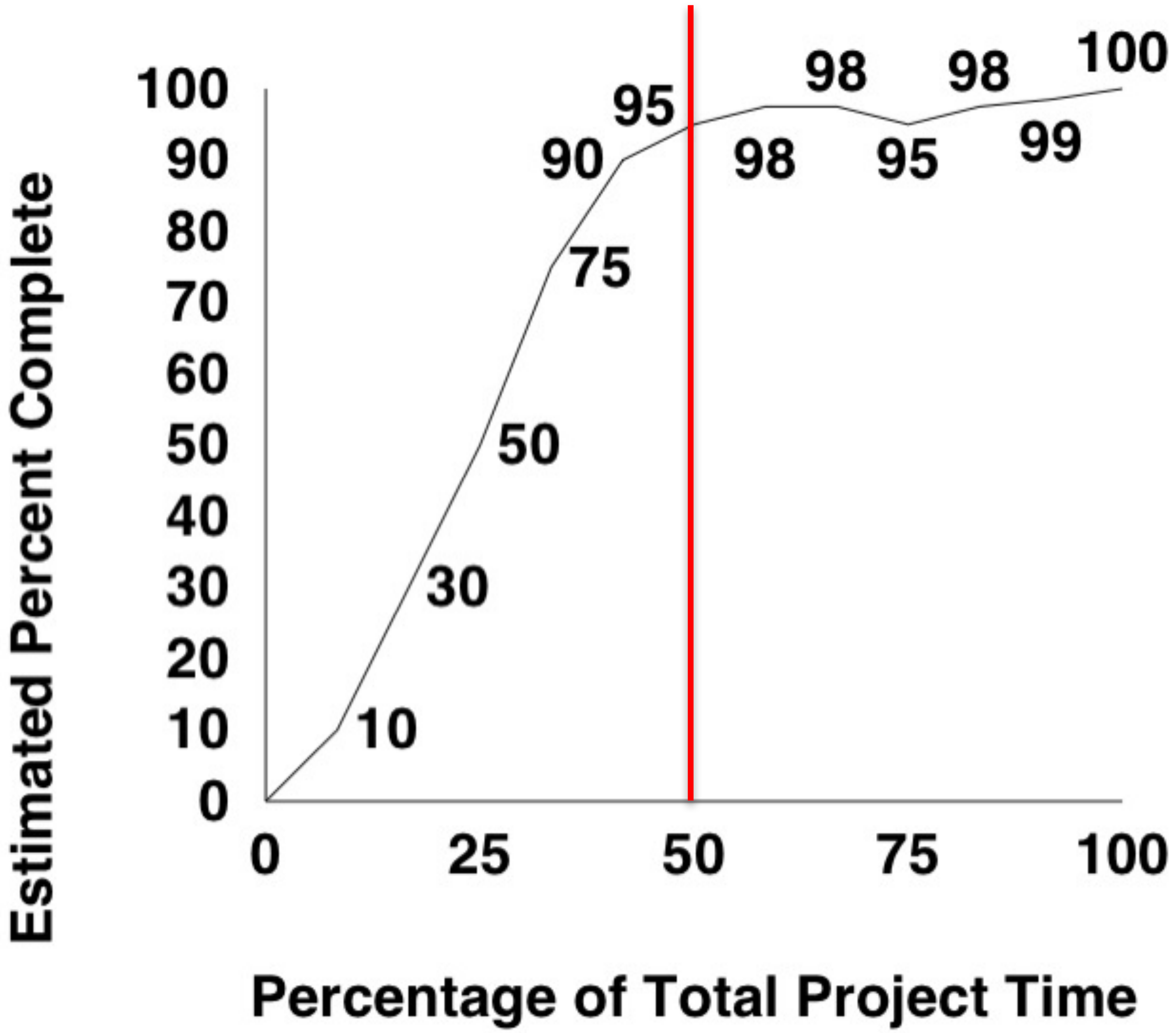
integration problems

separately developed software modules can interact in unintended and surprising ways….



integration problems

… to the point that software engineers routinely underestimate how long software integration takes.



poor understanding of the environment

errors also occur when software has an incorrect or imprecise model of its operating environment.

Denver Airport Automated Baggage System

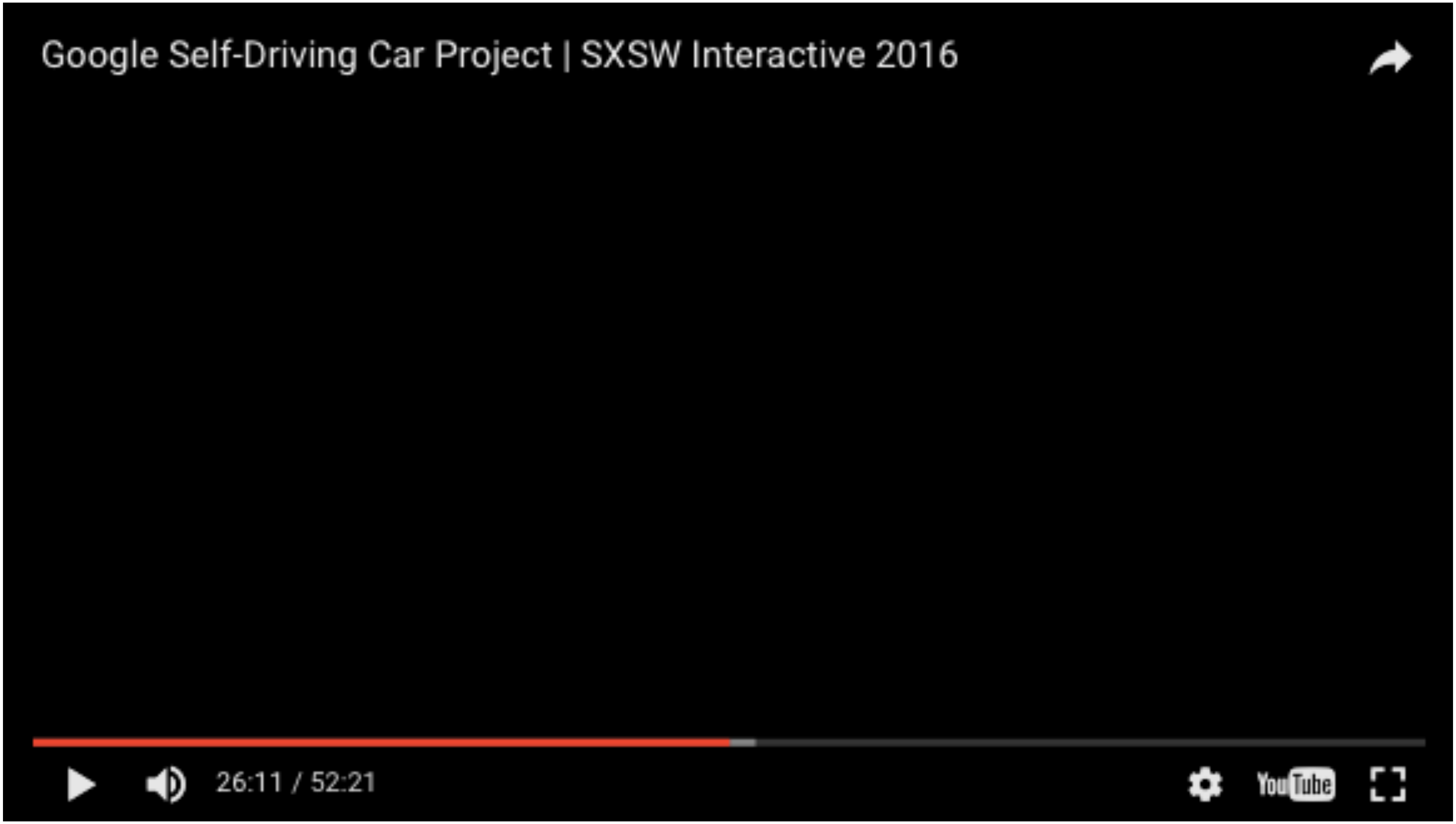
understanding software's environment

much of the purpose of test driving the Google car is to acquire data to build a world model for the car

1. Precise mapping of streets
2. "Learning" the behaviours of pedestrians, cyclists, etc.

understanding software’s environment

Chris Urmson, director of Google’s self-driving car project, keynote at South by Southwest 2016

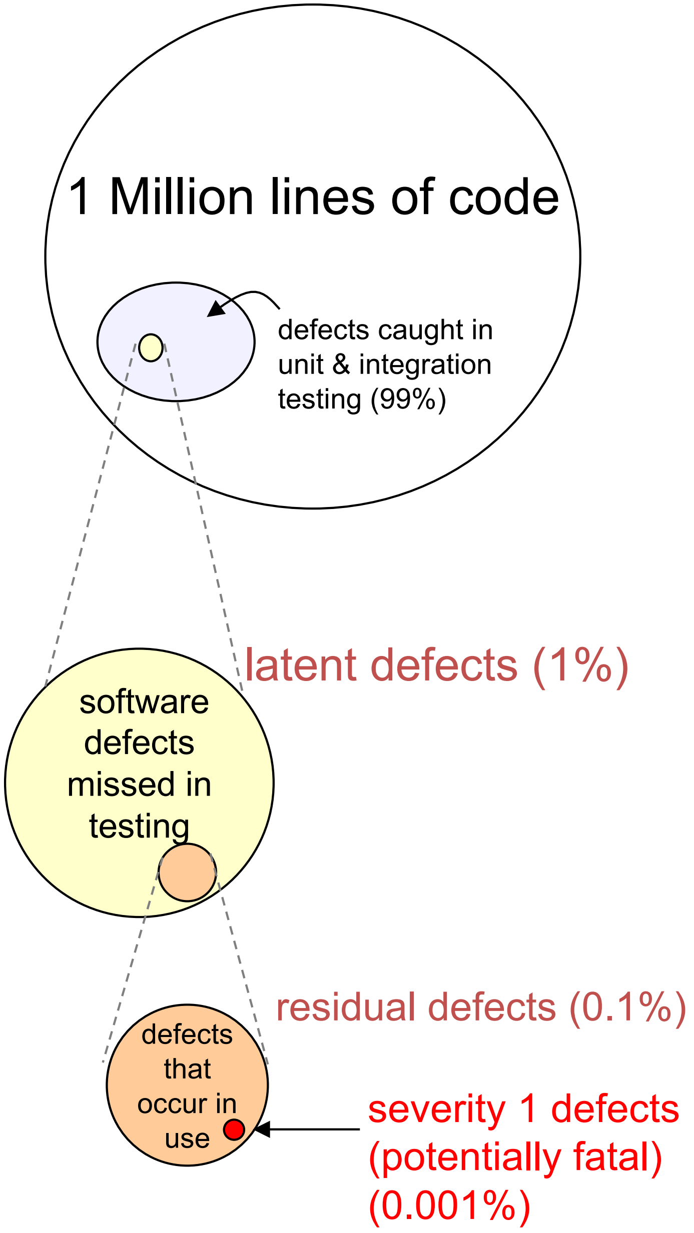


Google Self-Driving Car Project | SXSW Interactive 2016

residual defects in software

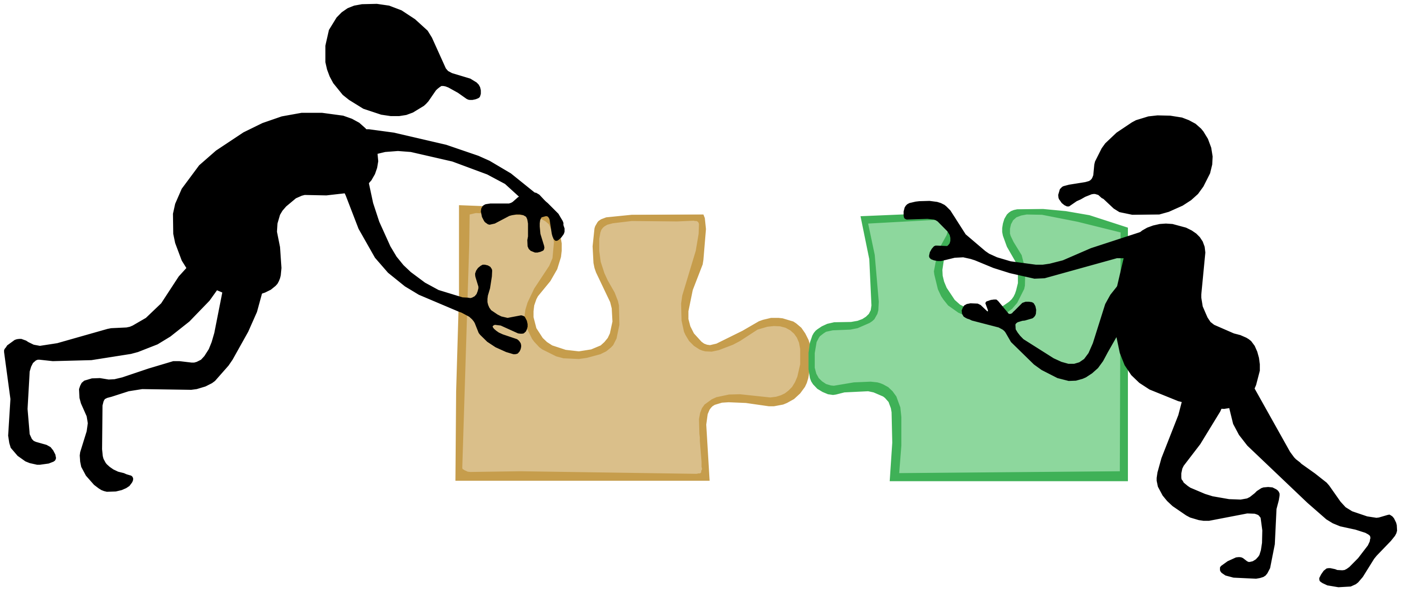
Estimates put the residual defect rate for a good industry-standard software process at 1-10 per 1000 lines of code

The residual defect rate for an exceptionally good (e.g., NASA) software process can be as low as 0.1 per 1000 lines of code


1 Million lines of code
defects caught in unit & integration testing (99%)
latent defects (1%)
software defects missed in testing
residual defects (0.1%)
defects that occur in use
severity 1 defects (potentially fatal) (0.001%)

NASA, Flight Software Complexity Briefing, 2008
(Accessed Jan. 20, 2017: https://acc.dau.mil/adl/en.../FlightSoftwareComplexityBriefing_v5.ppt)

feature interactions



what is a feature?

feature: a unit of functionality

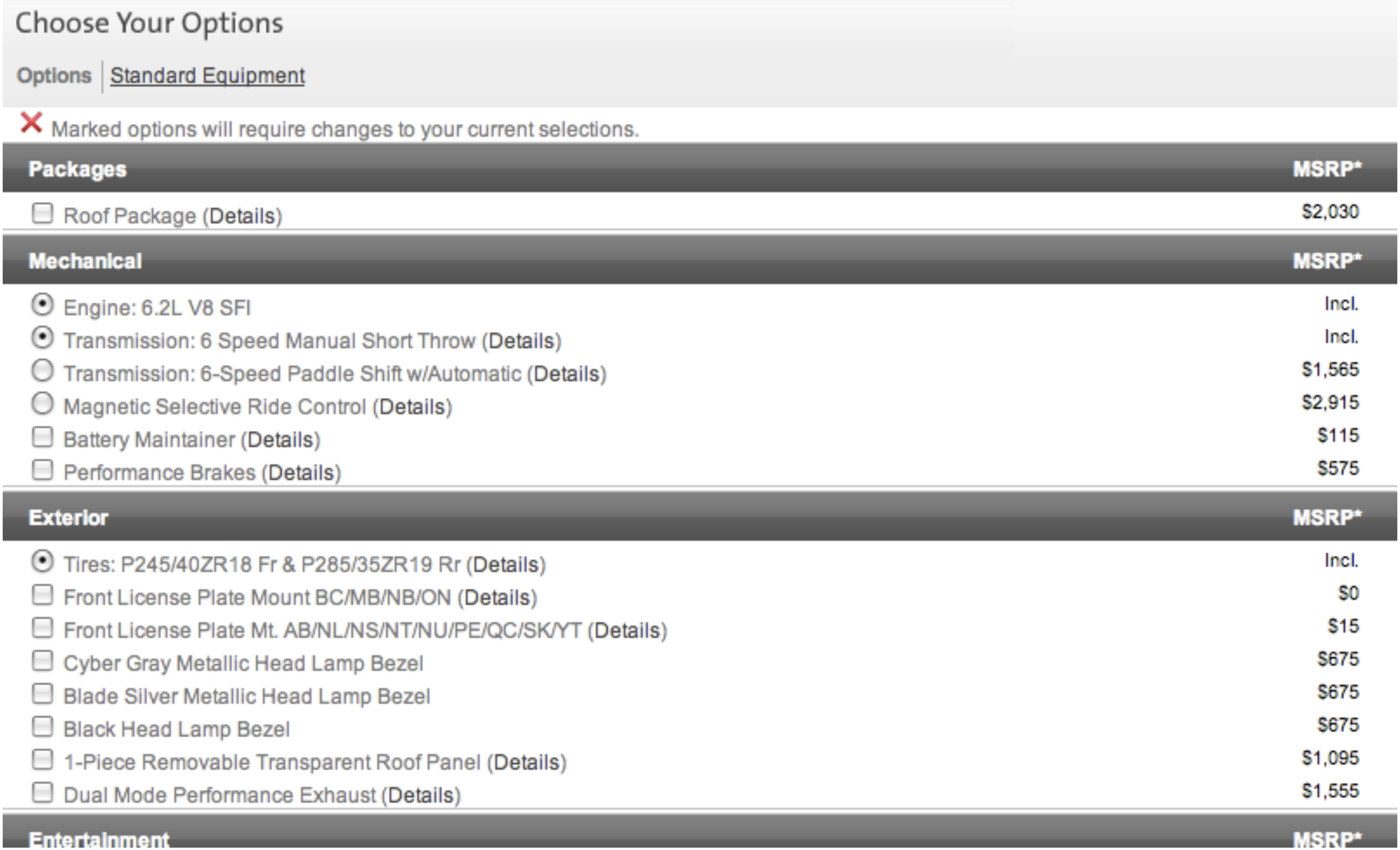
Undo
Tables, Figures
Change tracking
Word count
Spell check
Watermarks
Hyperlinks
Email

Call display
Call display blocking
Redial
Voicemail
Call waiting
Call transfer
Three-way calling
Distinctive ring

features
comparison shopping

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[1]	•	•	•	•
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[2]		•	•	•
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[2]		•	•	•
Archive emails or email folders from Microsoft Outlook or IBM® Lotus Notes with one-button ease[2]		•	•	•
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[2]			•	•
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®[2]			•	•
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®				•

features
mass customization



Choose Your Options

Options | Standard Equipment

✕ Marked options will require changes to your current selections.

Packages	MSRP*
□ Roof Package (Details)	$2,030

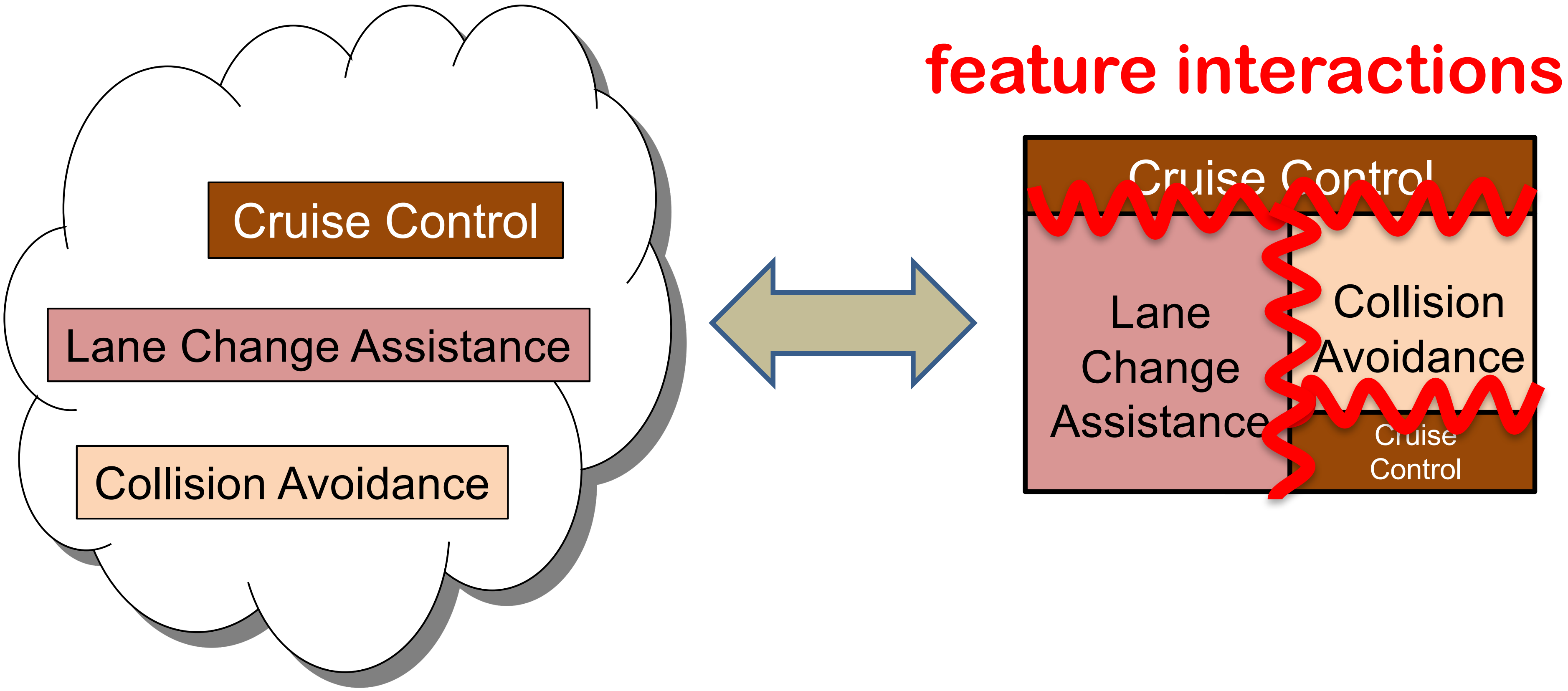
Mechanical	MSRP*
● Engine: 6.2L V8 SFI	Incl.
● Transmission: 6 Speed Manual Short Throw (Details)	Incl.
○ Transmission: 6-Speed Paddle Shift w/Automatic (Details)	$1,565
○ Magnetic Selective Ride Control (Details)	$2,915
□ Battery Maintainer (Details)	$115
□ Performance Brakes (Details)	$575

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*

feature-oriented software development

feature : a work piece



feature interactions

Cruise Control

Lane Change Assistance

Collision Avoidance

Cruise Control

Lane Change Assistance

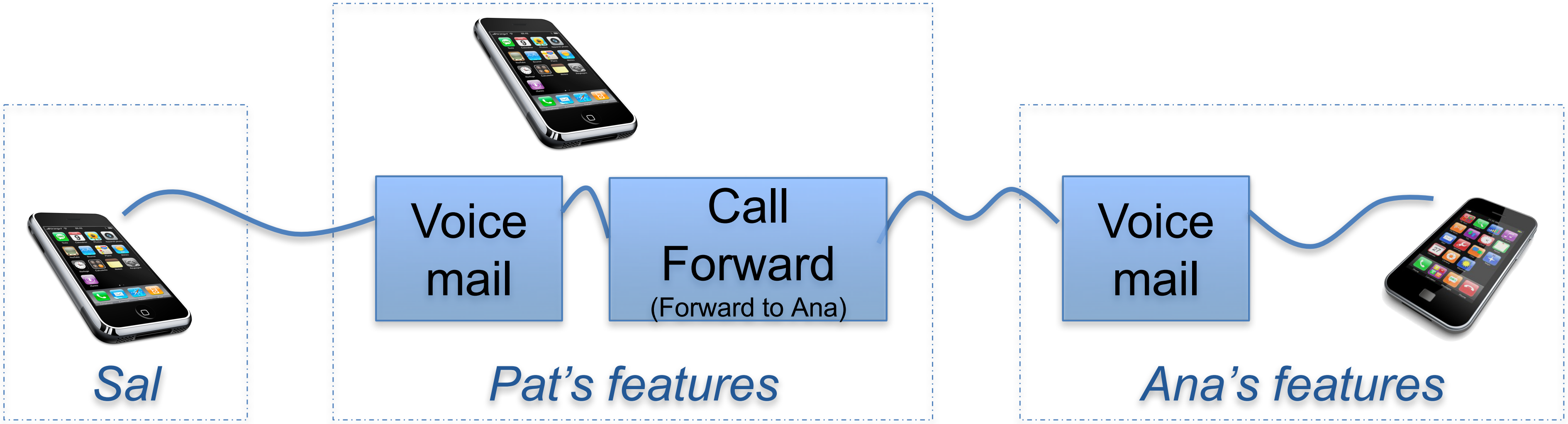
Collision Avoidance

Cruise Control

stakeholders' mental model of system

feature-oriented software system

voice mail ⊕ call forward



- Pat forwards all of her calls to Ana
- Sal calls Pat
- The call attempt fails

Whose VM should react?

- what if Pat is a sales group and Ana is a sales representative?
- what if Pat is on a long leave of absence?

cruise control ⊕ traction control

cruise control

- vehicle set to maintain driver-specified speed

traction control

- brake fluid applied when wheels slip

interaction

- traction control applies brake to slipping wheel
- cruise control increases engine power (to maintain speed)
- driver senses “sudden acceleration”
  - vehicle becomes difficult to control

resolution

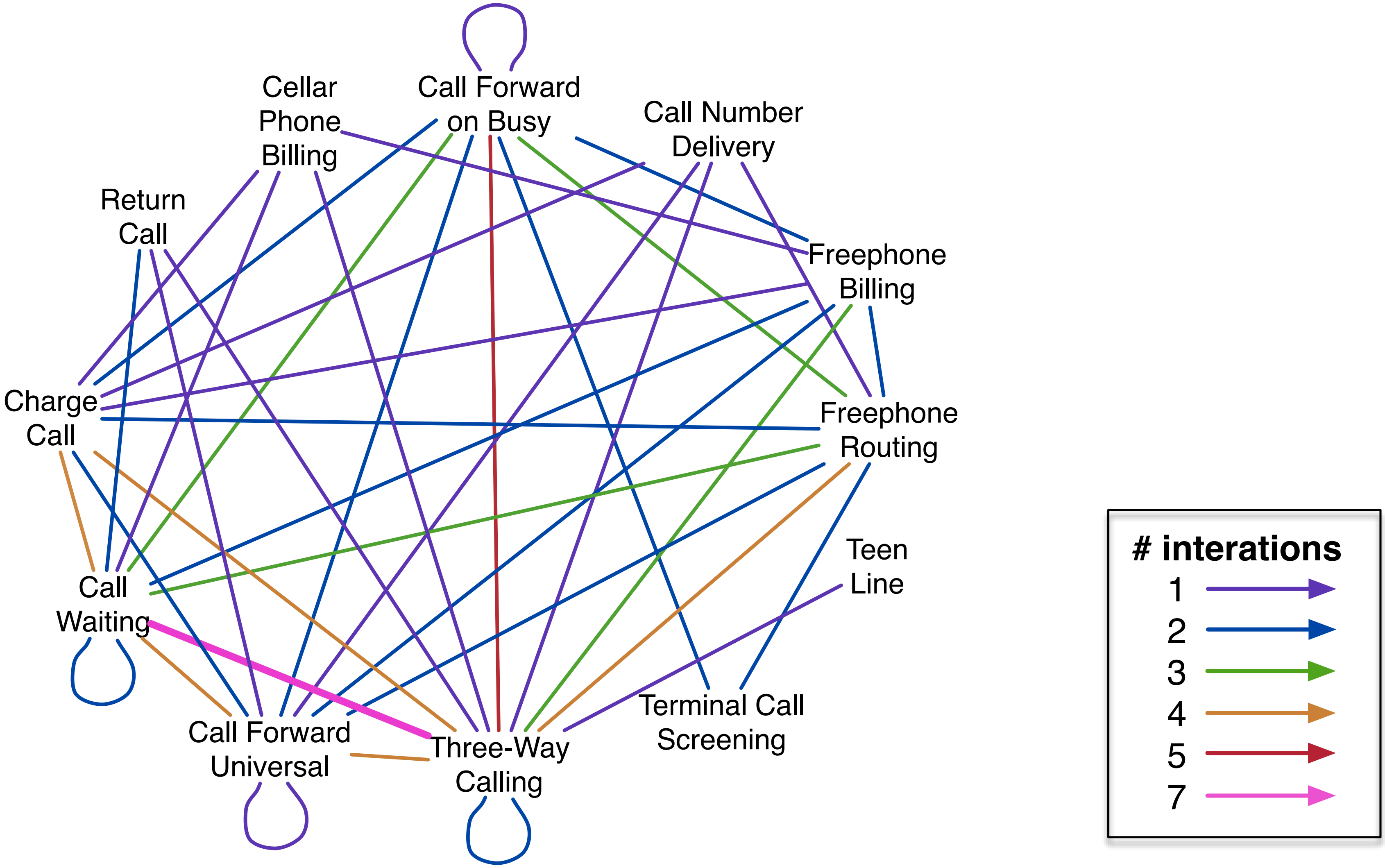
- advise drivers not to use cruise control on slippery roads

interaction resolutions as **exceptions**

$$\begin{aligned}\text{Cruise Control} = {} & \textit{basic cruise control} \\ & + e_{\textit{traction control}} \\ & + e_{\textit{speed limit control}} \\ & + e_{\textit{headway control}} \\ & + e_{\textit{forward collision alert}} \\ & + e_{\textit{brake pressed}} \\ & \dots\end{aligned}$$

lots of feature interactions

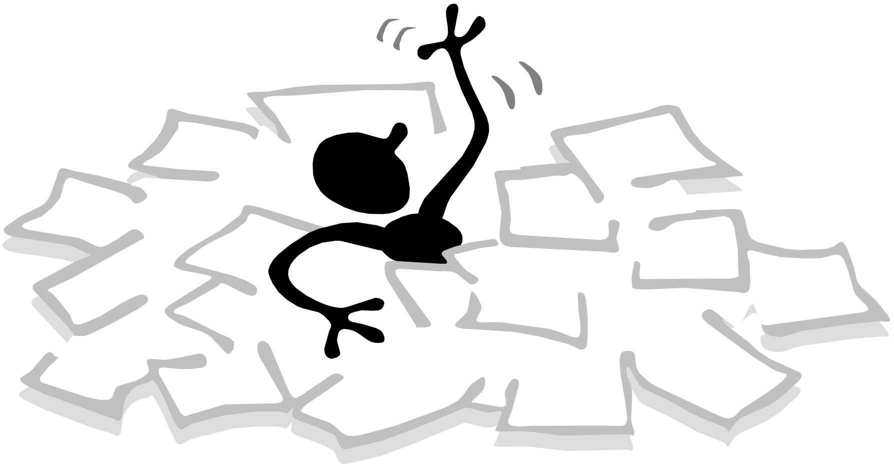
all interactions require work – to detect, debug, fix, and test



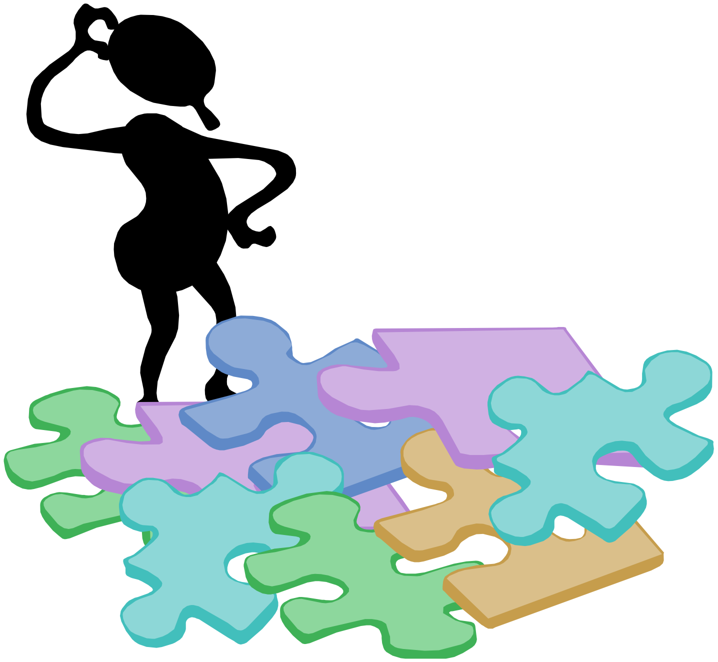
feature interaction problem

death by exceptions [Zave]

$$F_1 = f_1 + e_{f_2} + e_{f_3} + e_{f_4} + e_{f_5} + e_{f_6} + e_{f_7} + e_{f_8} + e_{f_9} + e_{f_{10}} + e_{f_{11}} + e_{f_{12}} + e_{f_{13}} + e_{f_{14}} + e_{f_{15}} + e_{f_{16}} + \ldots$$

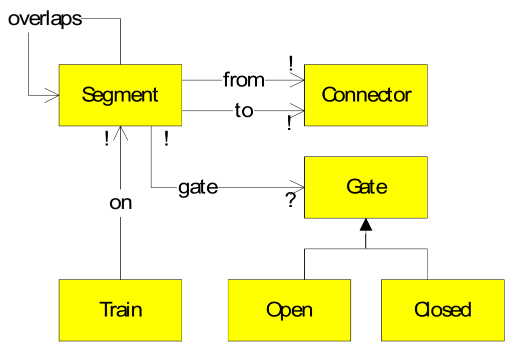


detecting feature interactions

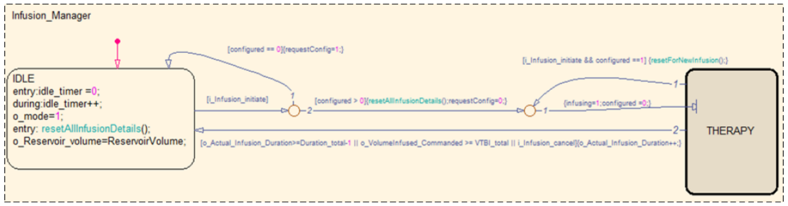


Waterloo Formal Methods (WatForm)

the use of mathematics to model and reason about computer systems - usually for the purpose of ensuring that the system will be acceptable.



models of structure
(e.g., software structure, rich data)


Infusion_Manager
IDLE
entry:idle_timer =0;
during:idle_timer++;
o_mode=1;
entry: resetAllInfusionDetails();
o_Reservoir_volume=ReservoirVolume;
[configured == 0]{requestConfig=1;}
[i_Infusion_initiate]
1
2
[configured > 0]{resetAllInfusionDetails();requestConfig=0;}
[i_Infusion_initiate && configured ==1] {resetForNewInfusion();}
1
{infusing=1;configured =0;}
1
2
[o_Actual_Infusion_Duration>=Duration_total-1 || o_VolumeInfused_Commanded >= VTBI_total || i_Infusion_cancel]{o_Actual_Infusion_Duration++;}
THERAPY

executable models
(e.g., automata, process algebras)

AG(NavUpd=AflyUpd →
(WpnDel=BOC) v (WpnDel=BOC))

property languages
(e.g., logics, constraint languages)

feature interaction as a math problem

executable model of feature

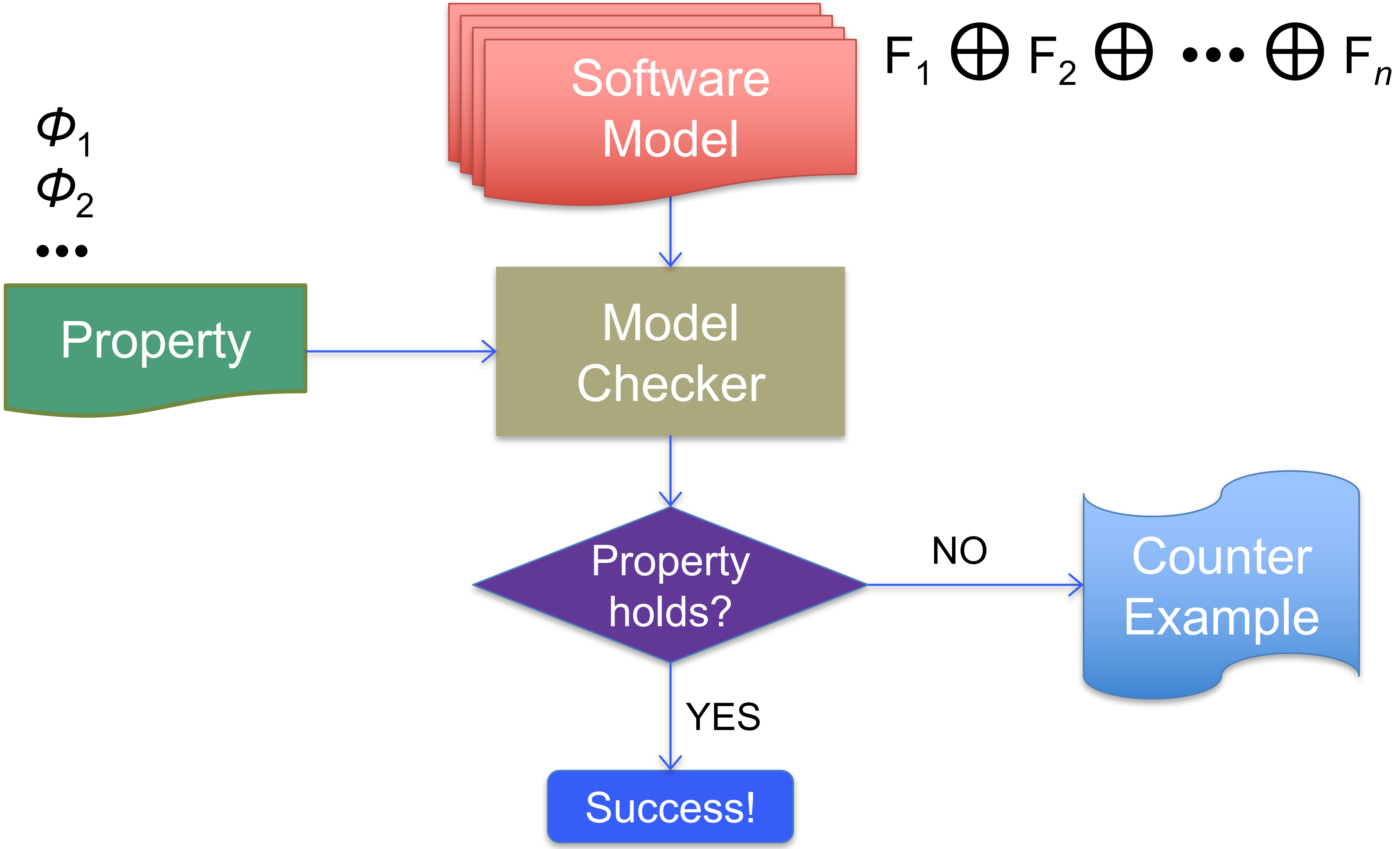
property of feature

$$
\begin{array}{c}
F_1 \models \Phi_1 \\
F_2 \models \Phi_2 \\
\vdots \\
F_n \models \Phi_n
\end{array}
$$

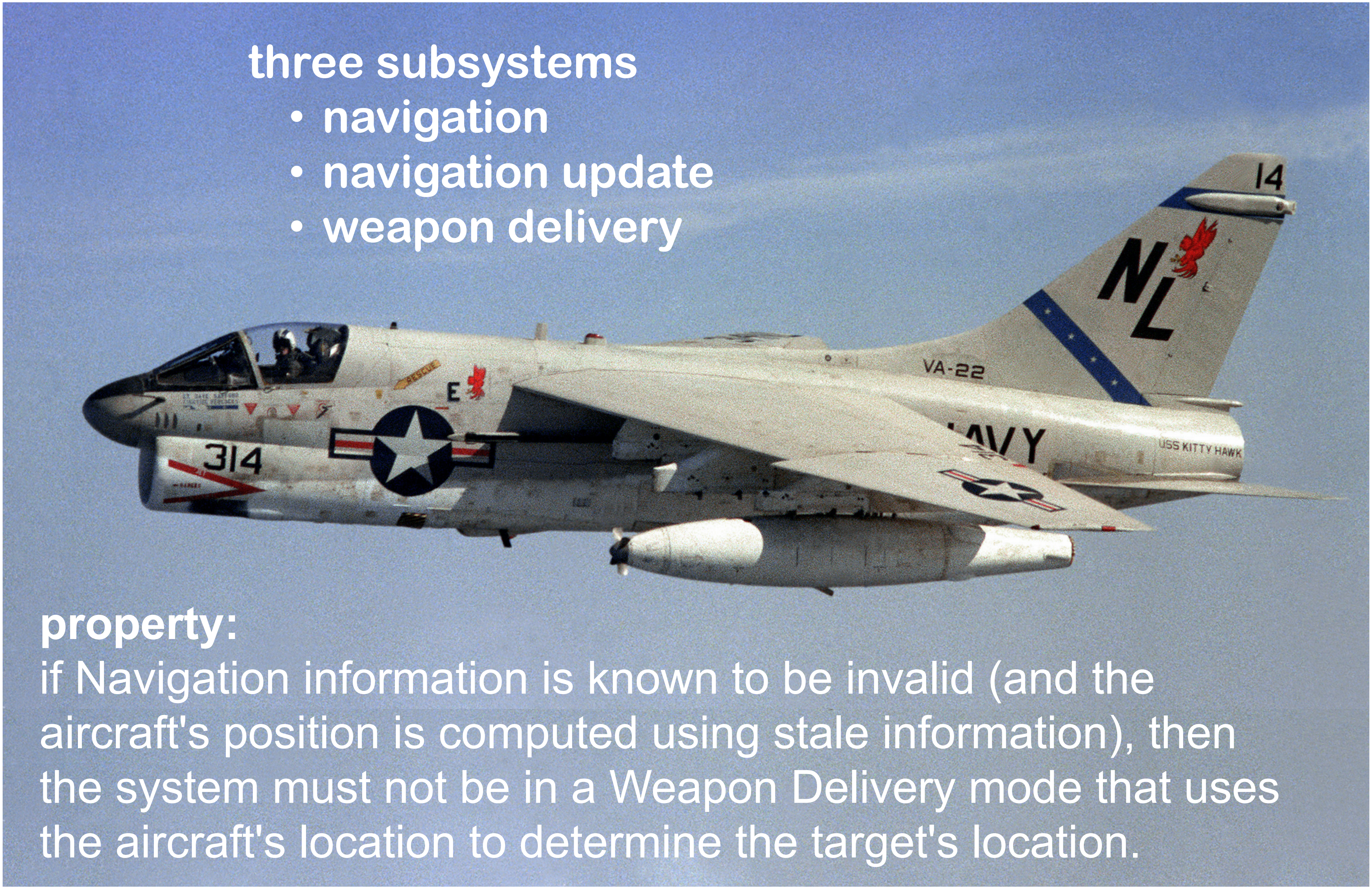
$$\underbrace{F_1 \oplus F_2 \oplus \cdots \oplus F_n}_{\text{feature composition (= product)}} \not\models \Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_n$$

model checking

Clarke, Emerson ‘81, Queille, Sifakis ‘82



example: US navy aircraft A-7E

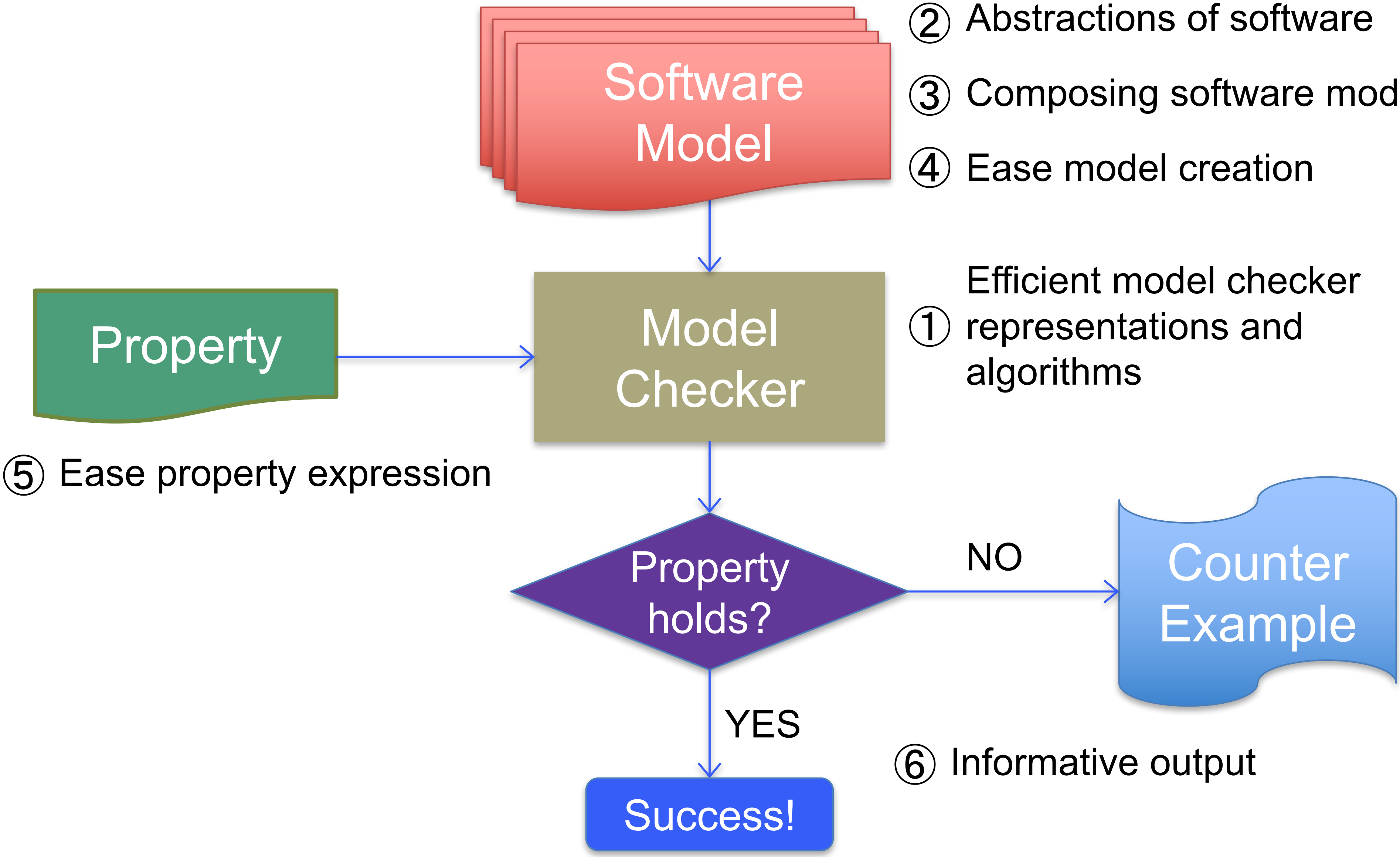


three subsystems

- navigation
- navigation update
- weapon delivery

property:
if Navigation information is known to be invalid (and the aircraft's position is computed using stale information), then the system must not be in a Weapon Delivery mode that uses the aircraft's location to determine the target's location.

research problems



Software Model

② Abstractions of software

③ Composing software models

④ Ease model creation

Property

Model Checker

① Efficient model checker representations and algorithms

⑤ Ease property expression

Property holds?

NO

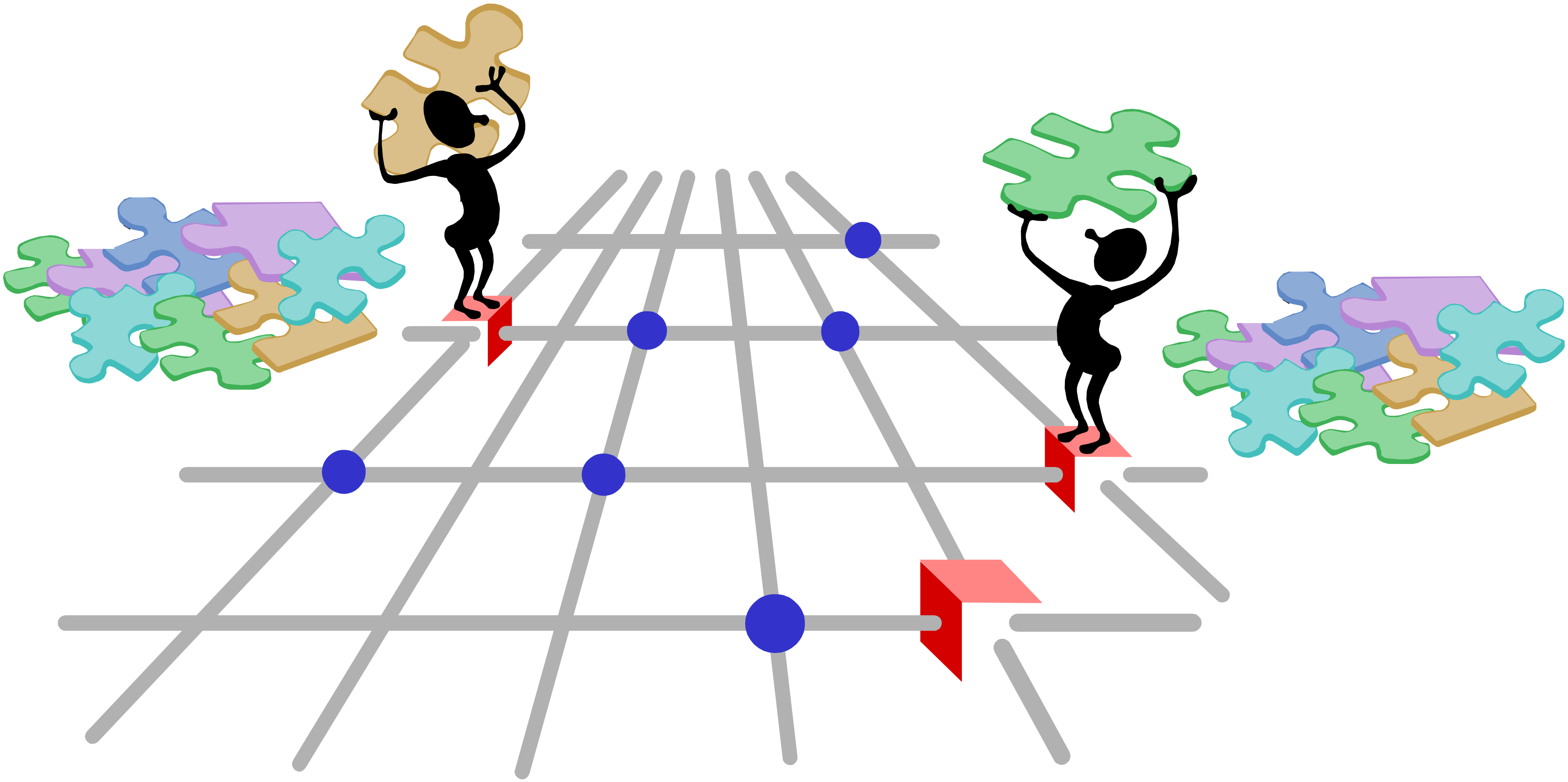
Counter Example

YES

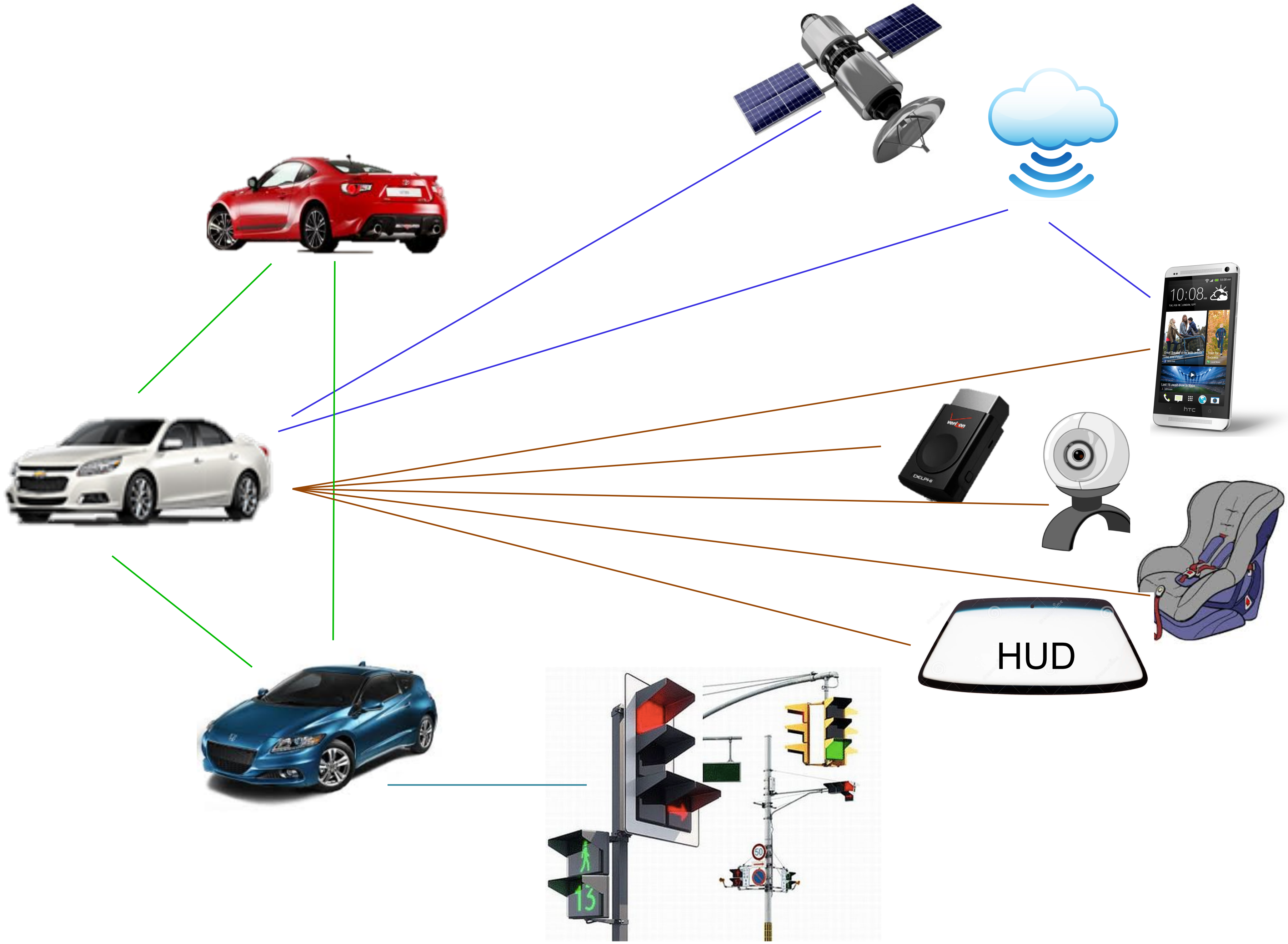
Success!

⑥ Informative output

system of systems

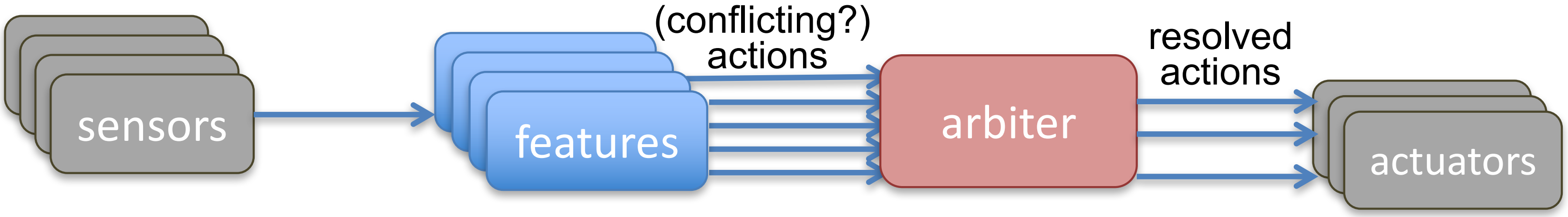


connected products / after-market upgrades



runtime interaction resolution

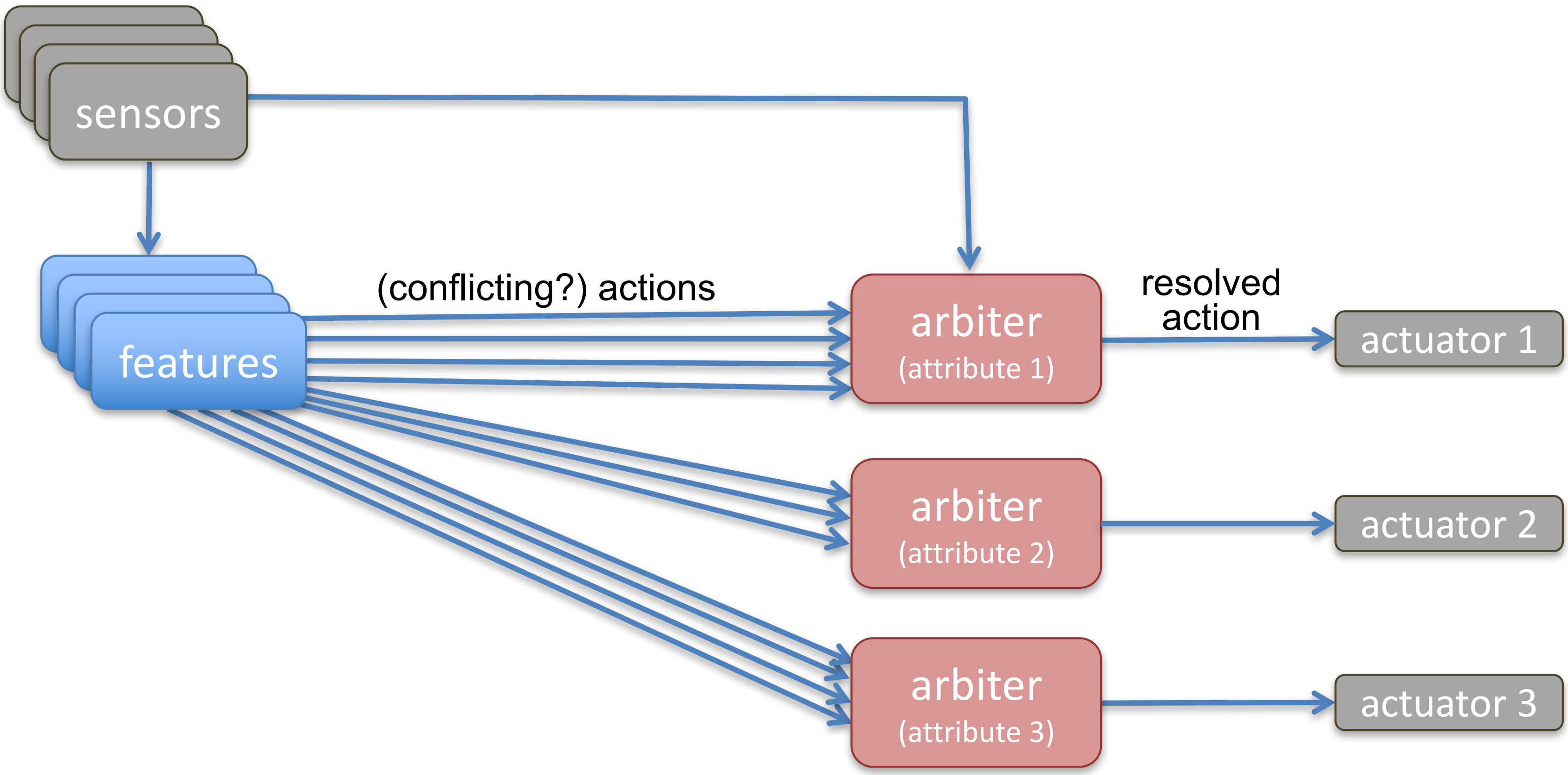
centralized arbiter solution:



… doesn’t work for systems of systems

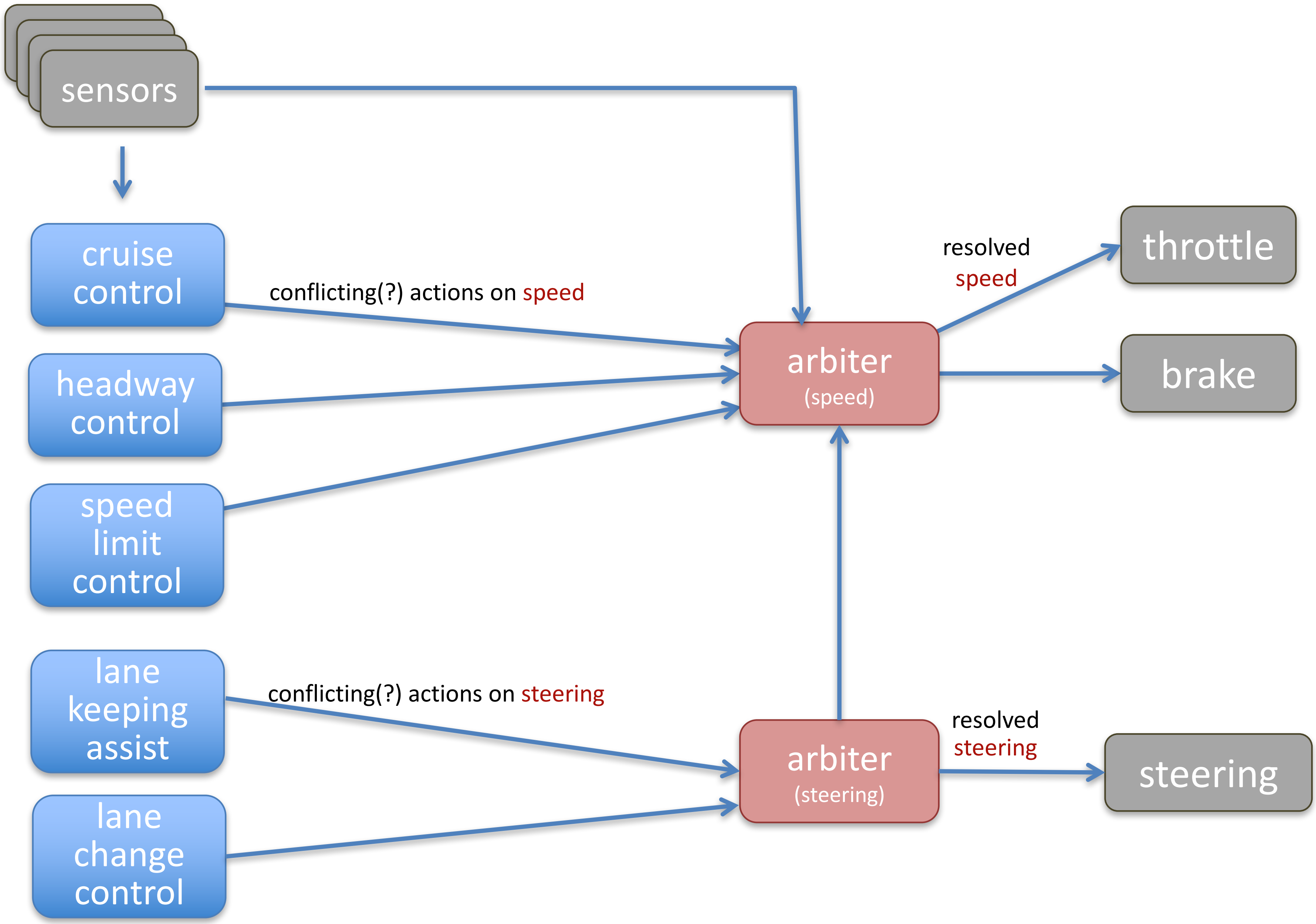
runtime interaction resolution

actuator-specific arbiter solution:



+ arbiters are co-located with their respective actuator
+ arbiters are actuator specific
+ arbiters are feature agnostic

example


sensors
cruise control
headway control
speed limit control
lane keeping assist
lane change control
conflicting(?) actions on speed
conflicting(?) actions on steering
arbiter (speed)
arbiter (steering)
resolved speed
resolved steering
throttle
brake
steering

takeaways

it is unrealistic to assume that a zero-defect rate in software is possible

mathematical models and automated reasoning can help to detect tricky interaction errors within a product

for interconnected products, we need strategies for resolving interactions at runtime