

# Automated Analysis of Natural-Language Requirements: Industrial Needs and Opportunities

AIRE'2018 @ RE'2018

Lionel Briand  
Interdisciplinary Centre for ICT Security, Reliability, and Trust (SnT)  
University of Luxembourg, Luxembourg

# Acknowledgements

- Mehrdad Sabetzadeh
- Chetan Arora
- Shiva Nejati
- Fabrizio Pastore
- Chunhui Wang

# Introduction

# Applications

- Requirements to support a **shared understanding** among stakeholders in (large) projects, e.g., software engineers and domain experts
- Requirements as **contract** with customers
- Requirements to **support compliance** with standards, e.g., traceability to tests
- Requirements to **support quality assurance**, e.g., system (security) testing
- Requirements to support **change control**
- Requirements to support **product-line configuration**

**For many of these applications,  
there is little automated tool support.**

# Forms of NL Requirements

- **Natural language statements, complying or not with templates**
- **User stories, following various templates**
- **Use case specifications, possibly structured and restricted**
- **Mixing natural language and models, e.g., class and activity diagrams**

**The best form of requirements depends on context and targeted applications.**

# Contextual Factors

- **No “right” way to express requirements**
- Domain complexity and criticality
- Regulatory compliance, e.g., standards
- Project size, team distribution, and number of stakeholders
- Background of stakeholders and communication challenges
- Presence of product lines with multiple customers
- Importance of early contractual agreement
- Frequency and consequences of changes in requirements

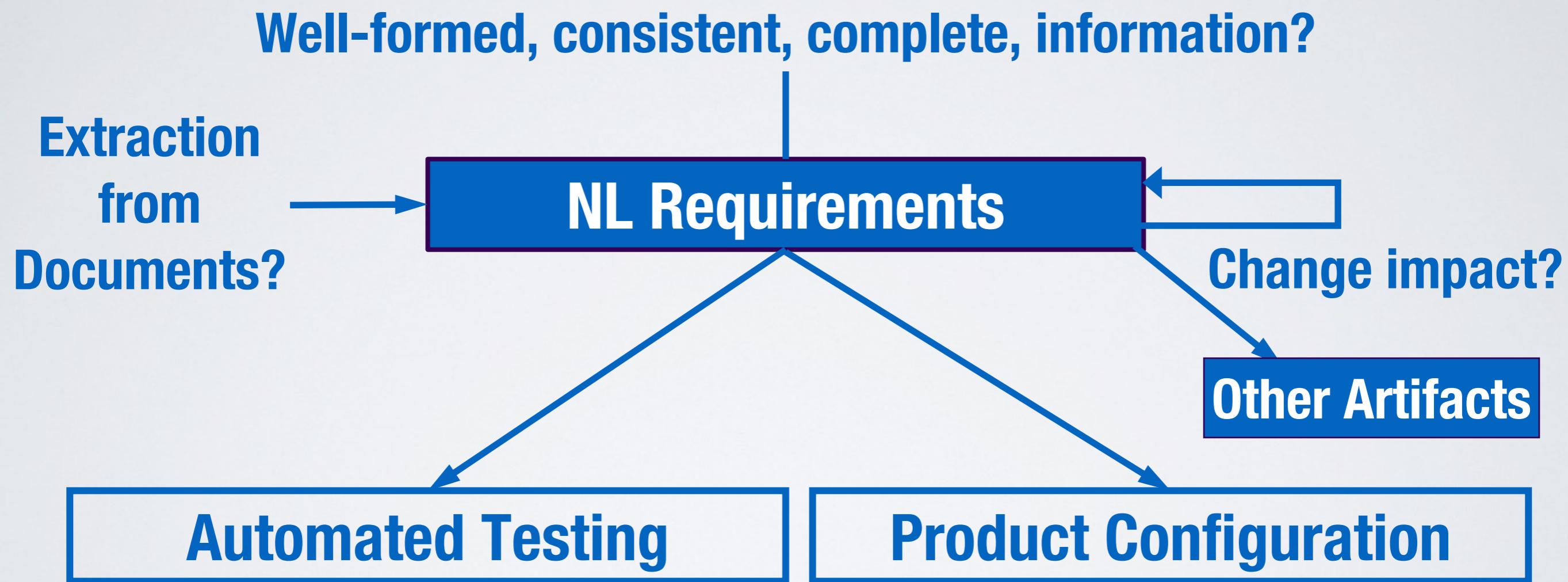
# Observations

- Natural language in requirements **won't go away.**
- The cost of rigorous requirements engineering must be justified by **clear automation benefits.**
- **Limited research** given the many and varying industrial automation needs, in widely varying contexts.

# Outline

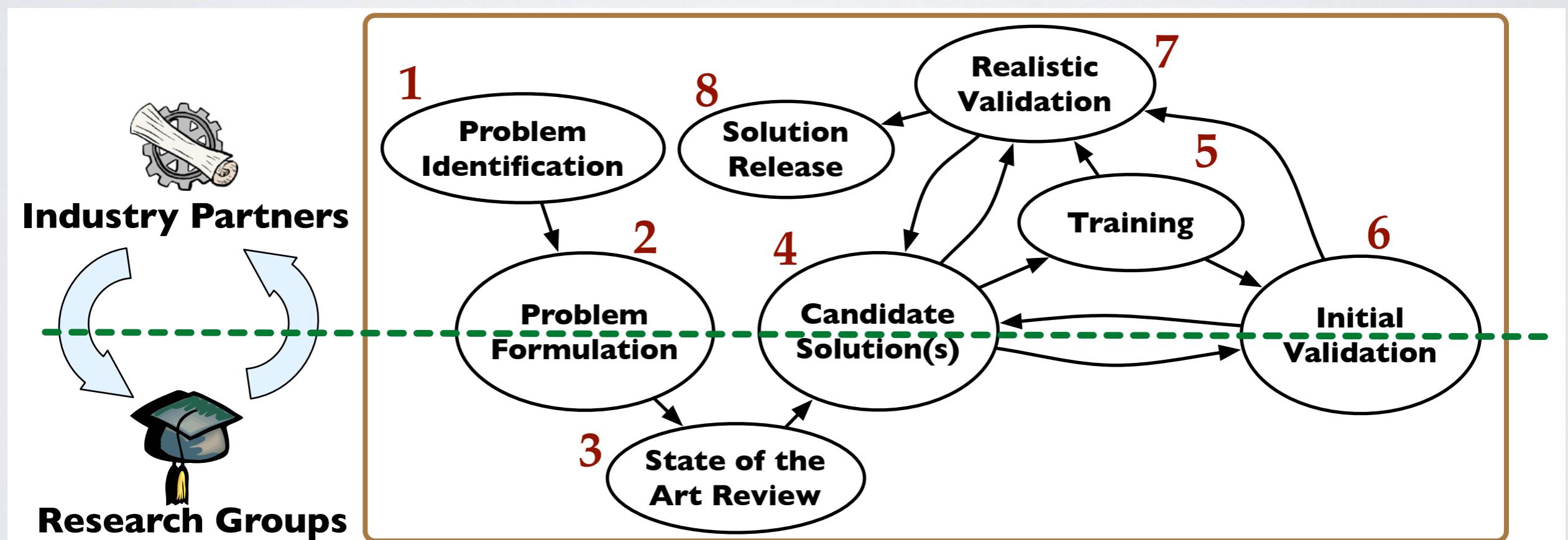
- Report on a variety of research projects
- Collaborations with industry
- Various objectives and applications
- Examples from automotive and satellite
- Lessons learned and reflections

# Overview



# Mode of Collaboration

- Research driven by industry needs
- Realistic evaluations
- Combining research with innovation and technology transfer



Adapted from [Gorschek et al. 2006]

# Change Impact Analysis

# Supporting Change

- Requirements change frequently
- Changes have **side-effects** on other requirements, design decisions, test cases ...
- How do we **support such changes** in ways that scale to hundreds of requirements or more?
- **Automated impact analysis**

# Inter-Requirements



Inter-Requirements  
Change Impact Analysis

# Approach

- Hundreds of requirements
- No traceability
- We propose an approach based on: (1) Natural Language Processing, (2) Phrase syntactic and semantic similarity measures

# Example

- **R1:** The mission operation controller shall transmit satellite status reports to the user help desk.
- **R2:** The satellite management system shall provide users with the ability to transfer maintenance and service plans to the user help desk.
- **R3:** The mission operation controller shall transmit any detected anomalies with the user help desk.

# Change

- **R1:** The mission operation controller shall transmit satellite status reports to the user ~~help desk~~ document repository.
- **R2:** The satellite management system shall provide users with the ability to transfer maintenance and service plans to the user help desk.
- **R3:** The mission operation controller shall transmit any detected anomalies with the user help desk.

# Challenge #1

## Capture Changes Precisely

- R1: The mission operation controller shall transmit satellite status reports to the user ~~help desk~~ document repository.
- R2: The satellite management system shall provide users with the ability to transfer maintenance and service plans to the user help desk.
- R3: The mission operation controller shall transmit any detected anomalies with the user help desk.

# Challenge #2

## Capture Change Rationale

- R1: The mission operation controller shall transmit satellite status reports to the user help desk document repository.
- R2: The satellite management system shall provide users with the ability to transfer maintenance and service plans to the user help desk.
- R3: The mission operation controller shall transmit any detected anomalies with the user help desk.

# Challenge #2

## Change Rationale

- R1: The mission operation controller shall **transmit** satellite status reports to the user help desk document repository.
- R2: The satellite management system shall provide users with the ability to transfer maintenance and service plans to the user help desk.
- R3: The mission operation controller shall **transmit** any detected anomalies with the user help desk.

### Possible Rationales:

- 1: We want to globally rename “user help desk”**
- 2: Avoid communication between “mission operation controller” and “user help desk” (R3)**
- 3: We no longer want to “transmit satellite status reports” to “user help desk” but instead to “user document repository” (only R1)**

**Determine conditions under which change should propagate**

# Solution Characteristics

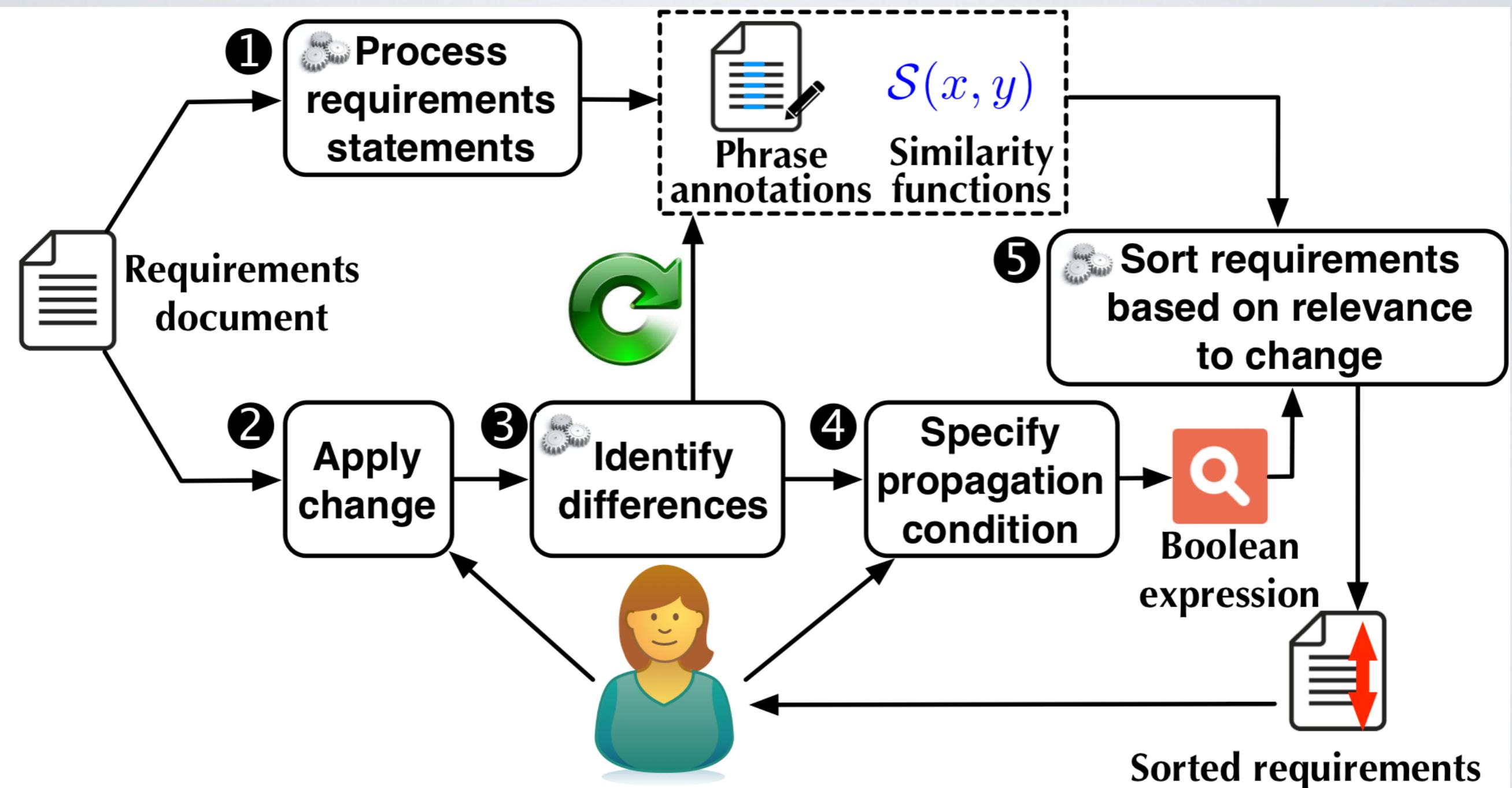
- **Account for the phrasal structure of requirements**

The mission operation controller shall transmit satellite status reports to the user ~~help desk~~ document repository.

user help desk, **Deleted**  
user document repository, **Added**

- Account for **change rationale** expressed by user (propagation condition)
- Consider **semantically-related phrases** that are not exact matches and **close syntactic variations** across requirements: quantitative matching of condition

# Narcia



# Evaluation



1 case study



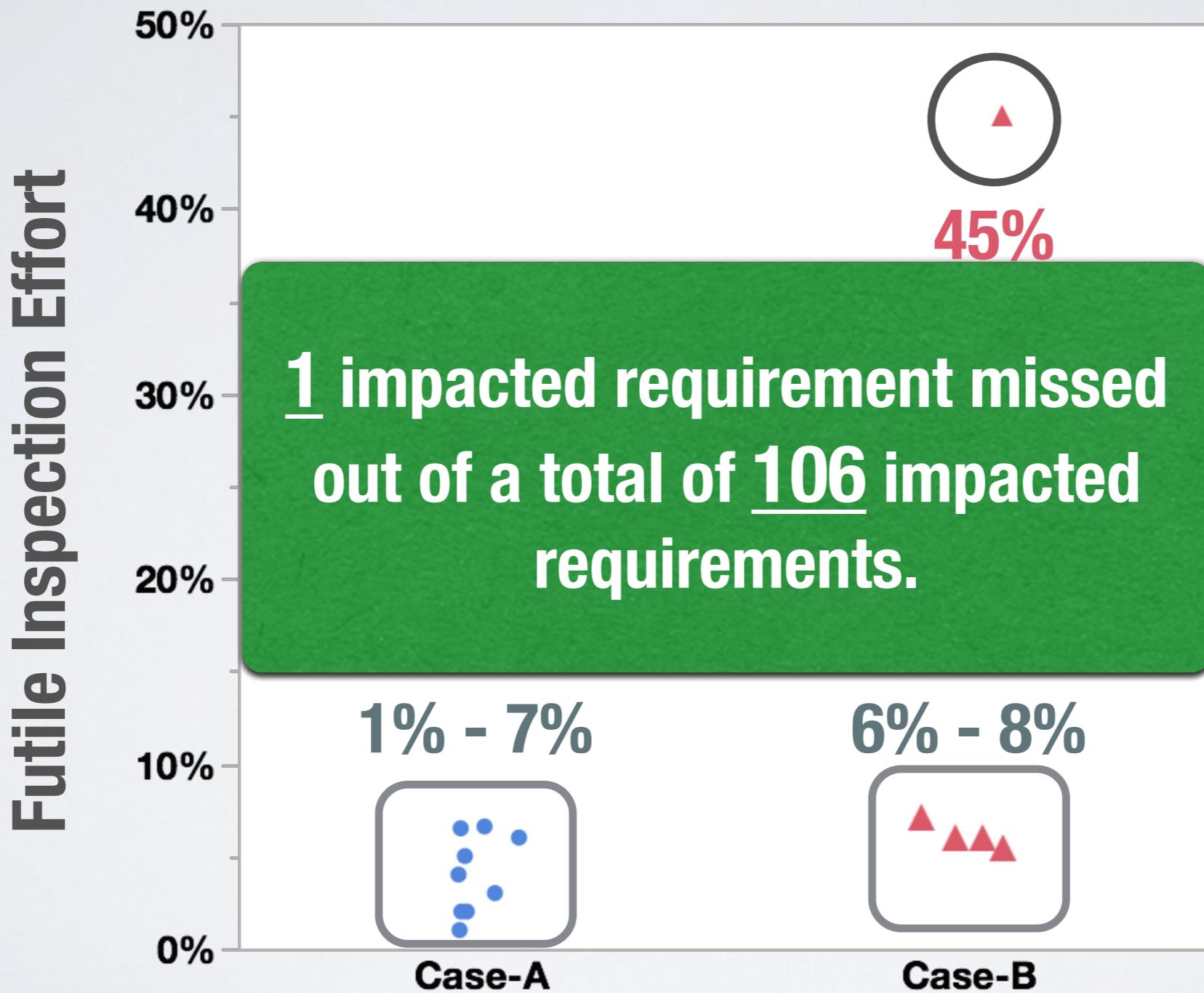
158 Requirements  
9 change scenarios

1 case study

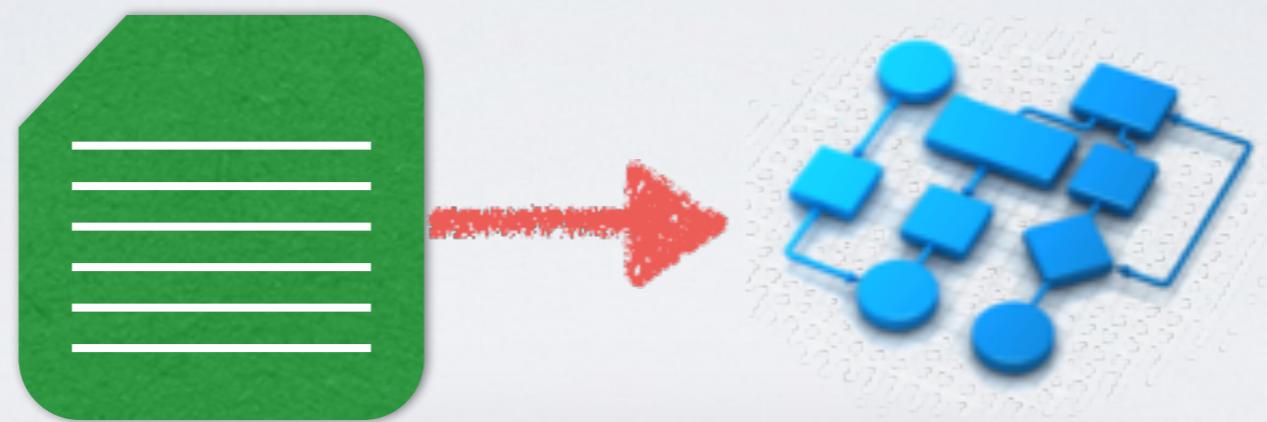


72 Requirements  
5 Change  
Scenarios

# Effectiveness of Our Approach



# Requirements to Design

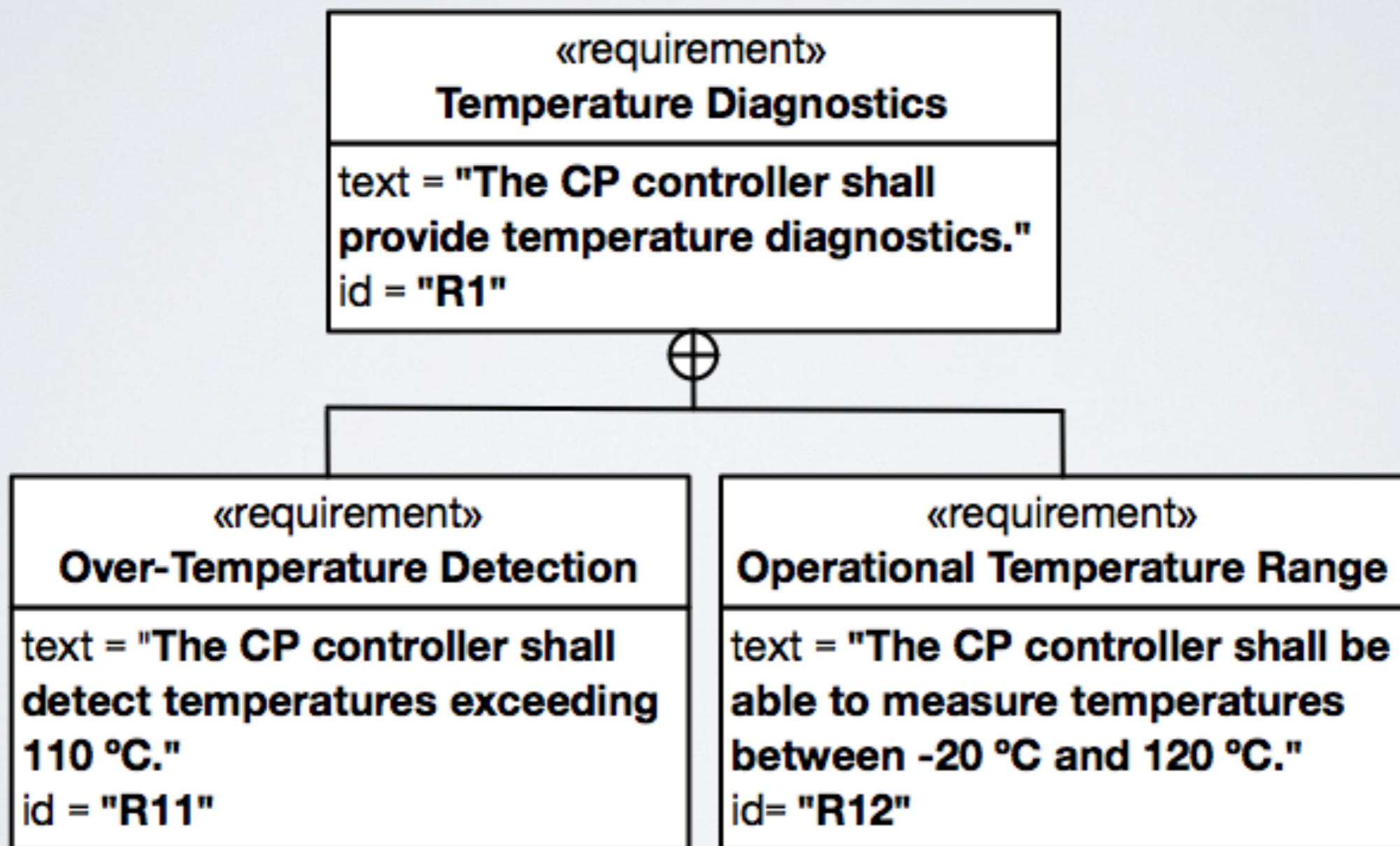


**Requirements-to-Design  
Change Impact Analysis**

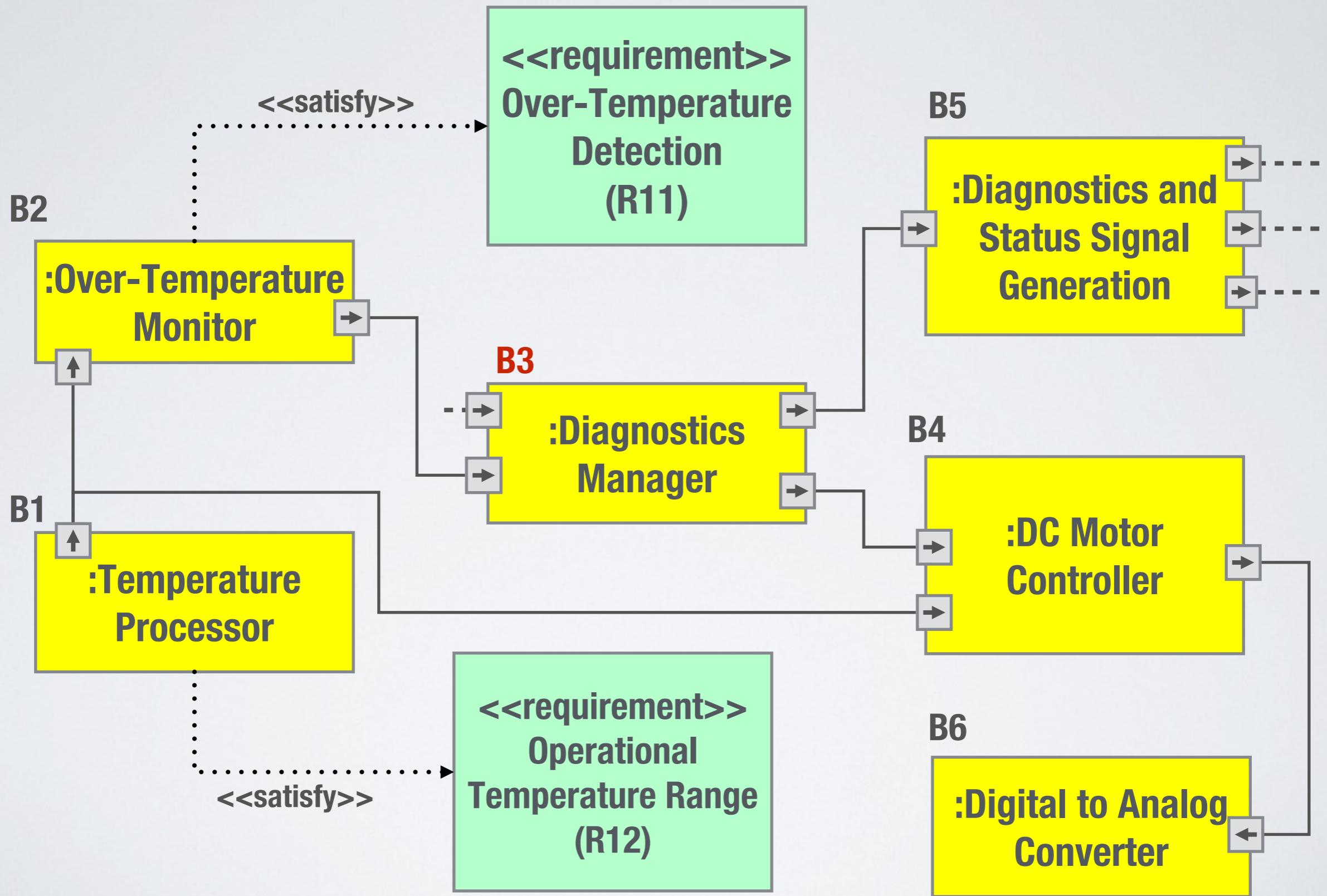
# Motivations

- **Rigorous change management** required by many standards and customers in safety critical systems, and embedded systems in general in many industry sectors
- Impact of requirements changes on **design decisions**
- **Complete and precise design impact set**
- **SysML** commonly used as embedded and cyber-physical system design representation

# Requirements Diagram

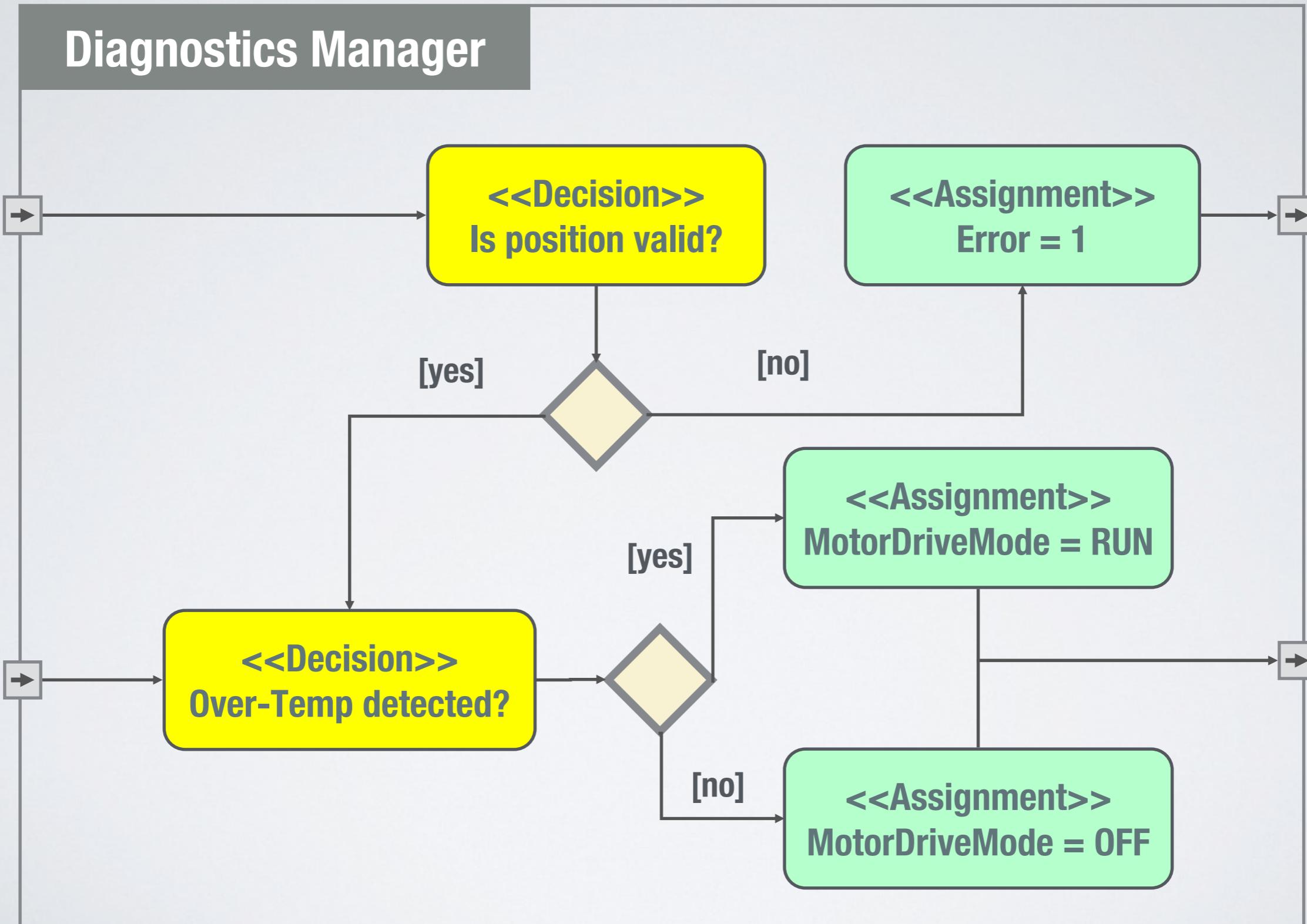


# Structural Diagram

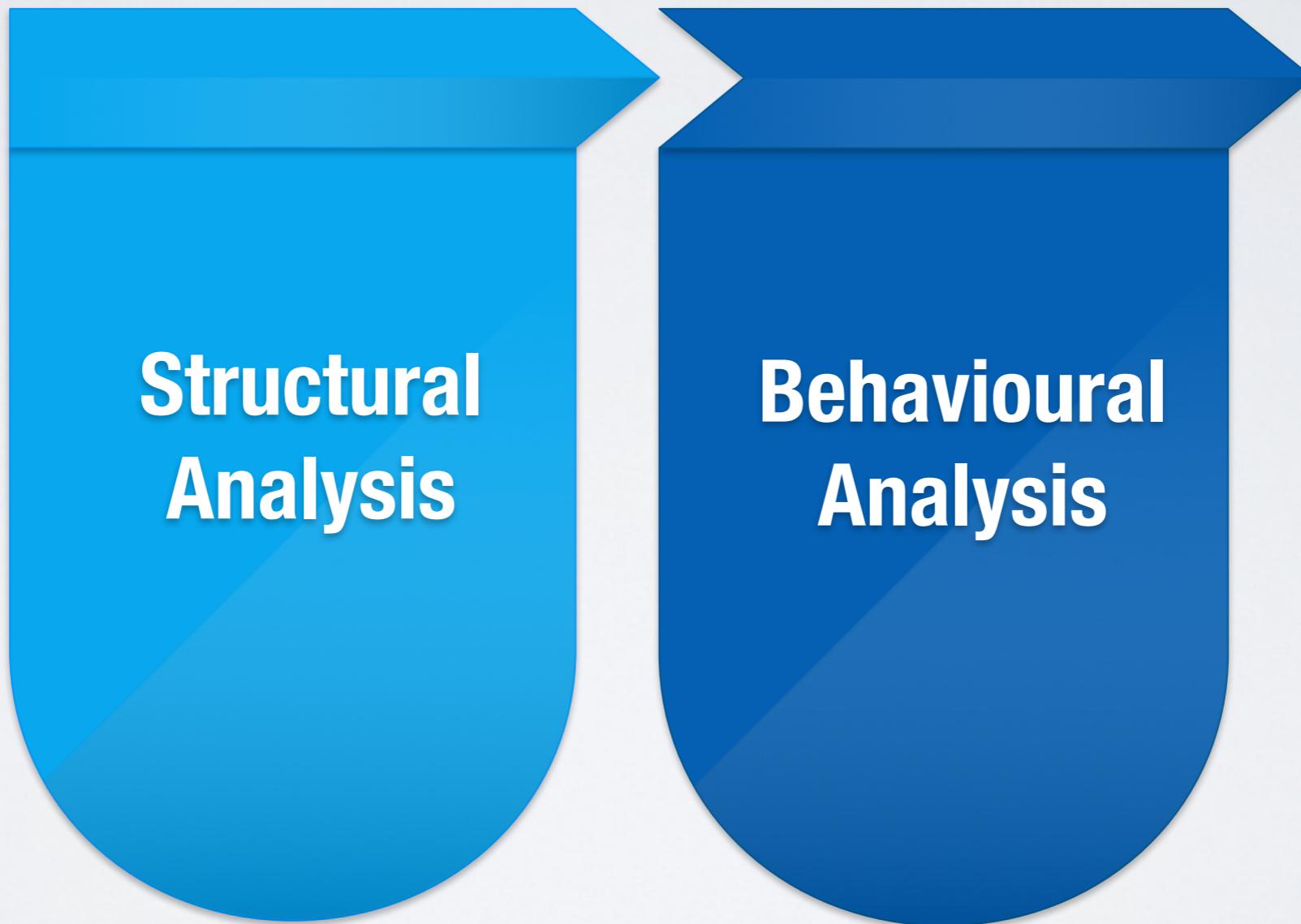


# Behavioural Diagram

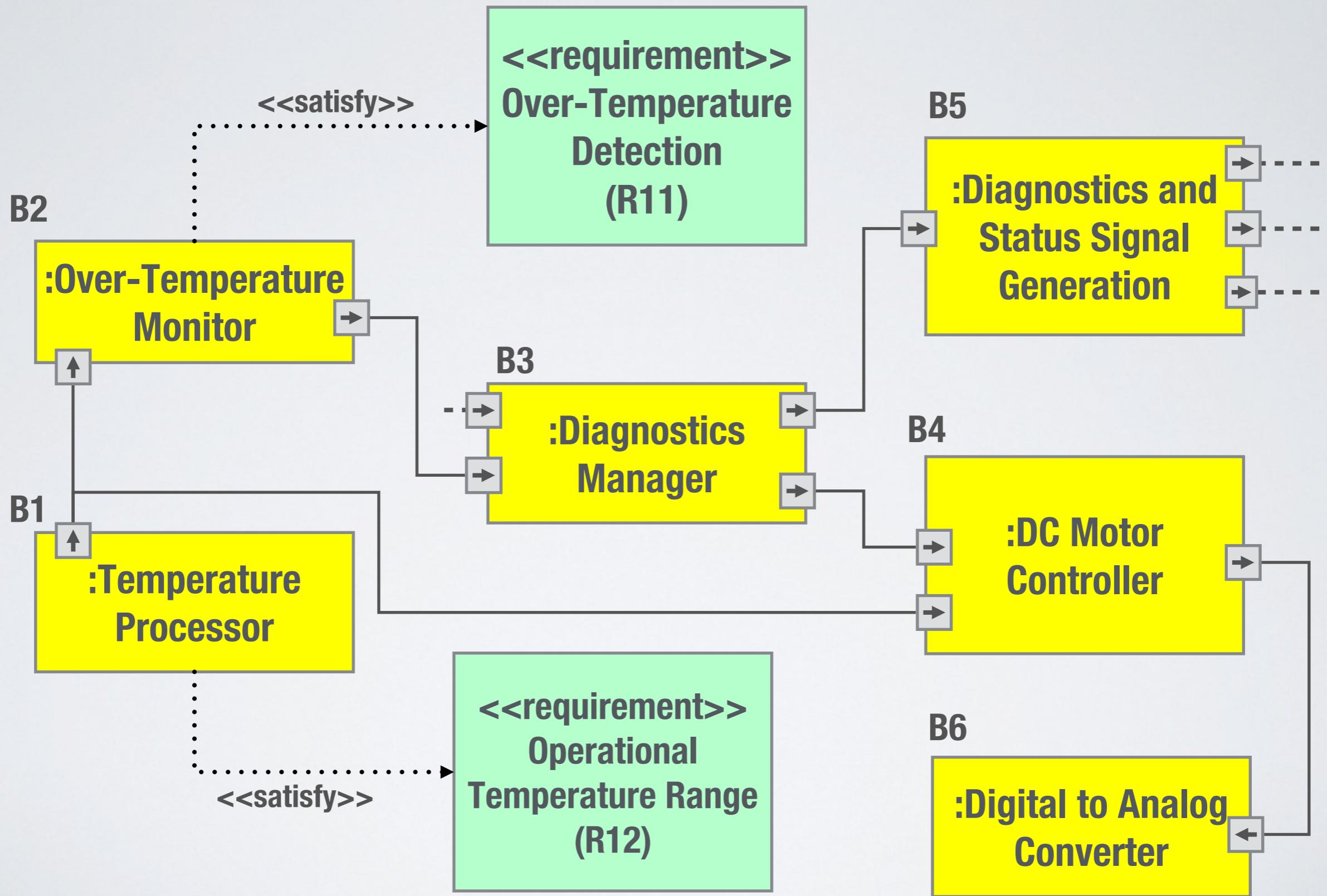
B3



# Compute Impacted Elements



# Structural Diagram

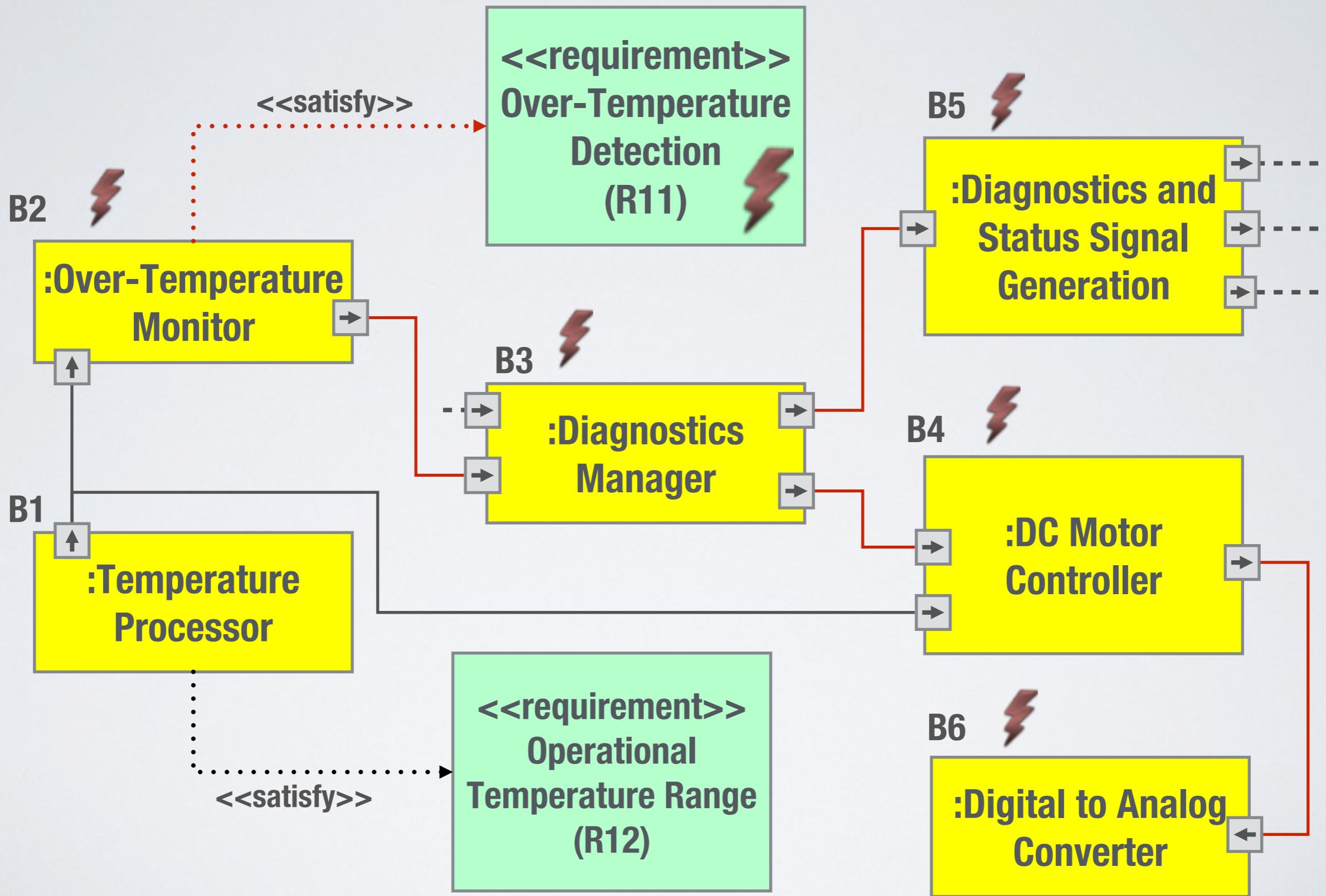


# Structural Diagram

<<requirement>>

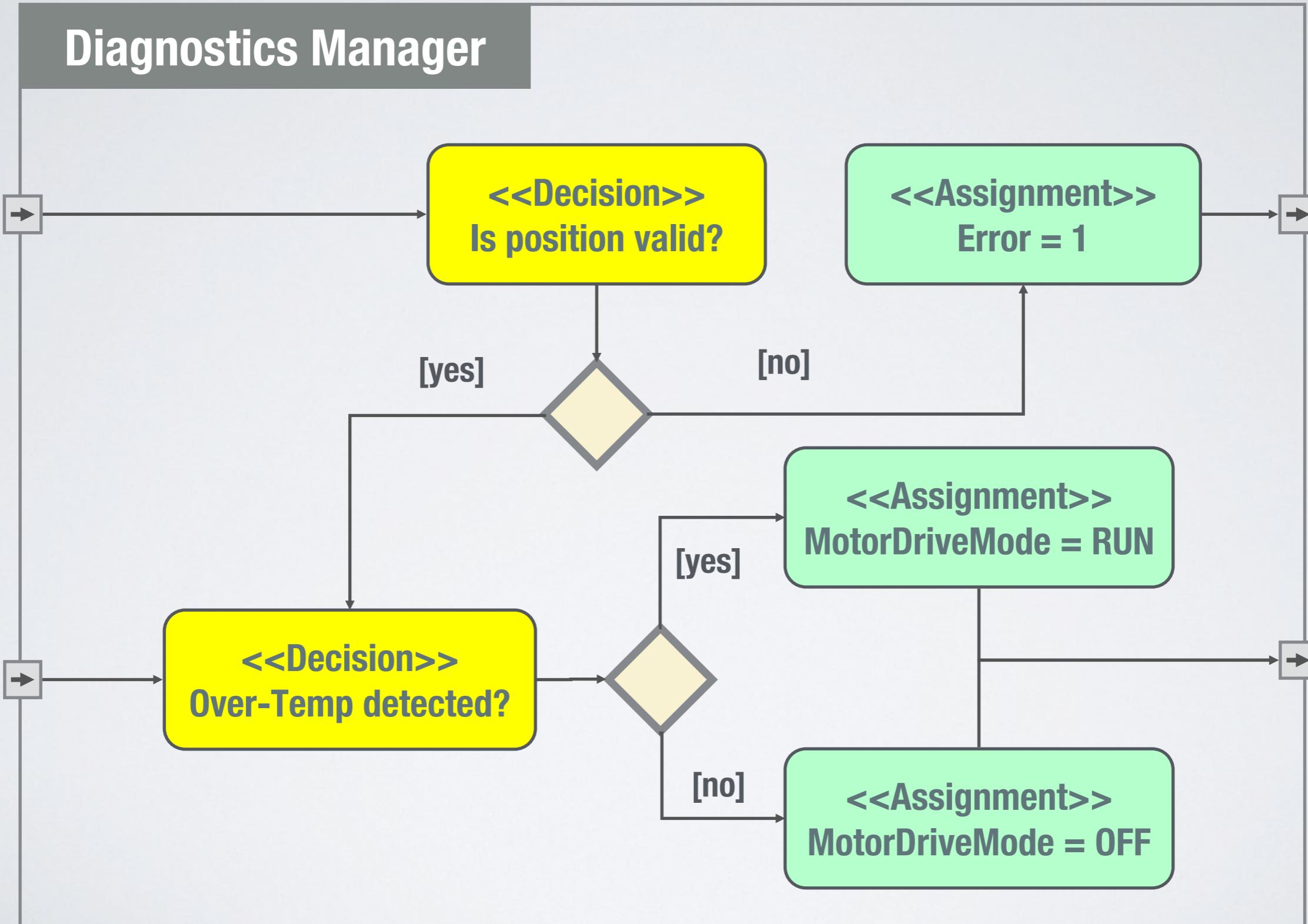
**Change to R11: Change over temperature detection level to 147 C  
from 110 C.**

# Structural Diagram



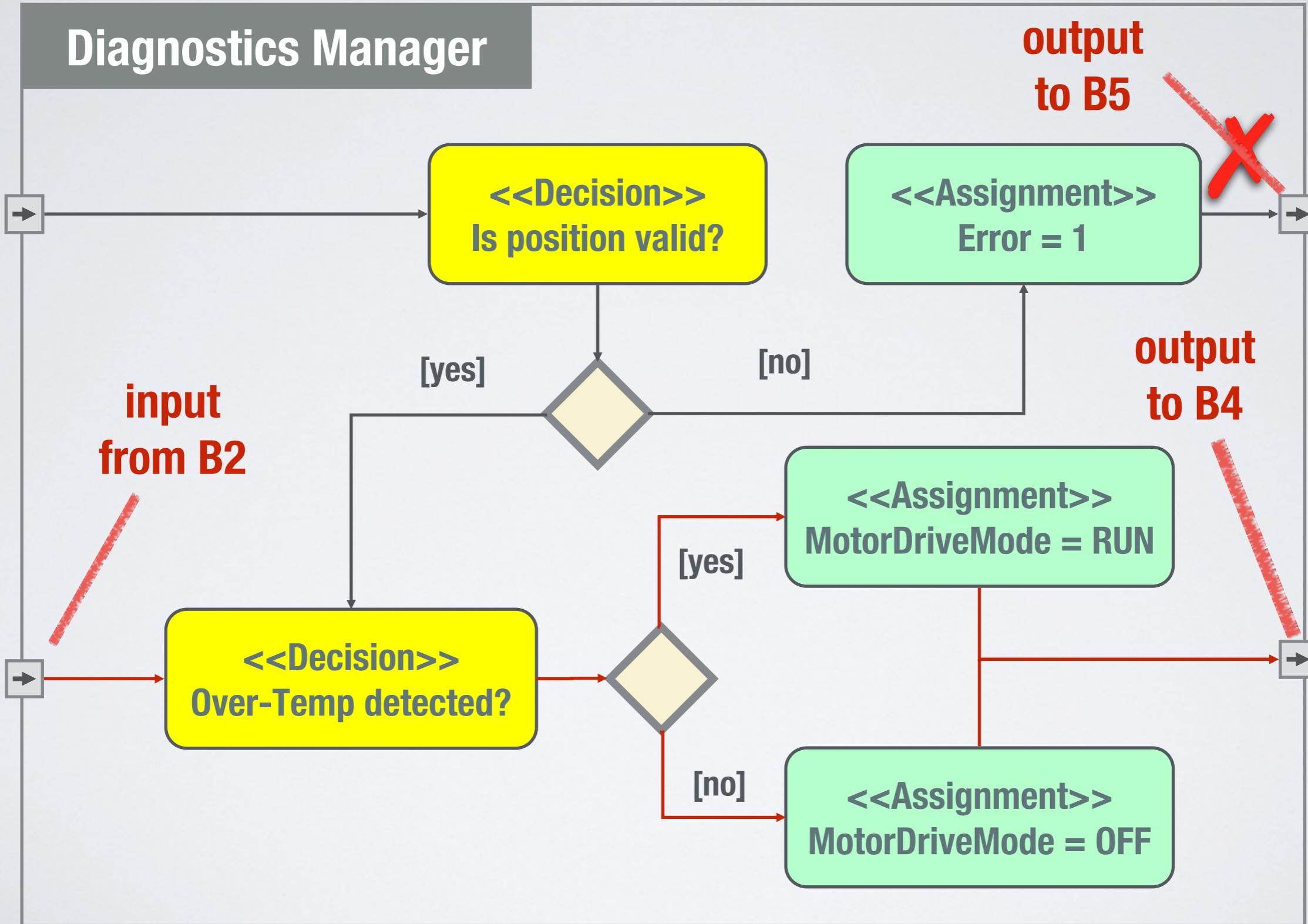
# Behavioural Diagram

B3

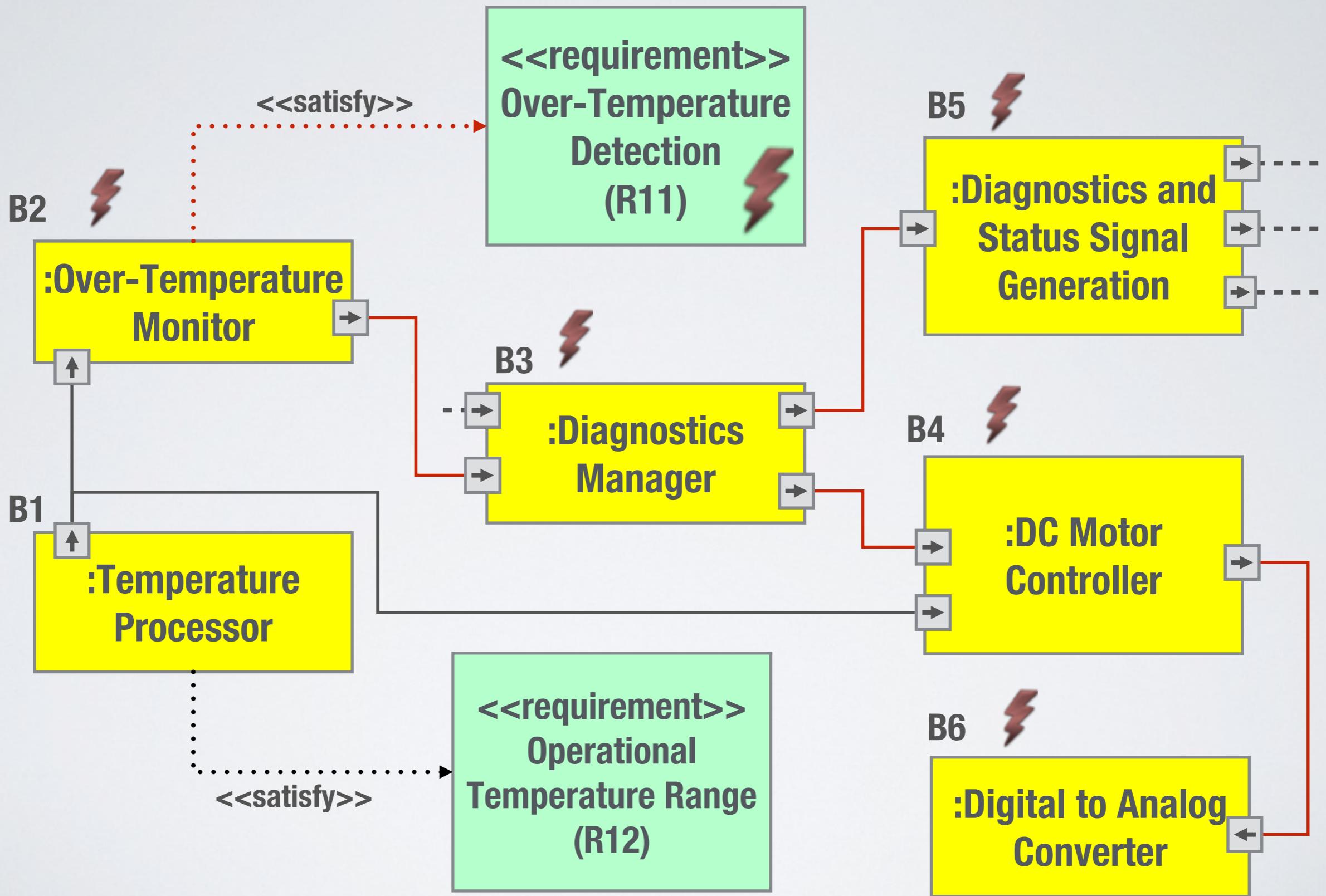


# Behavioural Diagram

B3



# Structural Diagram



# Rank Elements

**Change to R11: Change over temperature detection level to 147 C from 110 C.**

**B2, B3, B4, B6**

**Natural Language Processing Analysis**

**B2  
B6  
B3  
B4**

**Ranked according to likelihood of impact**

# Change Statements

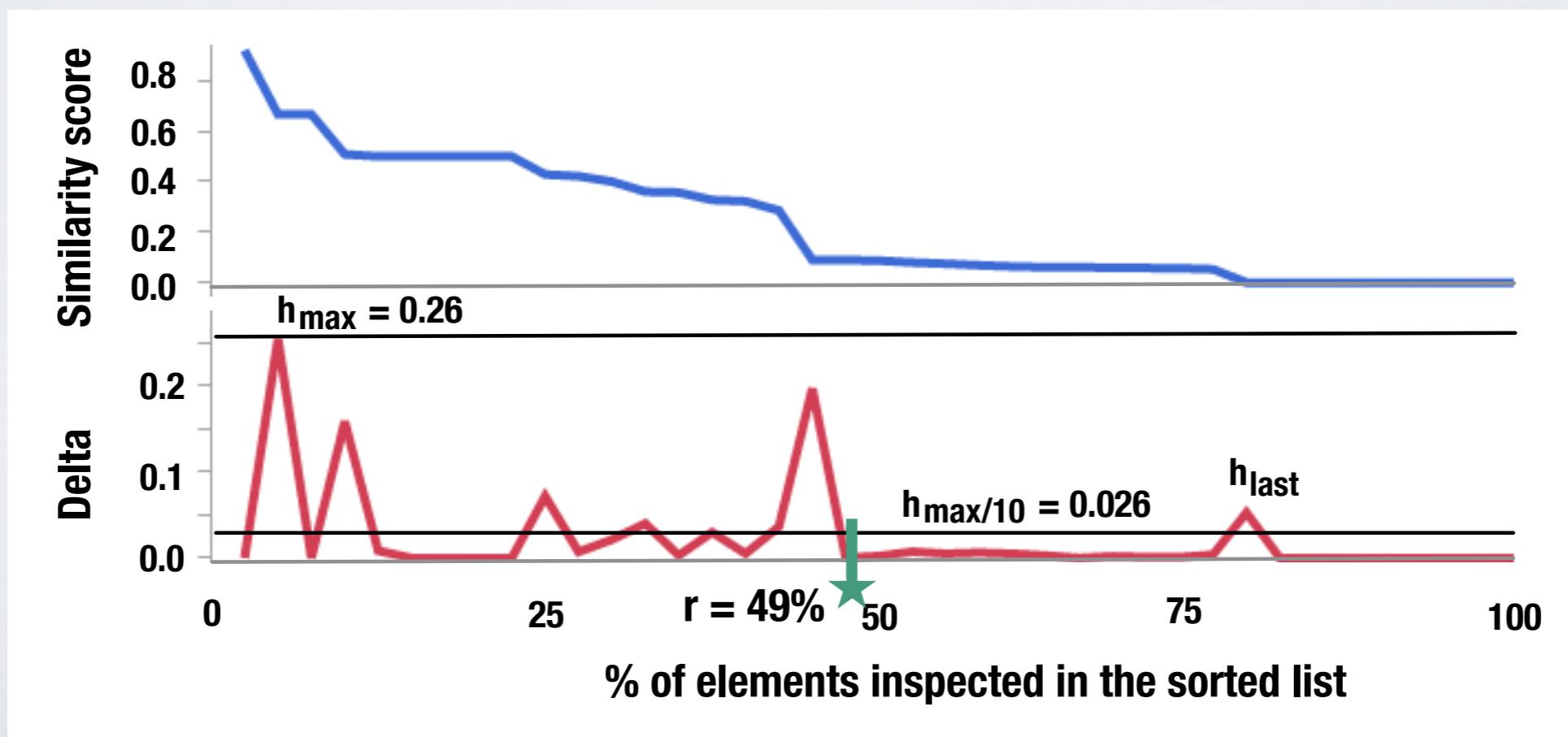
- **Informal inputs from systems engineers regarding impact of changes**
- **Example:** “Temperature lookup tables and voltage converters need to be adjusted”

# Natural Language Processing

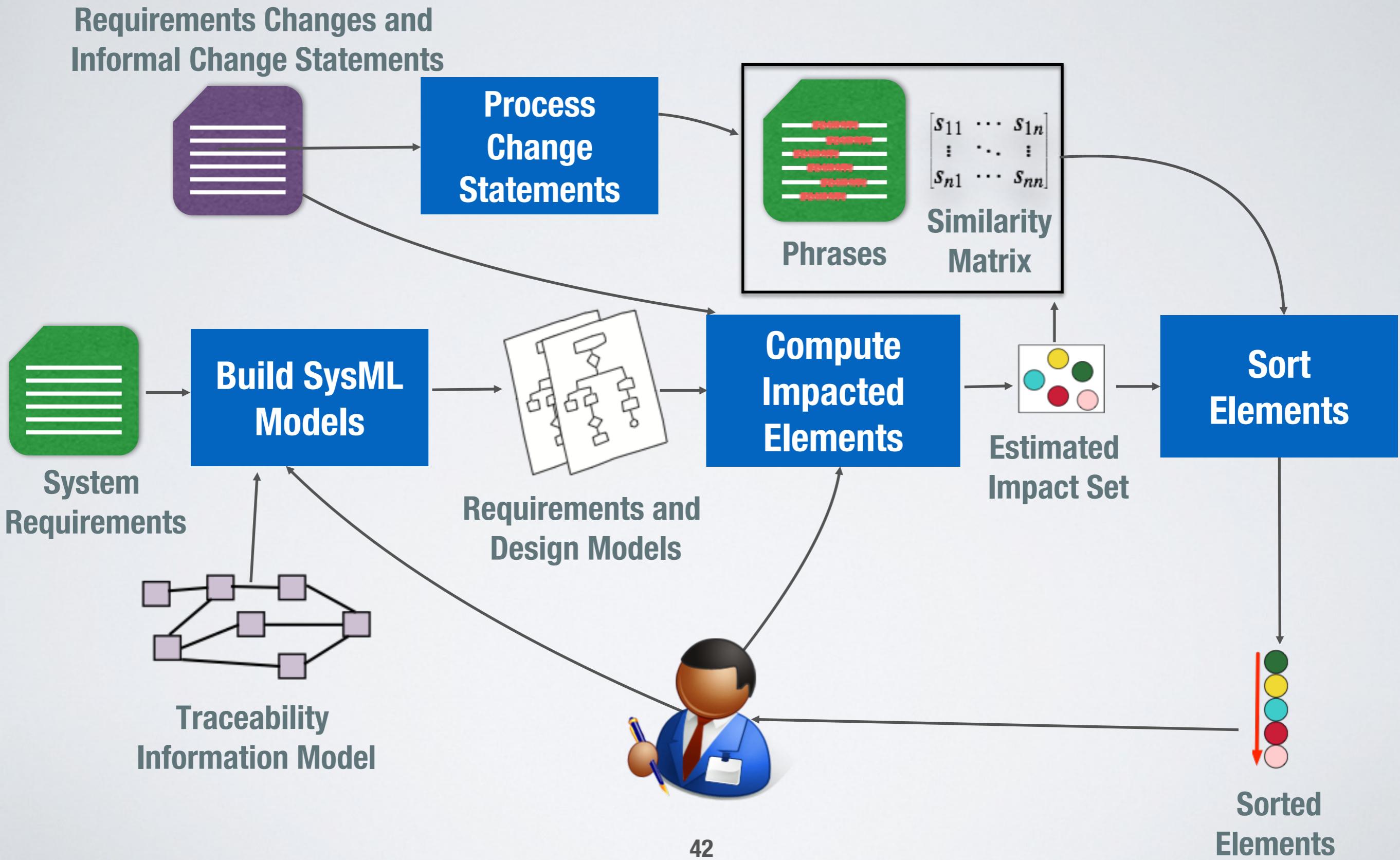
- Compute **similarity scores** (syntactic and semantic) **between model elements labels and change statements**
- Experimented with **pairwise combinations** of syntactic and semantic measures
- Sort the design elements obtained after structural and behavioral analysis based on the similarity scores
- Engineers inspect the **sorted lists** to identify impacted elements

# Identifying a Subset to Inspect

- Pick the last significant peak in delta similarity between two successive elements
- Point beyond which the similarity scores can no longer adequately tell apart the elements



# Overview

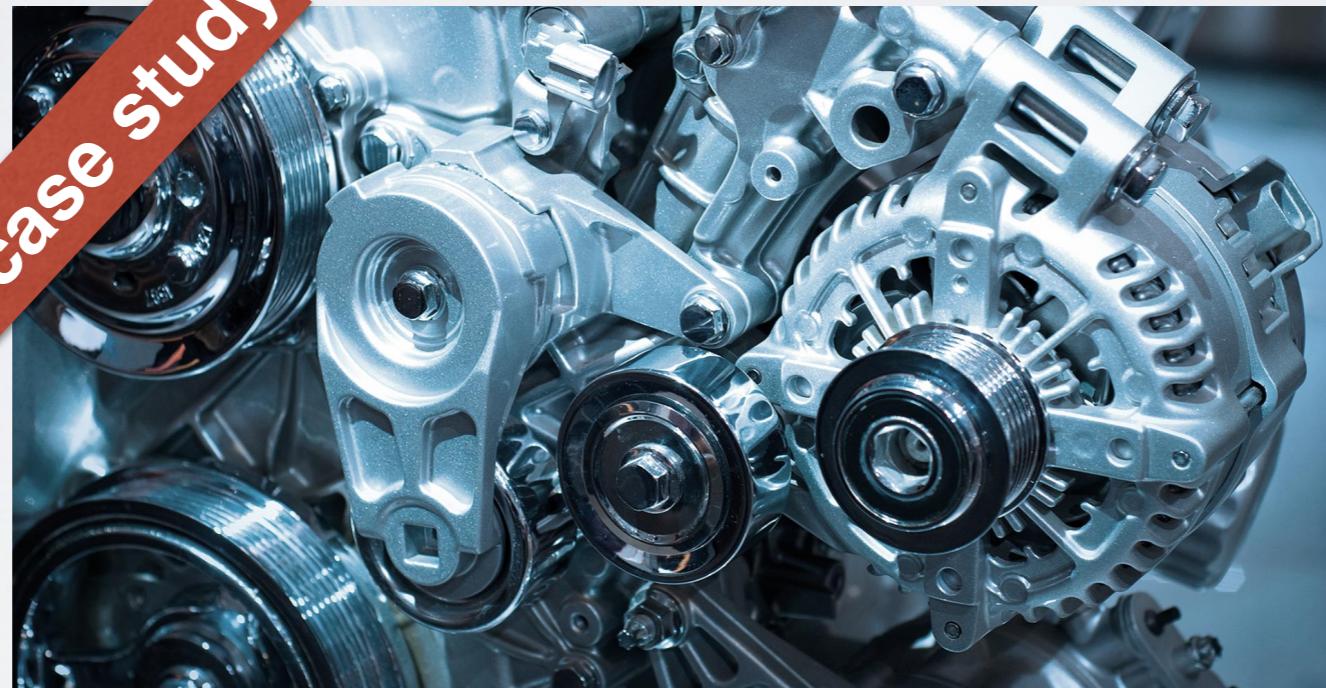


# Evaluation

**DELPHI**

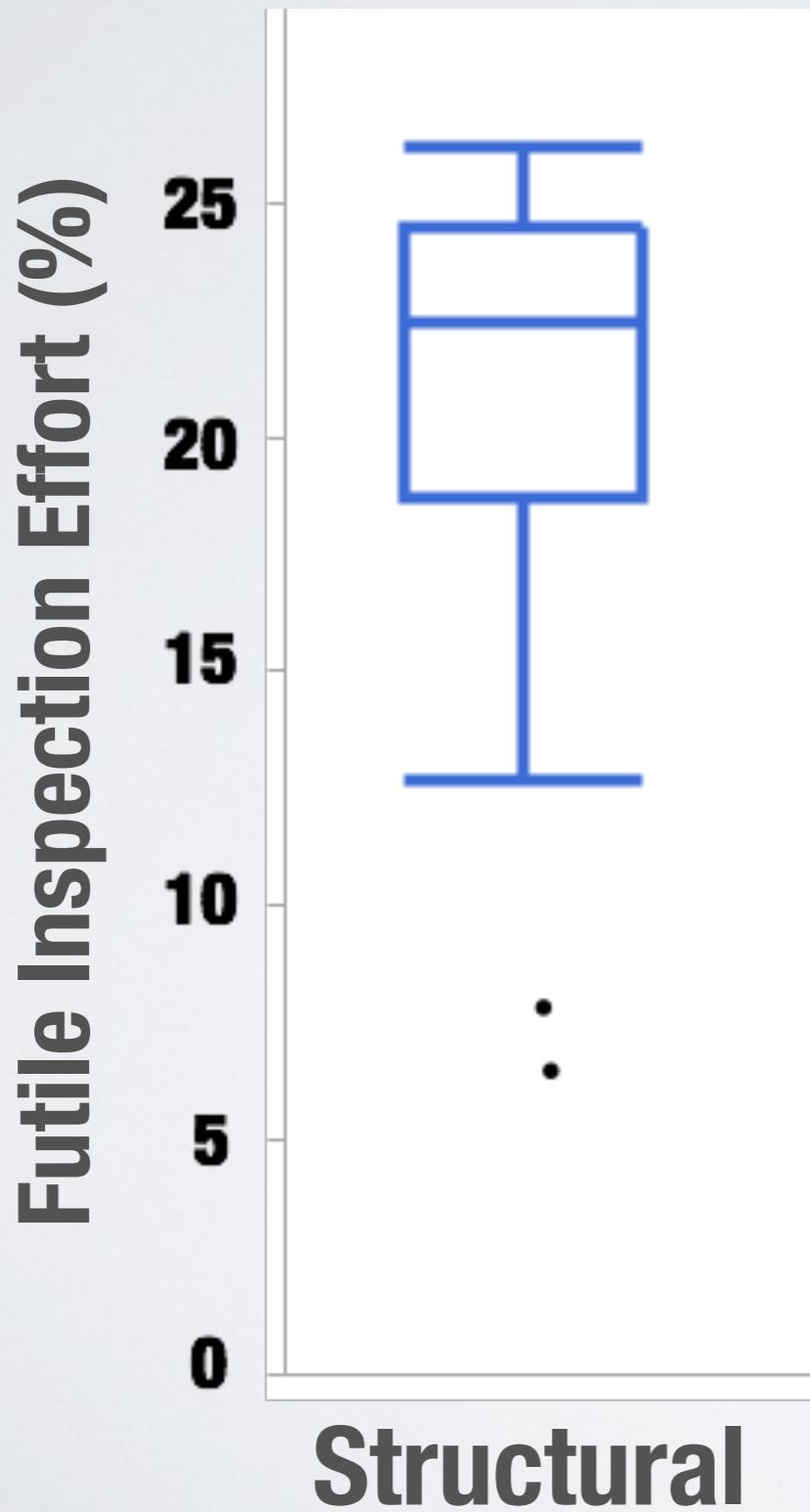
Innovation for the Real World

1 Case Study

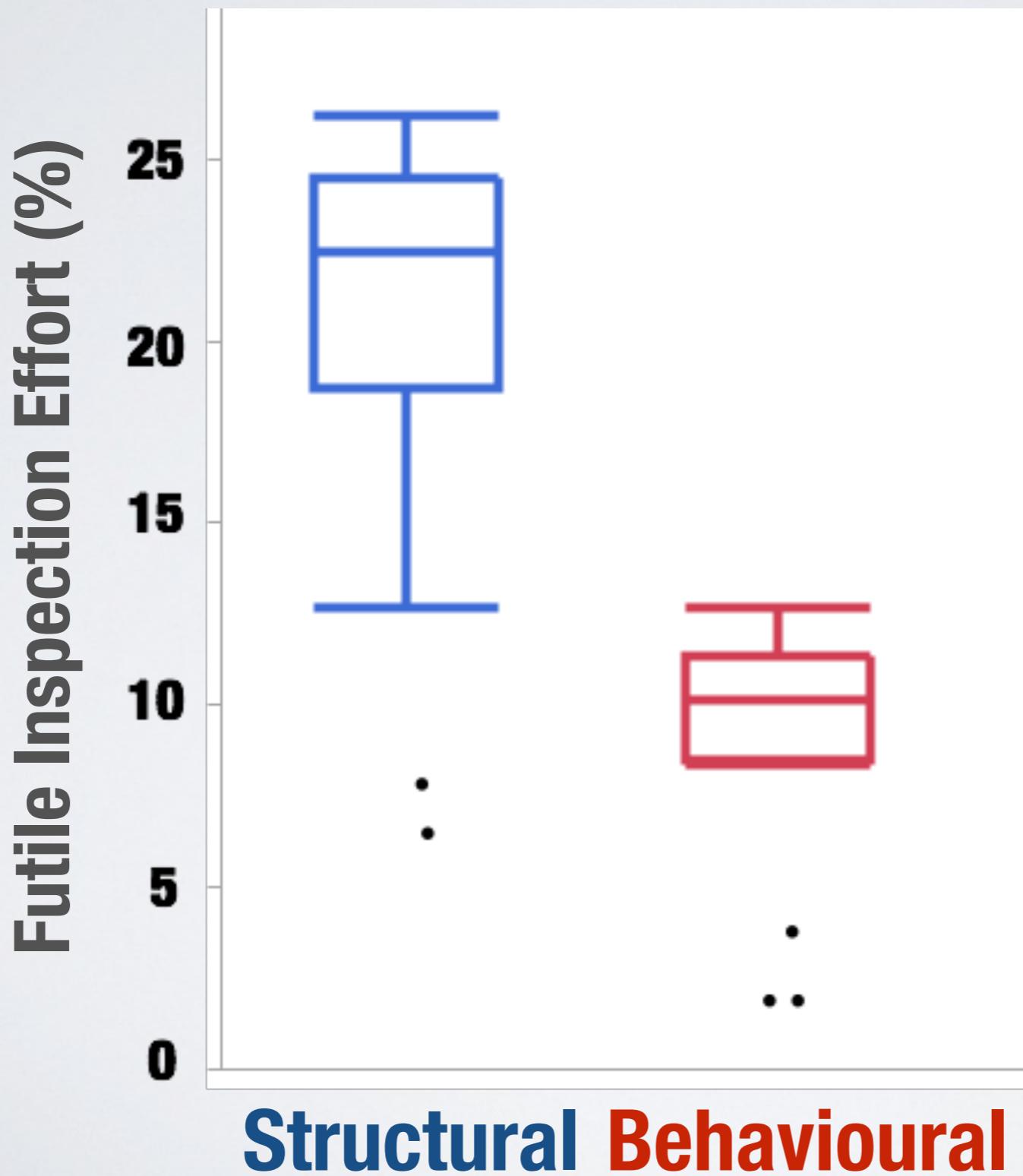


370 elements  
16 change scenarios

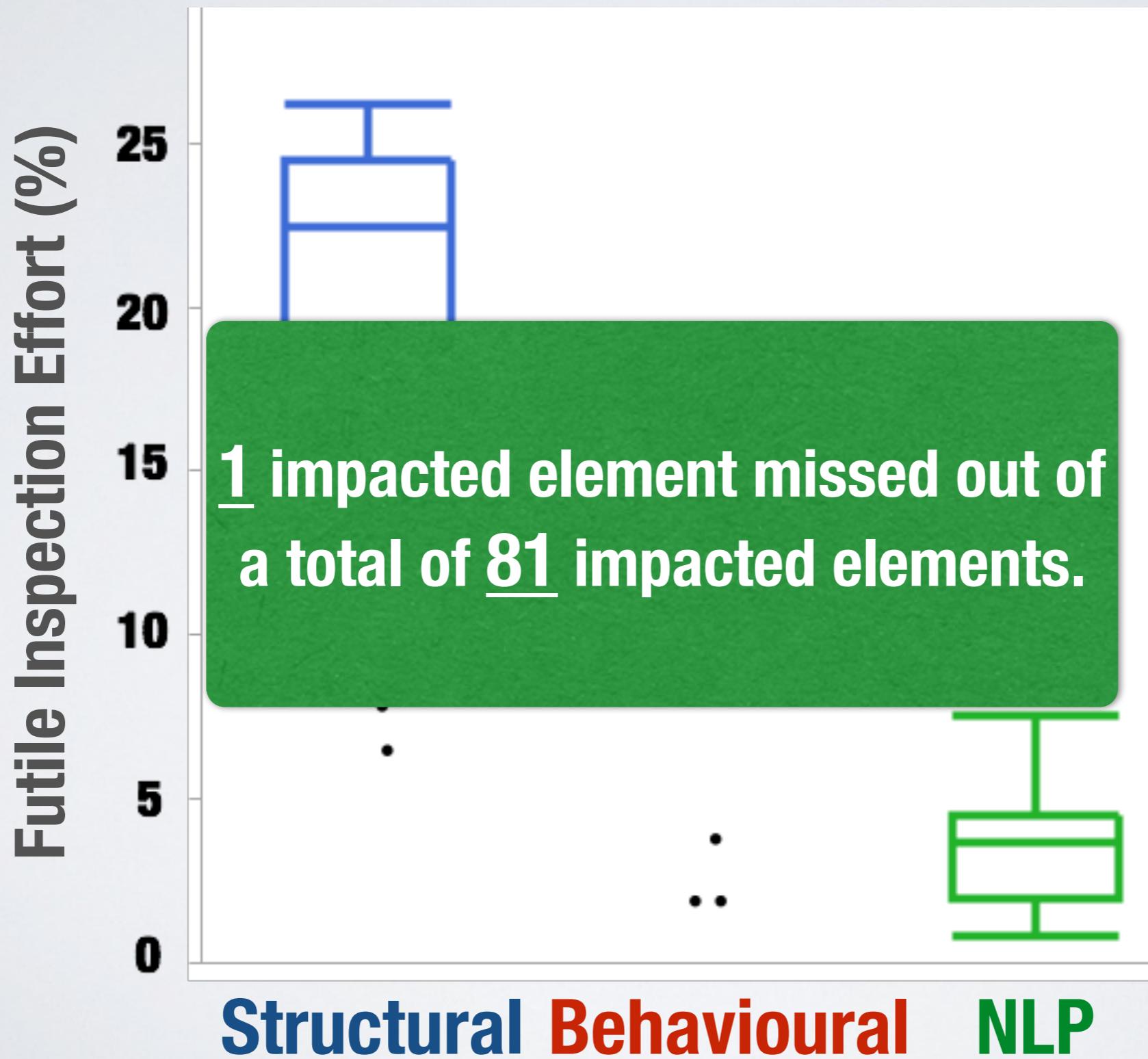
# Effectiveness of Our Approach



# Effectiveness of Our Approach



# Effectiveness of Our Approach



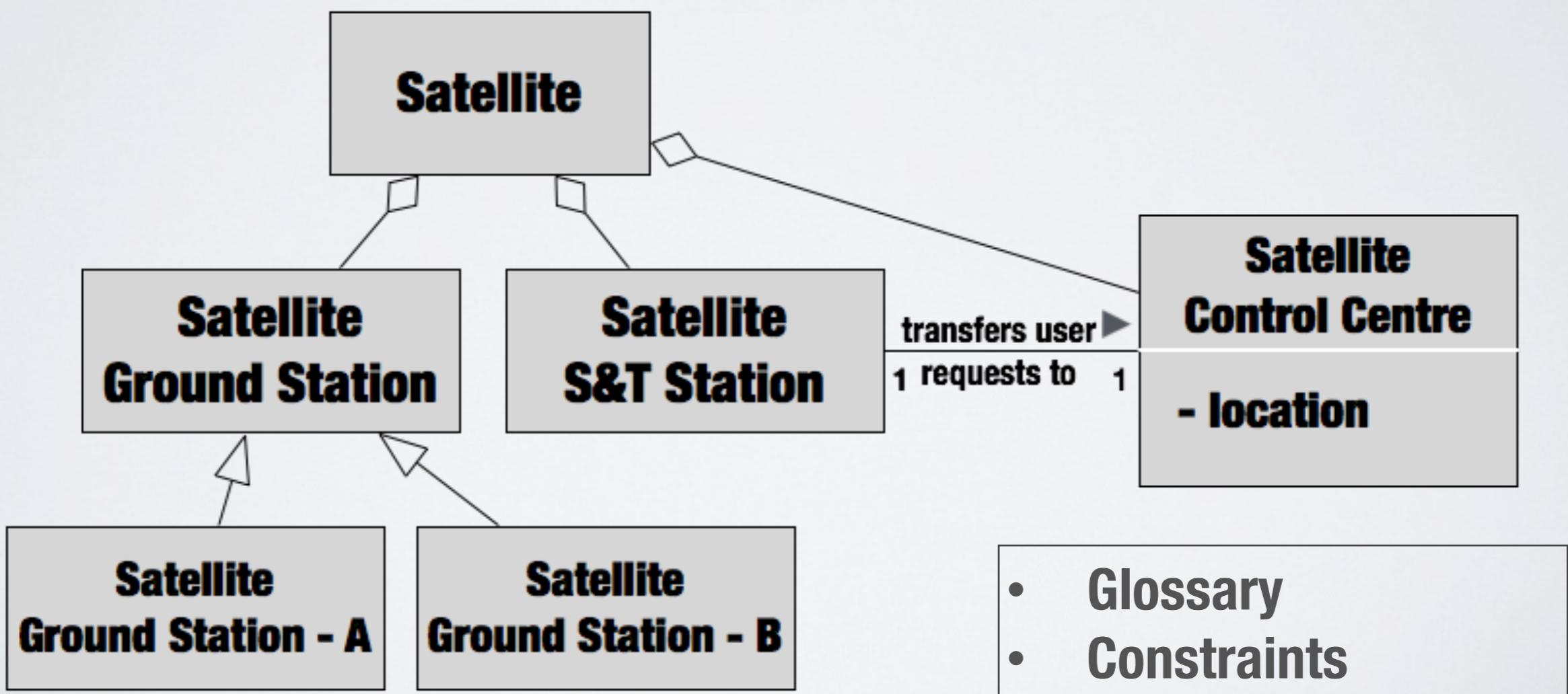
# Extracting Domain Knowledge

# Domain Knowledge

- All requirements depend, more or less explicitly, on domain knowledge
- Domain-specific concepts and terminology
- In practice: Not always consistent among all stakeholders
- Software engineers often have a superficial understanding of the application domain they target
- Extracting domain knowledge from requirements: Glossary, domain model ...

# Domain Models

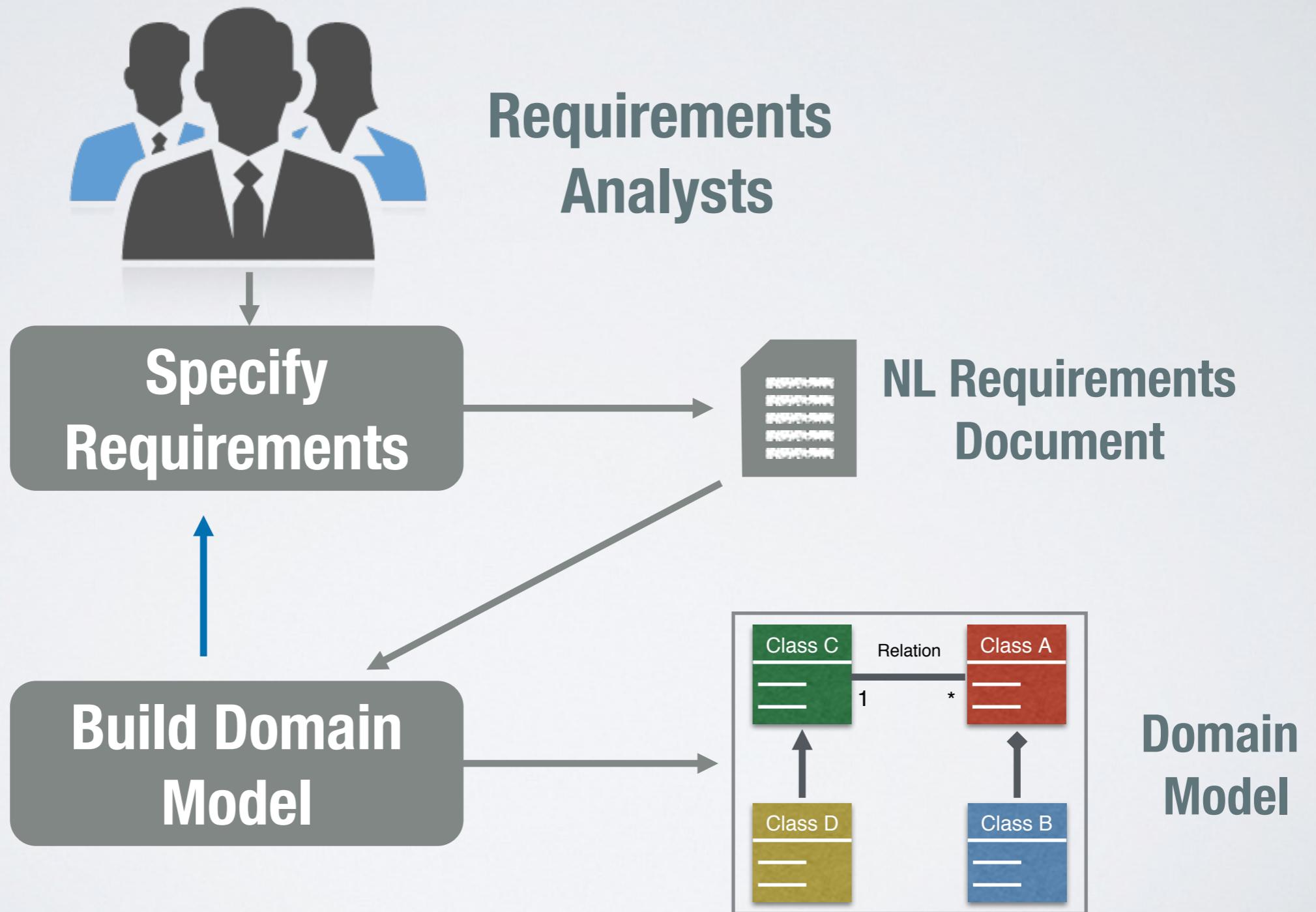
A domain model is a representation of conceptual entities or real-world objects in a domain of interest



# Motivation

- Representation of important **domain concepts and their relations**
  - **Facilitate communication** between stakeholders from different backgrounds
  - Help identify **inconsistencies** in terminology, etc.
  - Helps assess **completeness** of requirements
- In practice, domain models **are not preceding the elicitation and writing of requirements**

# Context



# Problem Definition

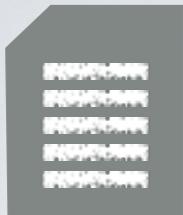


- Manually building domain models is **laborious**
- **Automated support is required** for building domain models

# State of the Art

- **Multiple approaches exist** for extracting domain models or similar variants from requirements using **extraction rules**
  - Majority assume **specific structure**, e.g., restricted NL
  - Extraction of **direct relations only** but not indirect ones
  - **Limited empirical results** on industrial requirements

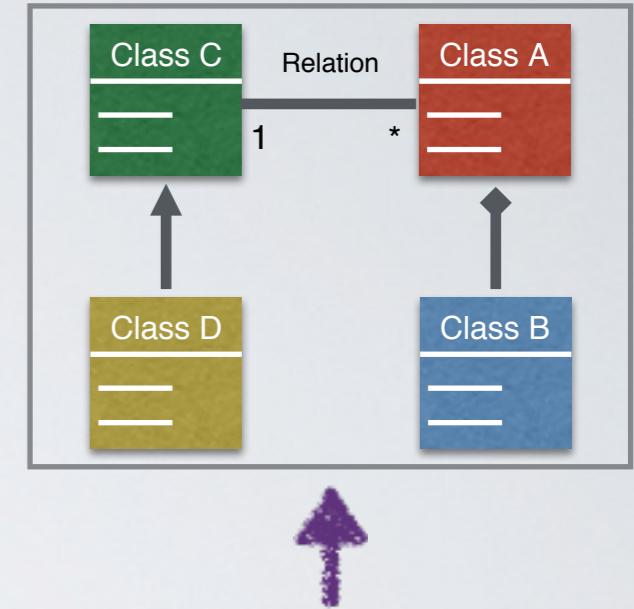
# Approach



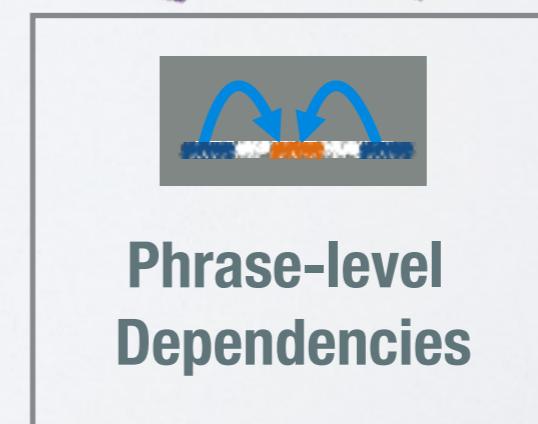
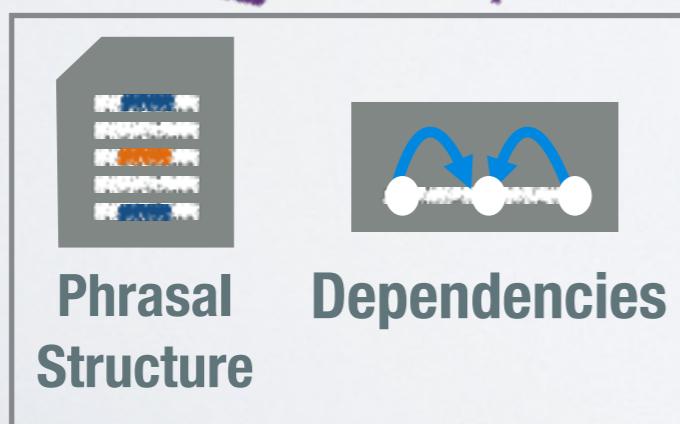
NL  
Requirements

Process  
Requirements  
Statements

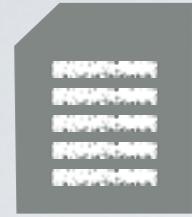
Lift  
Dependencies to  
Semantic Units



Construct  
Domain Model



# Approach

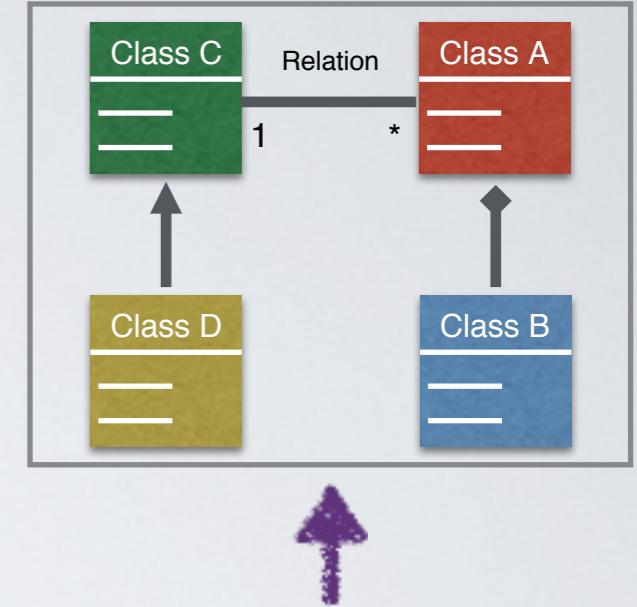


NL  
Requirements



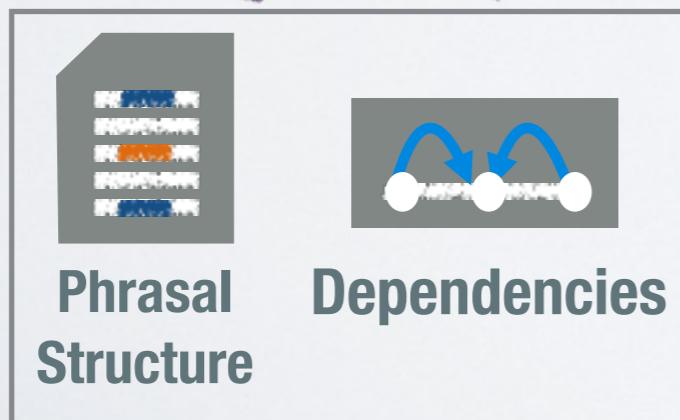
Process  
Requirements  
Statements

Lift  
Dependencies to  
Semantic Units



Domain  
Model

Construct  
Domain Model



Phrasal  
Structure

Dependencies

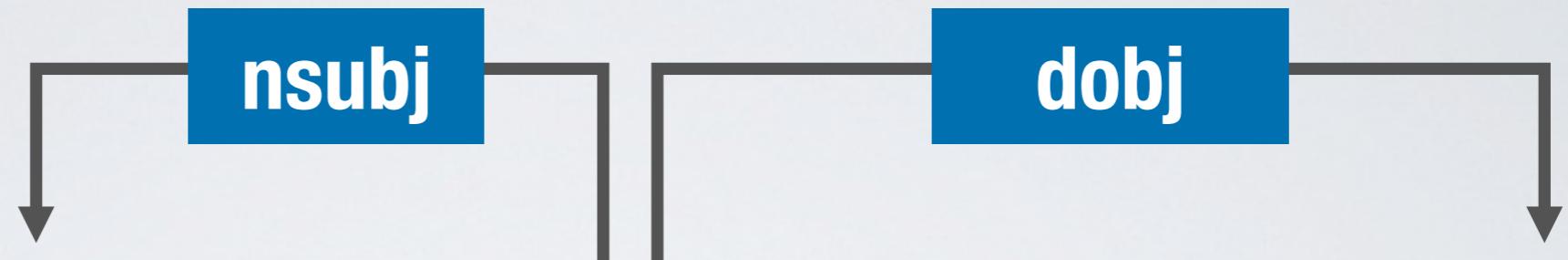


Phrase-level  
Dependencies

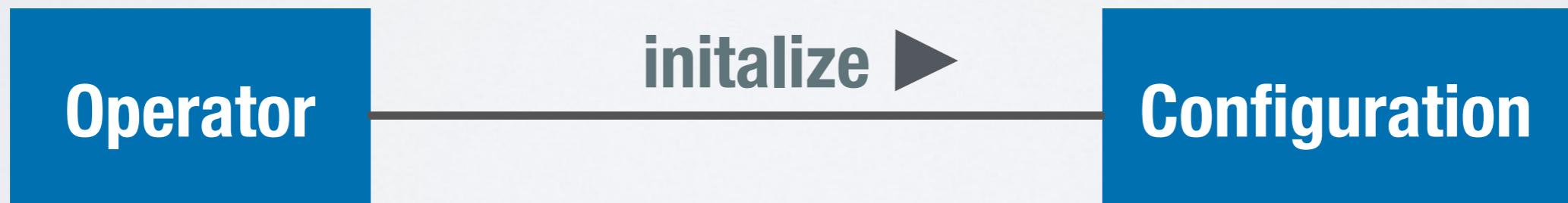


Extraction  
Rules

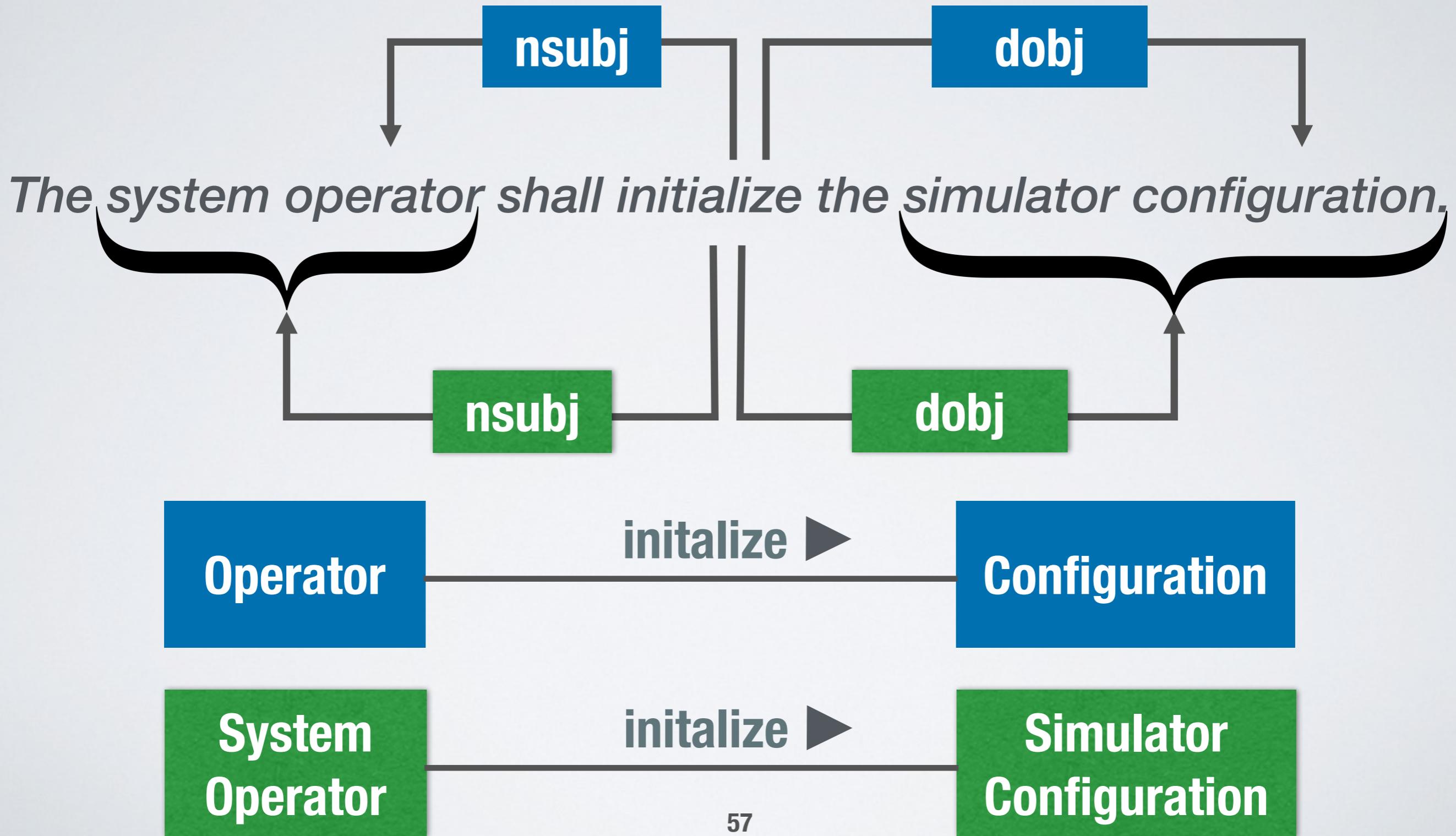
# Grammatical Dependencies



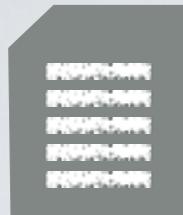
*The system operator shall initialize the simulator configuration.*



# Lift Dependencies to Semantic Units



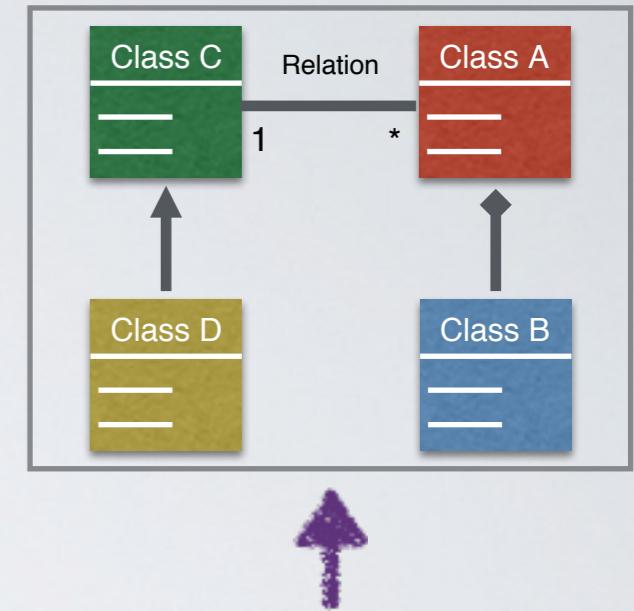
# Approach



NL  
Requirements

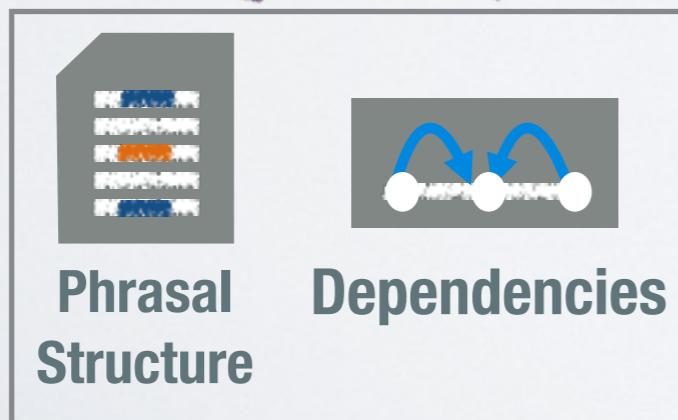
Process  
Requirements  
Statements

Lift  
Dependencies to  
Semantic Units



Domain  
Model

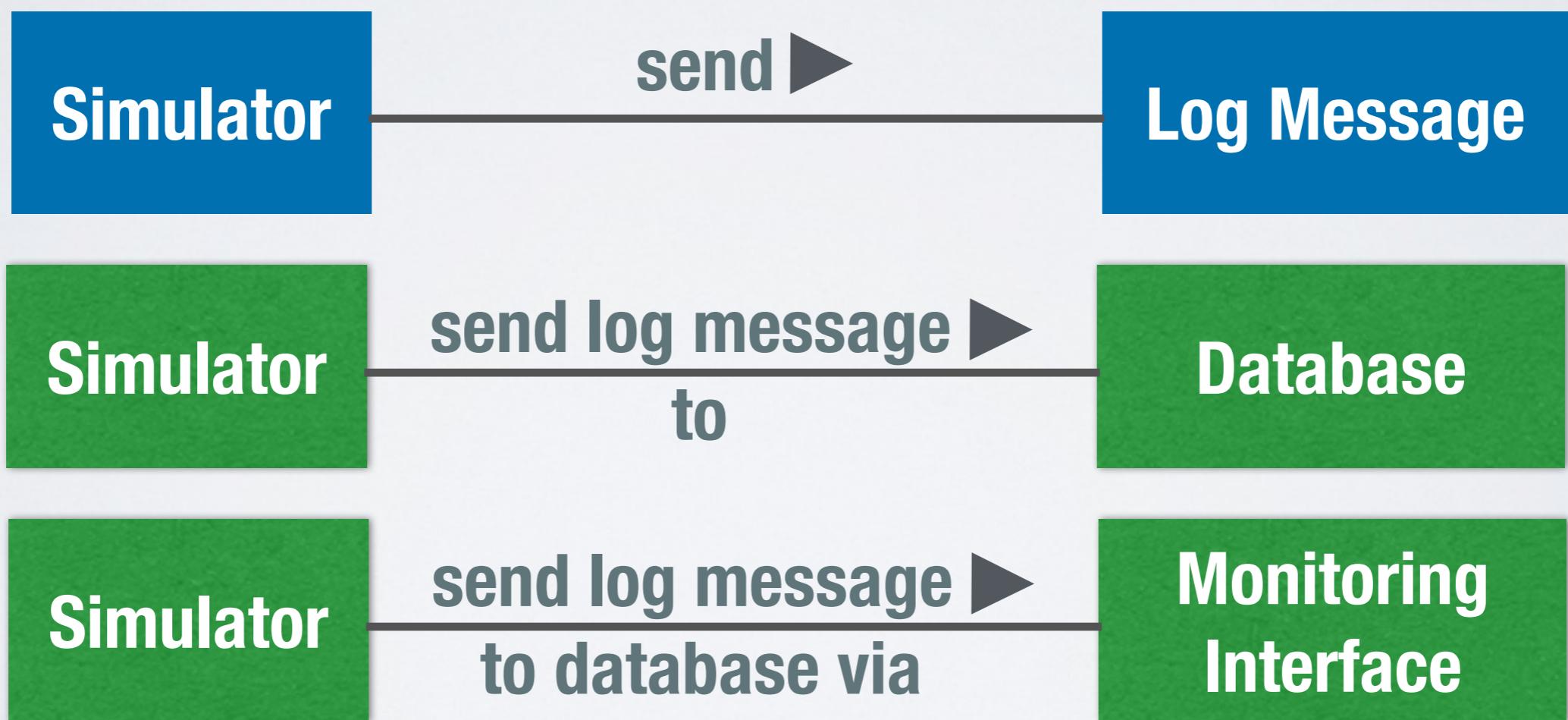
Construct  
Domain Model



23 Extraction  
Rules

# Link Paths – Indirect Relations

*The simulator shall send log messages to the database via the monitoring interface.*



# How useful is our approach?

1 case study



50 Requirements  
213 Relations

**SES<sup>▲</sup>**

your satellite company

- Interview survey with experts
- Correctness and Relevance of each relation
- Missing relations in each requirement

# Results

**Correctness: 88% (avg.)**

**Relevance: 37% (avg.)**

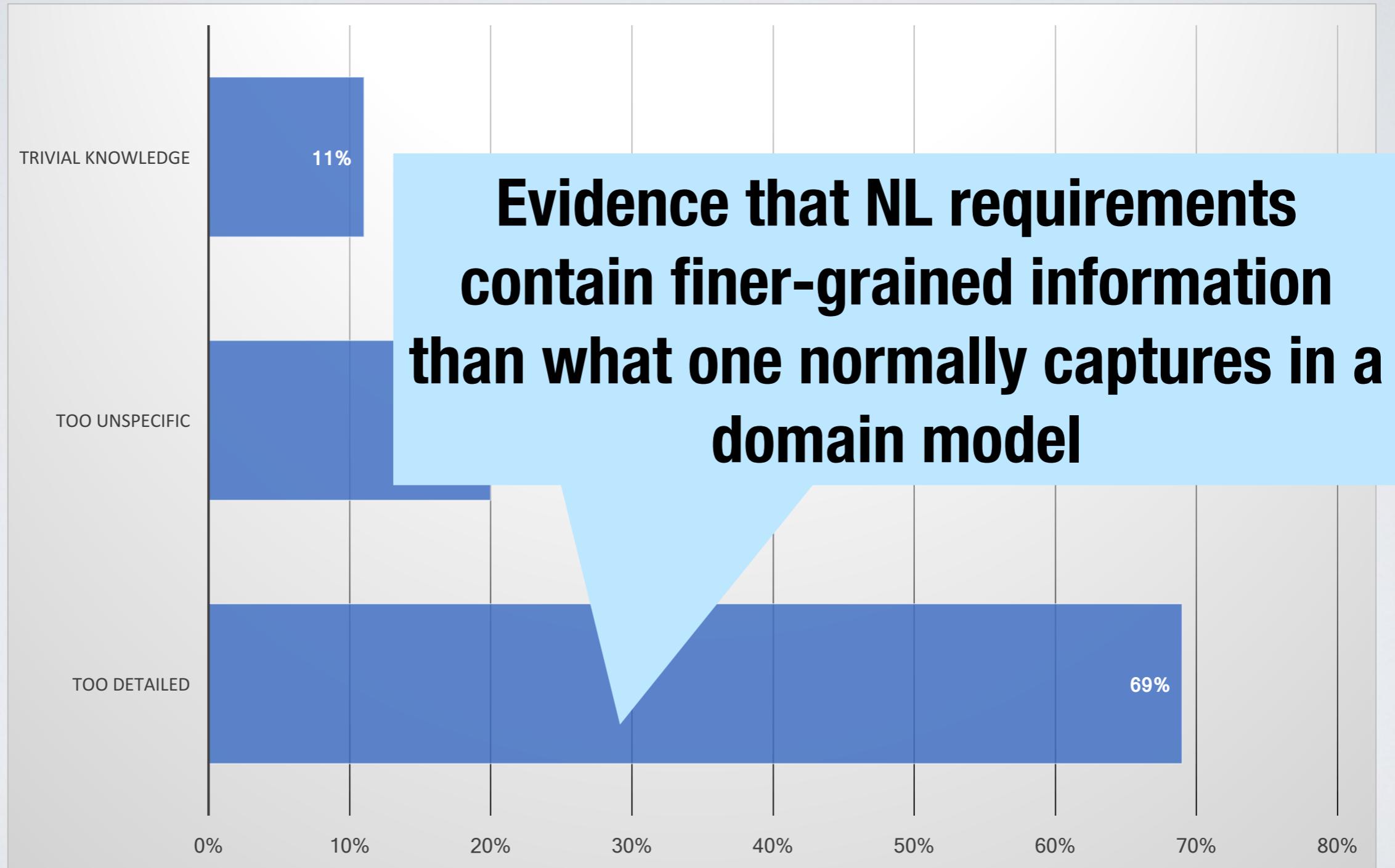
**Missed: 10% (avg.)**

**Incorrectness largely explained by NLP errors**

**Breakdown of the remaining 63%**

**12% are incorrect  
51% are **correct but superfluous****

# Statistics for Superfluouslyness

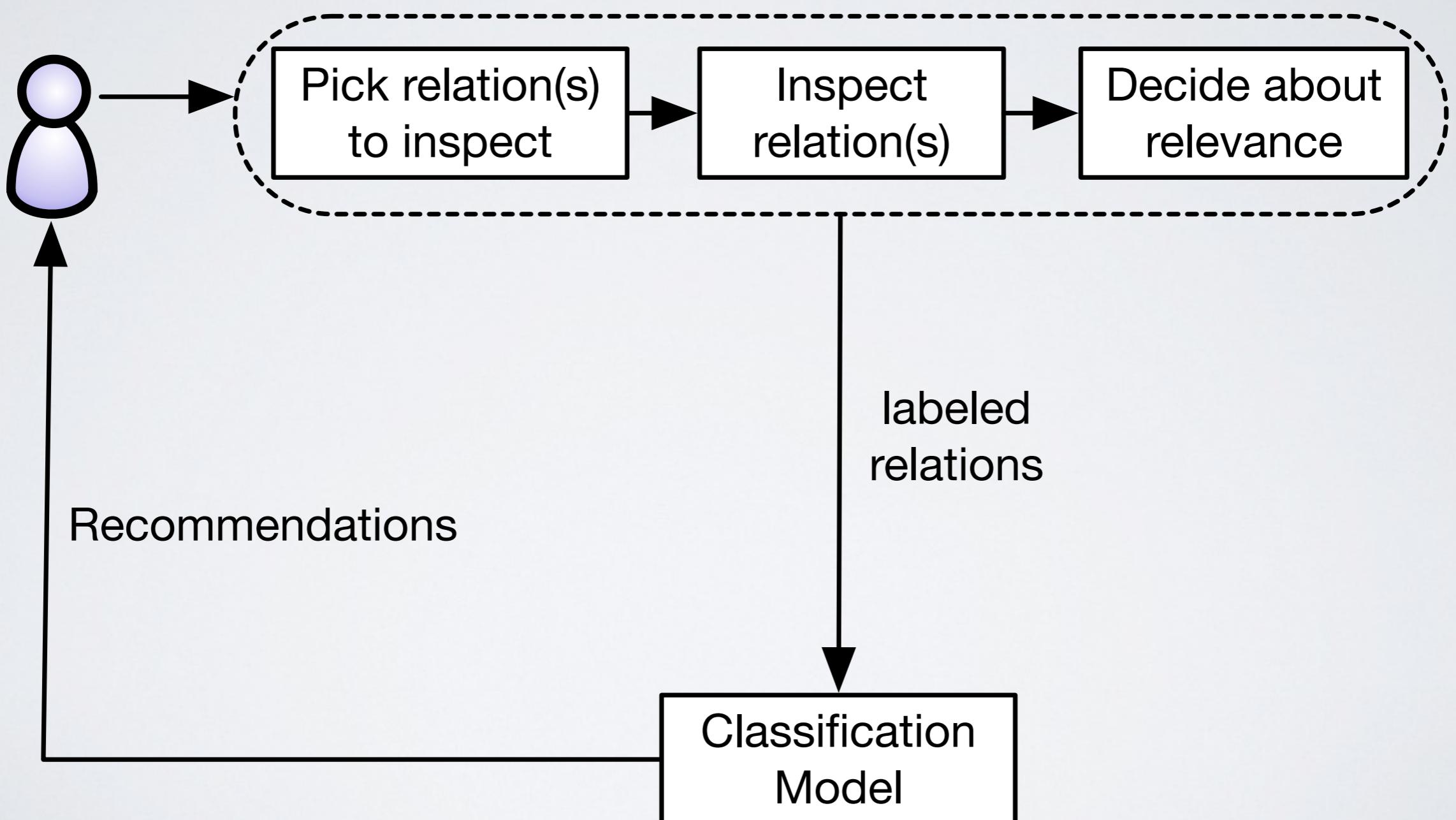


**Can we improve the relevance of  
model extraction results?**

# Active Learning

- **Definition:** Machine learning paradigm in which a learning technique interactively requests inputs from an external source in order to improve the accuracy of the machine learning model.
- **Application:** We process analysts' feedback, and dynamically apply the logic gleaned from the feedback for reducing superfluous information.

# Active Learning Feedback Loop



# Example Features (1/2)

## **Label-independent (Never updated):**

- Type of the relation: Association, Aggregation, Generalization
- The extraction rule that produced the relation
- Number of tokens in the relation's end points (concepts)

# Example Features (2/2)

## **Label-dependent (Updated with new relations):**

- Number of relevant relations (in the training set) extracted from the same requirement as the given relation
- Number of relevant relations in the training set that share one end concept with the given relation
- Number of relevant relations in the training set that share both end concept with a given relation

# Effectiveness of Detecting Superfluous Relations

- **96% of the recommendations made are correct**
  - The approach is unlikely to throw the analyst off-course.
- **45% of the superfluous relations are automatically marked**
  - Potentially significant savings
- We do not need a large seed training set: 30-40 relations

# Requirements-Driven Testing

# Traceability

- In many domains, various types of traceability are required
- For example, in automotive (ISO 26262), traceability between requirements and system tests: **requirements-driven testing**
- **Many requirements, many tests, therefore many traces ...**
- Automation is required

# Context

IEE develops real-time embedded systems:

- Automotive **safety sensing systems**
- Automotive **comfort & convenience systems**,  
e.g., Smart Trunk Opener



International Electronics  
& Engineering (IEE)



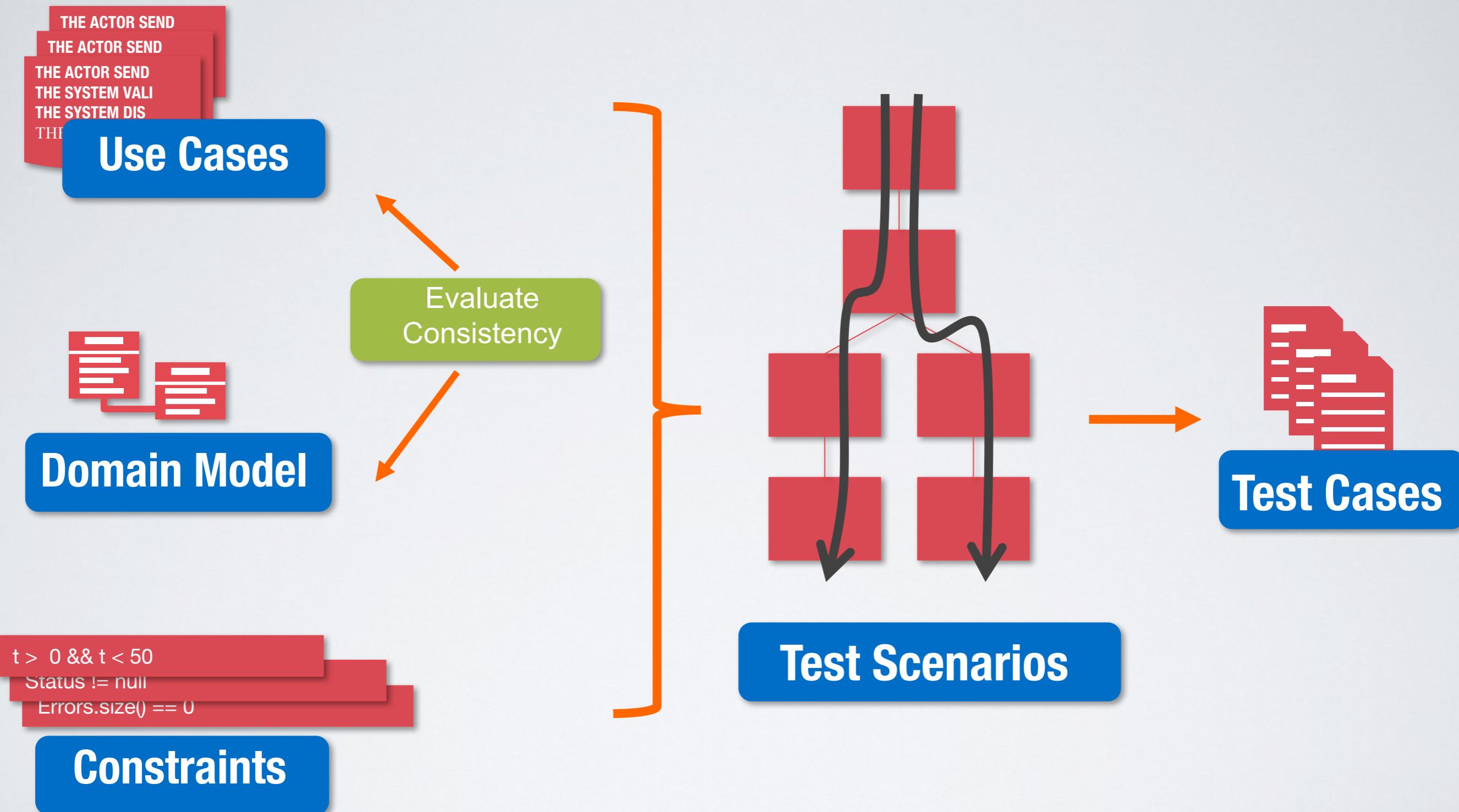
# Objectives

- Support **generation test cases from requirements**
- Capture and **create traceability information** between test cases and requirements
  - Requirements are captured through **use cases**
  - Use cases are used to **communicate** with customers and the system test team
  - Complete and precise **behavioral models** are not an option:  
**too difficult and expensive** (no model-based testing)

# Strategy

- **Analyzable** use case specifications
- Automatically extract test model from the use case specifications using **Natural Language Processing**
- **Minimize modeling**, domain modeling only
- **No behavioral modeling**

# UMTG



# Restricted Use Case Modeling: RUCM

- RUCM is based on a (1) **template**, (2) **restriction rules**, and (3) specific **keywords** constraining the use of natural language in use case specifications
- RUCM **reduces ambiguity** and **facilitates automated analysis of use cases**
- **Conformance** is supported by a tool based on NLP

**Use Case Name:** Identify Occupancy Status

**Actors:** AirbagControlUnit

**Precondition:** The system has been initialized

...

## Basic Flow

1. The seat **SENDS** occupancy status **TO** the system.

**Postcondition:** The occupant class for airbag control has been sent.  
2. **INCLUDE USE CASE** Classify occupancy status.

3. The system **VALIDATES THAT** the occupant class for airbag control is valid.

4. The system **SENDS** the occupant class for airbag control **TO** AirbagControlUnit.

## Specific Alternative Flow

**RFS 3**

**Postcondition:** The previous occupant class for airbag control has been sent.

1. **IF** the occupant class for airbag control is not valid **THEN**

2. The system **SENDS** the previous occupant class for airbag control **TO**

1

## Elicit Use Cases

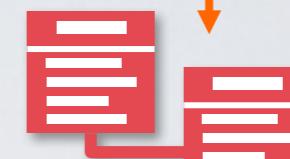
THE ACTOR SEND  
THE ACTOR SEND  
THE ACTOR SEND  
THE SYSTEM VALIDATE

RUCM  
Use Cases

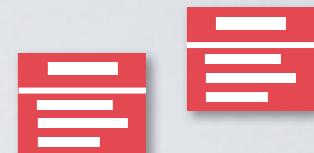
Evaluate  
Consistency

2

## Model the Domain



Domain Model



Missing Entities

4

Identify Constraints

TEMPERATURE IS LOW

STATUS IS VALID

ERRORS ARE ABSENT

Constraint descriptions

5

Specify Constraints

$t > 0 \&& t < 50$

Status != null

Errors.size() == 0

OCL constraints

6

Generate  
Scenarios and  
Inputs

Object  
Diagrams

Test  
Scenarios

7

Generate  
Test Cases



Test Cases

Mapping Table

1

## Elicit Use Cases

THE ACTOR SEND  
THE ACTOR SEND  
THE ACTOR SEND  
THE SYSTEM VALIDATE  
THE SYSTEM VALIDATE  
THE ACTOR SEND

RUCM  
Use Cases

2

## Model the Domain



Domain Model

3

## Evaluate Consistency

4

## Identify Constraints



TEMPERATURE IS LOW

STATUS IS VALID

ERRORS ARE ABSENT

Constraint descriptions

5

## Specify Constraints

$t > 0 \&& t < 50$

Status != null

Errors.size() == 0

OCL constraints

6

## Generate Scenarios and Inputs

*Based on Natural Language Processing*

## Basic Flow

1. The seat **SENDS** occupancy status **TO** the system.

**DOMAIN ENTITY**



**INPUT STEP**

2. **INCLUDE USE CASE** Classify occupancy status.



**INCLUDE STEP**

3. The system **VALIDATES THAT**



**CONDITIONAL STEP**

the occupant class for airbag control is valid and

**CONSTRAINT**

the occupant class for seat belt reminder is valid.

**CONSTRAINT**

4. The system **SENDS** the occupant class for airbag control **TO**

AirbagControlUnit.

**DOMAIN ENTITY**



**OUTPUT STEP**

5. The system **SENDS** the occupant class for seat belt reminder **TO**

SeatBeltControlUnit.

**DOMAIN ENTITY**



**OUTPUT STEP**

6. The System Waits for next execution cycle.

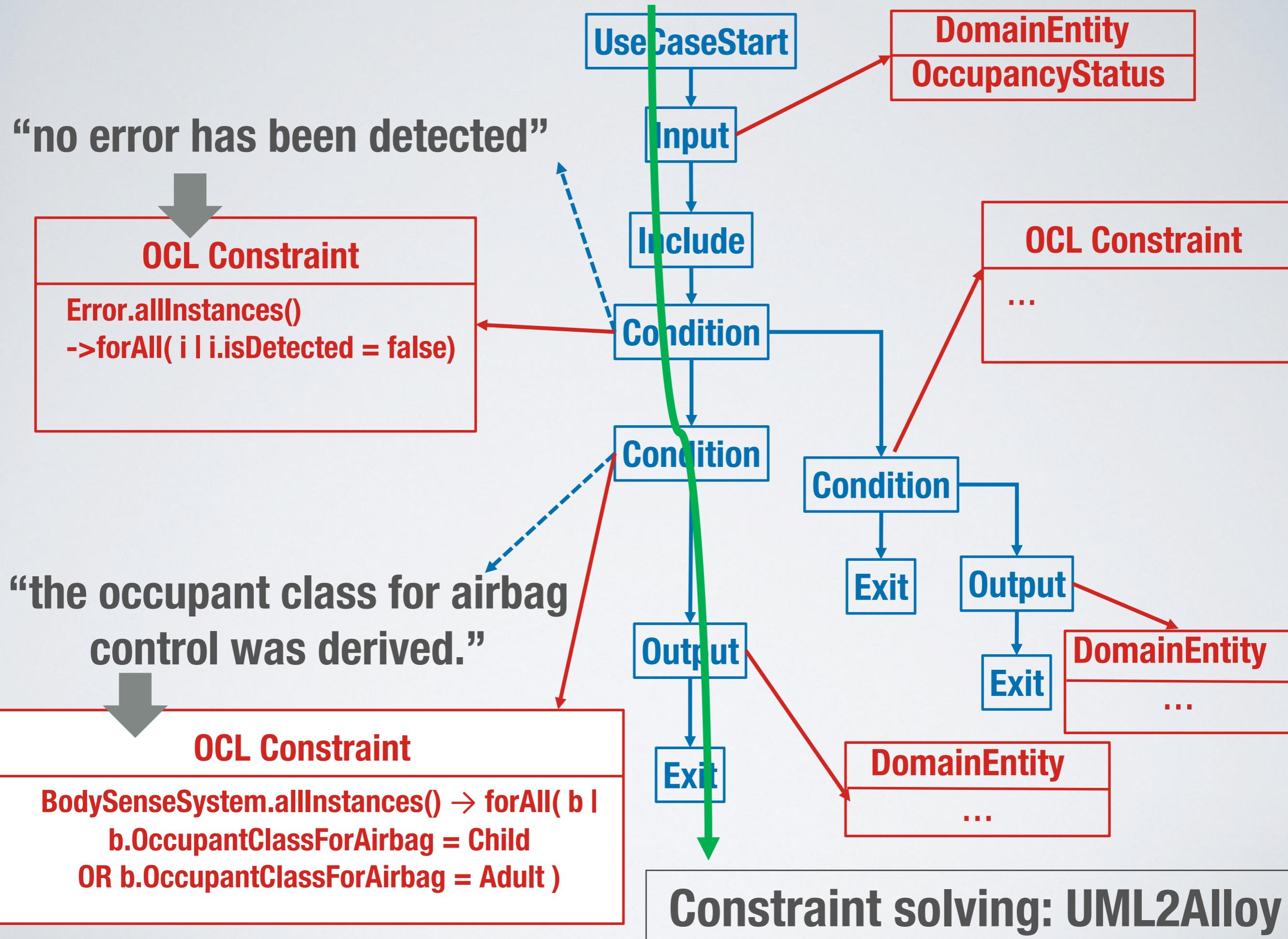


**INTERNAL STEP**

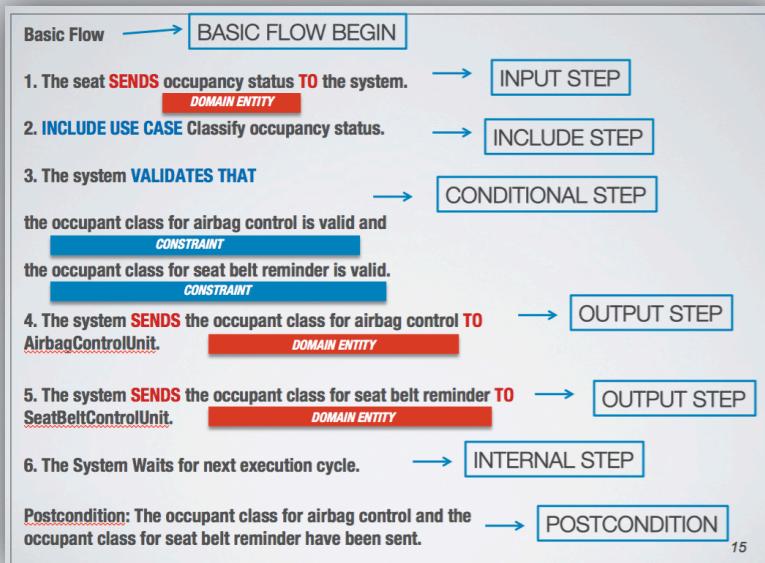
Postcondition: The occupant class for airbag control and the  
occupant class for seat belt reminder have been sent.



**POSTCONDITION**

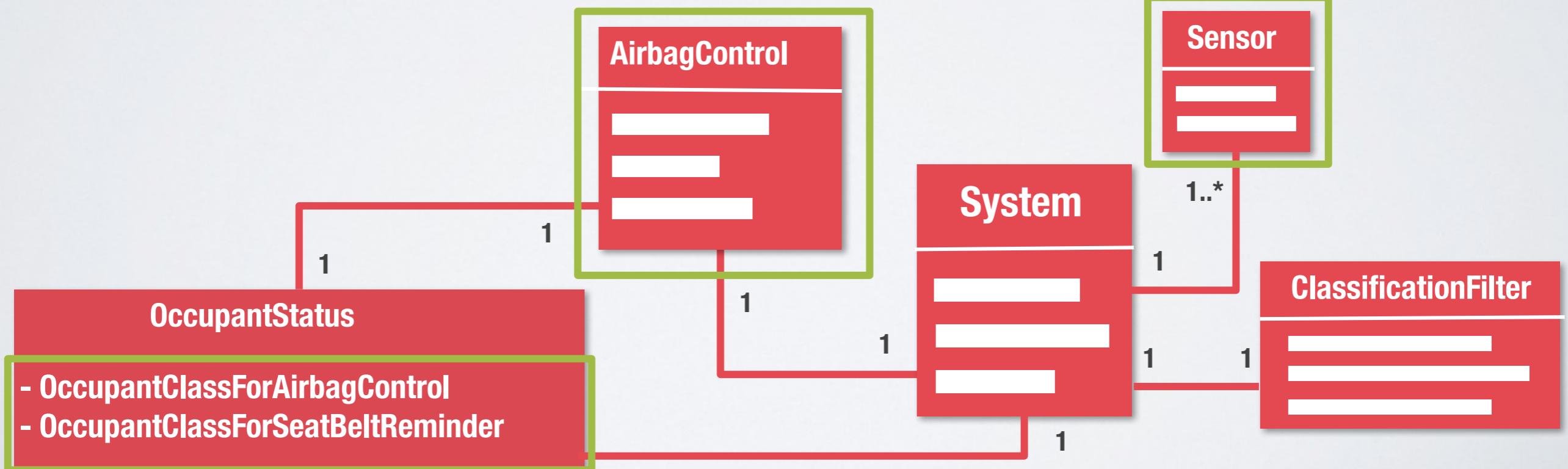


# Evaluate Model Consistency



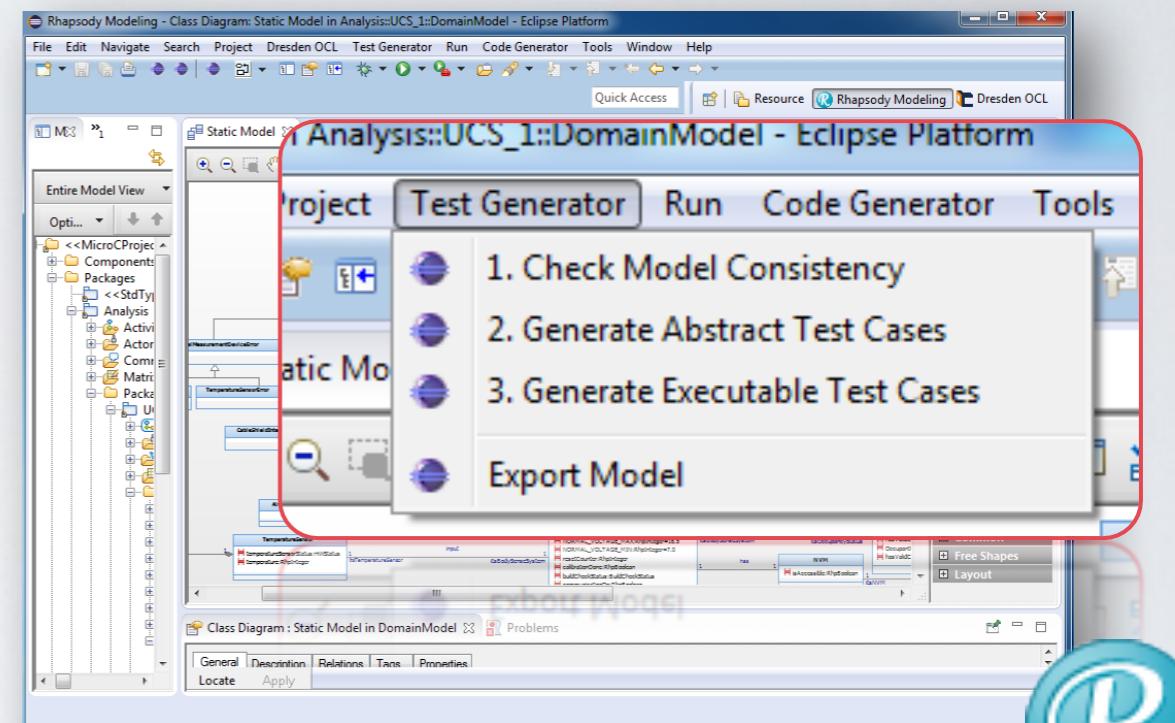
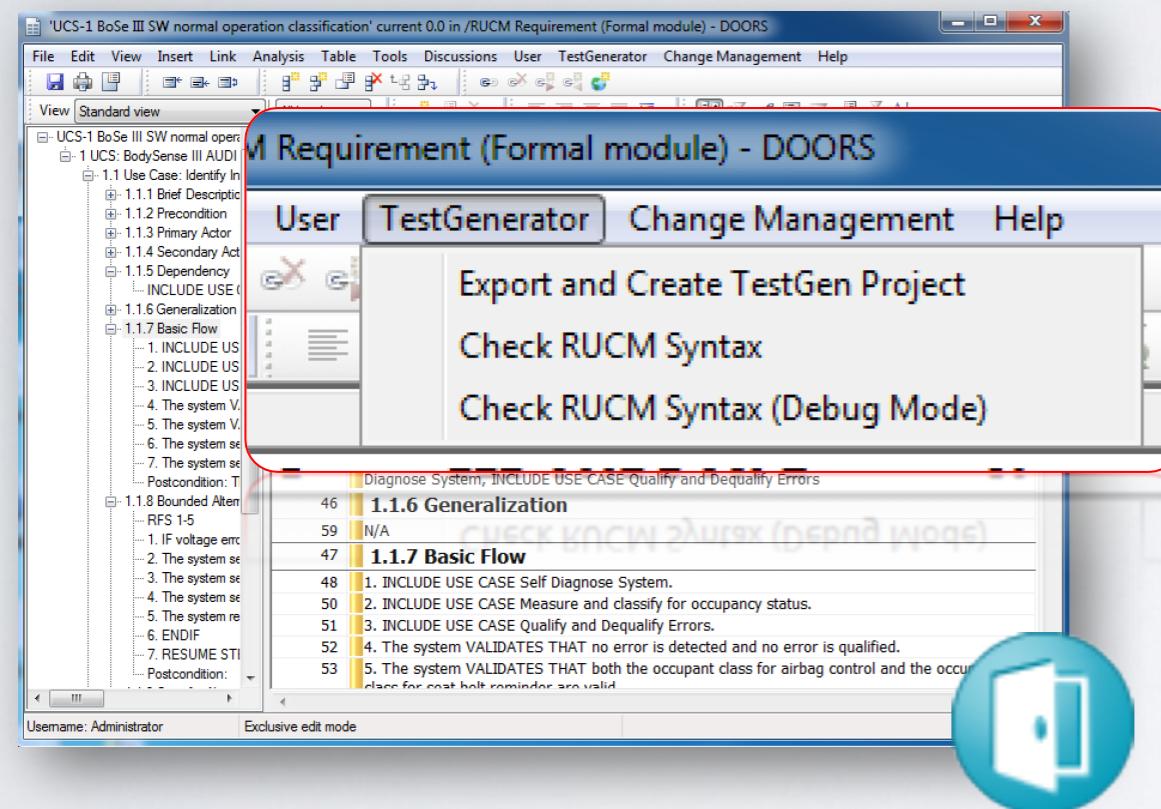
## Tagged Use Case

## Domain Entities





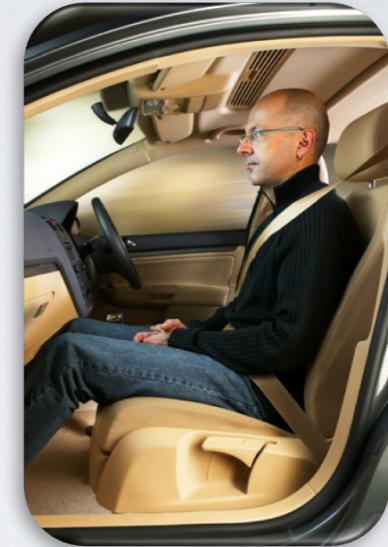
# Toolset integrated with IBM DOORS and Rhapsody



<https://sites.google.com/site/umtgTestGen/>

# Case Study

- BodySense, embedded system for detecting occupancy status in a car



- Evaluation:

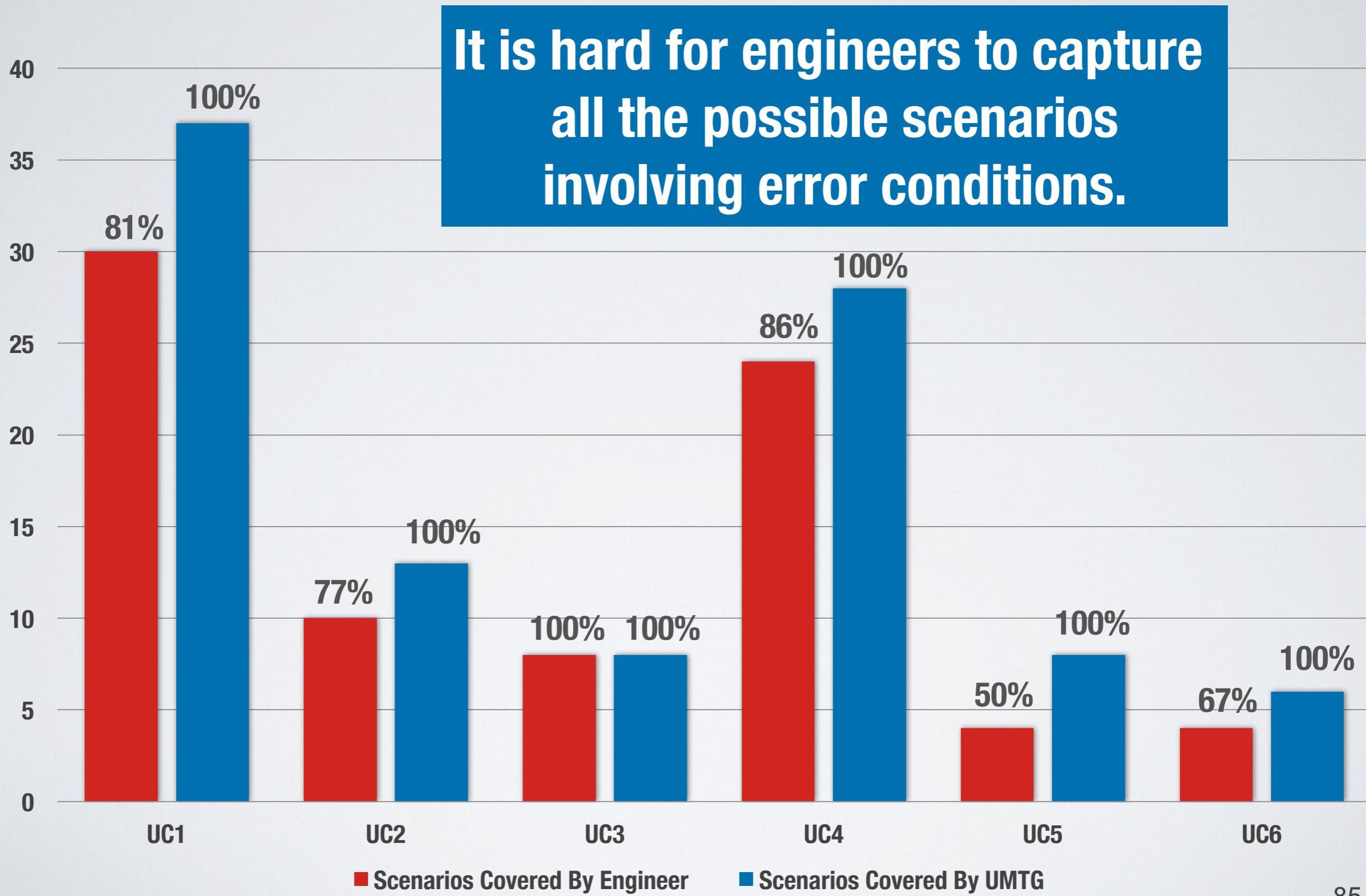
- Cost of additional modelling (Constraints)
- Effectiveness in terms of covered scenarios compared to current practice at IEE
- Keep in mind changes and repeated testing

# Costs of Additional Modeling

Use Case	Steps	Use Case Flows	OCL Constraints
UC1	50	8	9
UC2	44	13	7
UC3	35	8	8
UC4	59	11	12
UC5	30	8	5
UC6	25	6	12

**5 to 10 minutes to write each constraints  
=> A maximum of 10 hours in total**

# Effectiveness: scenarios covered



# Generating OCL Constraints

- Constraints may be a challenge in practice
- NLP: **Semantic Role Labeling**
- Determine the role of words in a sentence (e.g., affected actor)
- Match words with corresponding concepts in the domain model
- Generate an OCL formula based on patterns

# Semantic Role Labeling (SRL)

“no error has been detected”

A1

verb

Error.allInstances()->forAll( i | i.isDetected = false)

A1

verb

A0: actor that performs  
an activity

A1: actor that is affected by the  
activity described in a sentence

“The system detects temperature errors

A0

verb

A1

TemperatureError.allInstances()->forAll( i | i.isDetected = true)

A1

verb

# Empirical Evaluation

- Case study: BodySense, embedded system for detecting occupancy status in a car



- Evaluation:

- Automatically generate the OCL constraints required to automatically derive executable test cases
- Automatically generate executable test cases

# OCL generation: Precision and Recall

- 88 OCL constraints to be generated
- OCLGen generates 69 constraints
  - 66 correct, only 3 incorrect
- **Very high precision**

$$\text{precision} = \frac{\# \text{ Correctly generated constraints}}{\# \text{ Generated constraints}} = \frac{66}{69} = 0.97$$

- **High Recall**

$$\text{recall} = \frac{\# \text{ Correctly generated constraints}}{\# \text{ Constraints to be generated}} = \frac{66}{88} = 0.75$$

# Results: Limiting Factors

- **Imprecise specifications**

- “The system VALIDATES THAT the temperature is valid“

`BodySense.allInstances()->forAll( i | i.temperature < 200 )`

- **Inconsistent terminology**

- “The system VALIDATES THAT the occupancy status is valid“

`BodySense.allInstances()->forAll( i | i.occupancyStatus <> Empty )`

# Security Testing

```
1 MISUSE CASE Bypass Authorization Schema  
2 Description The MALICIOUS user accesses resources that are dedicated to  
3 a user with a different role.  
4 P |  
5 h | 1 MISUSE CASE Bypass Authorization Schema  
6 h | 2 Description The MALICIOUS user accesses resources that are dedicated to  
7 h | 3 a user with a different role.  
8 B |  
9 B | 1 MISUSE CASE Bypass Authorization Schema  
10 B | 2 Description The MALICIOUS user accesses resources that are dedicated to  
11 B | 3 a user with a different role.  
12 B | 4 B |  
13 B | 5 B | 1 MISUSE CASE Bypass Authorization Schema  
14 B | 5 B | 2 Description The MALICIOUS user accesses resources that are dedicated to  
15 B | 5 B | 3 a user with a different role.  
16 B | 6 B |  
17 B | 7 B | 1 FOREACH role  
18 B | 7 B | 2 The MALICIOUS user sends username and password to the system through  
19 B | 7 B | 3 FORM resource  
20 B | 7 B | 4 The MALICIOUS user requests the response from the system.  
21 B | 7 B | 5 The system sends a response page to the MALICIOUS user.  
22 B | 7 B | 6 The MALICIOUS user EXPLOITS the system using the response page and  
23 B | 7 B | 7 the role.  
24 B | 7 B | 8 7. ENDFOR  
25 B | 7 B | 9 8. EXIT  
26 B | 7 B | 10 9. The system sends a response page to the MALICIOUS user.  
27 B | 7 B | 11 10. The system VALIDATES THAT the user agreed to show his name  
28 B | 7 B | 12 11. The system VALIDATES THAT the user agreed to show his name  
29 B | 7 B | 13 12. The system VALIDATES THAT the user agreed to show his name  
30 B | 7 B | 14 Postcondition: The MALICIOUS user has executed a function dedicated to  
31 B | 7 B | 15 another user with different role.  
32 B | 7 B | 16 Specific Alternative Threat Flow (SATF1)  
33 B | 7 B | 17 1. IF the user contains a role parameter in the URL THEN  
34 B | 7 B | 18 2. POSTON /patients/role (HTTP POST)  
35 B | 7 B | 19 3. RESUME STEP1  
36 B | 7 B | 20 4. ENDIF.  
37 B | 7 B | 21 5. RESUME STEP1  
38 B | 7 B | 22 6. RESUME STEP1  
39 B | 7 B | 23 7. RESUME STEP1  
40 B | 7 B | 24 8. RESUME STEP1  
41 B | 7 B | 25 9. RESUME STEP1  
42 B | 7 B | 26 10. RESUME STEP1  
43 B | 7 B | 27 11. RESUME STEP1  
44 B | 7 B | 28 12. RESUME STEP1  
45 B | 7 B | 29 13. RESUME STEP1  
46 B | 7 B | 30 14. RESUME STEP1  
47 B | 7 B | 31 15. RESUME STEP1  
48 B | 7 B | 32 16. RESUME STEP1  
49 B | 7 B | 33 Postcondition: The malicious user cannot access the resource dedicated to  
users with a different role.
```

## Misuse Case Specifications

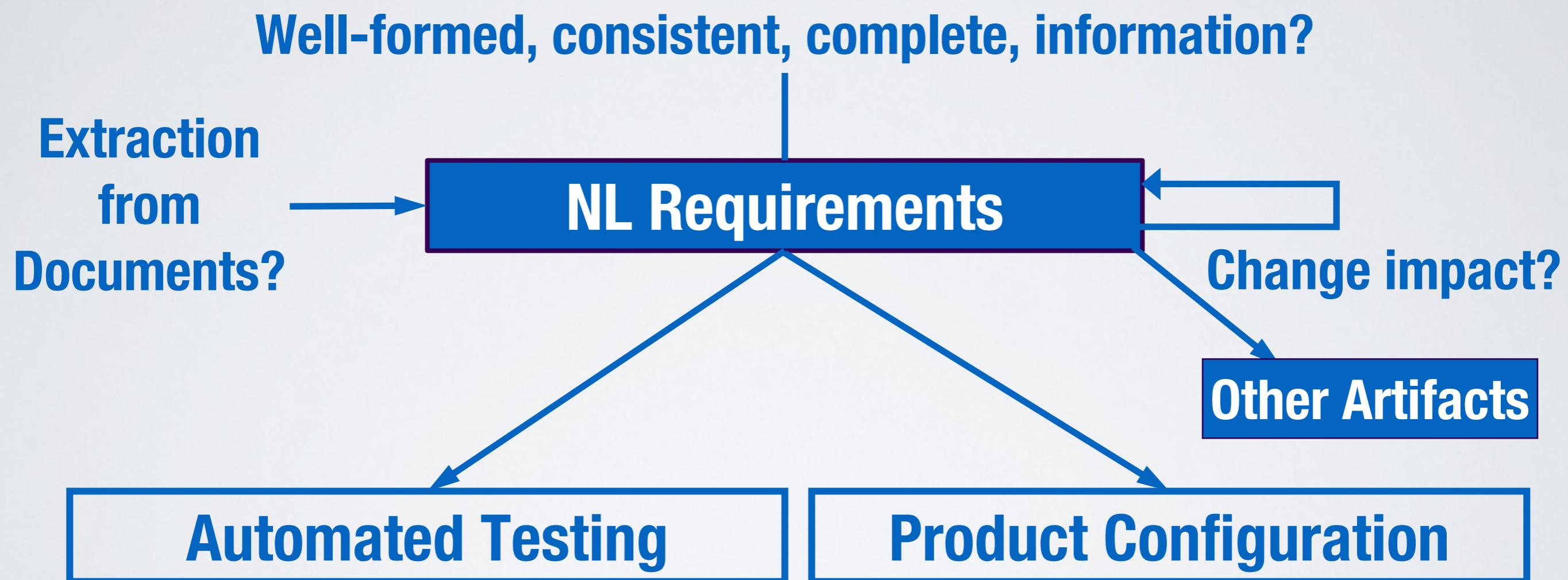
### Automated Generation of Executable Test Cases

### Security Vulnerability Testing

Addressing the identification of system vulnerabilities

```
1 SECURITY USE CASE Anonymize user data  
2 1. The system VALIDATES THAT the user agreed to show his name  
3 Postcondition: the user data can be visualized.  
4 Specific SECURITY USE CASE Anonymize user data  
5 RFS 1. 1. The system VALIDATES THAT the user agreed to show his name  
6 RFS 1. 2. Postcondition: the user data can be visualized.  
7 EXIT SECURITY USE CASE Anonymize user data  
8 Poston RFS 1. 1. The system VALIDATES THAT the user agreed to show his name  
9 SECUI 2.The s Postcondition: the user data can be visualized.  
10 SECUI 2.1. The system VALIDATES THAT the user agreed to show his name  
11 SECUI 2.2. Poston  
12 SECUI 2.3. EXIT  
13 SECUI 2.4. Specific Alternative Flow  
14 SECUI 2.5. Poston  
15 SECUI 2.6. EXIT  
16 SECUI 2.7. Specific Alternative Flow  
17 SECUI 2.8. Poston  
18 SECUI 2.9. EXIT  
19 SECUI 2.10. Specific Alternative Flow  
20 SECUI 2.11. Poston  
21 SECUI 2.12. EXIT  
22 SECUI 2.13. Specific Alternative Flow  
23 SECUI 2.14. Poston  
24 SECUI 2.15. EXIT  
25 SECUI 2.16. Specific Alternative Flow  
26 SECUI 2.17. Poston  
27 SECUI 2.18. EXIT  
28 SECUI 2.19. Specific Alternative Flow  
29 SECUI 2.20. Poston  
30 SECUI 2.21. EXIT  
31 SECUI 2.22. Specific Alternative Flow  
32 SECUI 2.23. Poston  
33 SECUI 2.24. EXIT  
34 SECUI 2.25. Specific Alternative Flow  
35 SECUI 2.26. Poston  
36 SECUI 2.27. EXIT  
37 SECUI 2.28. Specific Alternative Flow  
38 SECUI 2.29. Poston  
39 SECUI 2.30. EXIT  
40 SECUI 2.31. Specific Alternative Flow  
41 SECUI 2.32. Poston  
42 SECUI 2.33. EXIT  
43 SECUI 2.34. Specific Alternative Flow  
44 SECUI 2.35. Poston  
45 SECUI 2.36. EXIT  
46 SECUI 2.37. Specific Alternative Flow  
47 SECUI 2.38. Poston  
48 SECUI 2.39. EXIT  
49 SECUI 2.40. Specific Alternative Flow  
50 SECUI 2.41. Poston  
51 SECUI 2.42. EXIT  
52 SECUI 2.43. Specific Alternative Flow  
53 SECUI 2.44. Poston  
54 SECUI 2.45. EXIT  
55 SECUI 2.46. Specific Alternative Flow  
56 SECUI 2.47. Poston  
57 SECUI 2.48. EXIT  
58 SECUI 2.49. Specific Alternative Flow  
59 SECUI 2.50. Poston  
60 SECUI 2.51. EXIT  
61 SECUI 2.52. Specific Alternative Flow  
62 SECUI 2.53. Poston  
63 SECUI 2.54. EXIT  
64 SECUI 2.55. Specific Alternative Flow  
65 SECUI 2.56. Poston  
66 SECUI 2.57. EXIT  
67 SECUI 2.58. Specific Alternative Flow  
68 SECUI 2.59. Poston  
69 SECUI 2.60. EXIT  
70 SECUI 2.61. Specific Alternative Flow  
71 SECUI 2.62. Poston  
72 SECUI 2.63. EXIT  
73 SECUI 2.64. Specific Alternative Flow  
74 SECUI 2.65. Poston  
75 SECUI 2.66. EXIT  
76 SECUI 2.67. Specific Alternative Flow  
77 SECUI 2.68. Poston  
78 SECUI 2.69. EXIT  
79 SECUI 2.70. Specific Alternative Flow  
80 SECUI 2.71. Poston  
81 SECUI 2.72. EXIT  
82 SECUI 2.73. Specific Alternative Flow  
83 SECUI 2.74. Poston  
84 SECUI 2.75. EXIT  
85 SECUI 2.76. Specific Alternative Flow  
86 SECUI 2.77. Poston  
87 SECUI 2.78. EXIT  
88 SECUI 2.79. Specific Alternative Flow  
89 SECUI 2.80. Poston  
90 SECUI 2.81. EXIT  
91 SECUI 2.82. Specific Alternative Flow  
92 SECUI 2.83. Poston  
93 SECUI 2.84. EXIT  
94 SECUI 2.85. Specific Alternative Flow  
95 SECUI 2.86. Poston  
96 SECUI 2.87. EXIT  
97 SECUI 2.88. Specific Alternative Flow  
98 SECUI 2.89. Poston  
99 SECUI 2.90. EXIT  
100 SECUI 2.91. Specific Alternative Flow  
101 SECUI 2.92. Poston  
102 SECUI 2.93. EXIT  
103 SECUI 2.94. Specific Alternative Flow  
104 SECUI 2.95. Poston  
105 SECUI 2.96. EXIT  
106 SECUI 2.97. Specific Alternative Flow  
107 SECUI 2.98. Poston  
108 SECUI 2.99. EXIT  
109 SECUI 2.100. Specific Alternative Flow  
110 SECUI 2.101. Poston  
111 SECUI 2.102. EXIT  
112 SECUI 2.103. Specific Alternative Flow  
113 SECUI 2.104. Poston  
114 SECUI 2.105. EXIT  
115 SECUI 2.106. Specific Alternative Flow  
116 SECUI 2.107. Poston  
117 SECUI 2.108. EXIT  
118 SECUI 2.109. Specific Alternative Flow  
119 SECUI 2.110. Poston  
120 SECUI 2.111. EXIT  
121 SECUI 2.112. Specific Alternative Flow  
122 SECUI 2.113. Poston  
123 SECUI 2.114. EXIT  
124 SECUI 2.115. Specific Alternative Flow  
125 SECUI 2.116. Poston  
126 SECUI 2.117. EXIT  
127 SECUI 2.118. Specific Alternative Flow  
128 SECUI 2.119. Poston  
129 SECUI 2.120. EXIT  
130 SECUI 2.121. Specific Alternative Flow  
131 SECUI 2.122. Poston  
132 SECUI 2.123. EXIT  
133 SECUI 2.124. Specific Alternative Flow  
134 SECUI 2.125. Poston  
135 SECUI 2.126. EXIT  
136 SECUI 2.127. Specific Alternative Flow  
137 SECUI 2.128. Poston  
138 SECUI 2.129. EXIT  
139 SECUI 2.130. Specific Alternative Flow  
140 SECUI 2.131. Poston  
141 SECUI 2.132. EXIT  
142 SECUI 2.133. Specific Alternative Flow  
143 SECUI 2.134. Poston  
144 SECUI 2.135. EXIT  
145 SECUI 2.136. Specific Alternative Flow  
146 SECUI 2.137. Poston  
147 SECUI 2.138. EXIT  
148 SECUI 2.139. Specific Alternative Flow  
149 SECUI 2.140. Poston  
150 SECUI 2.141. EXIT  
151 SECUI 2.142. Specific Alternative Flow  
152 SECUI 2.143. Poston  
153 SECUI 2.144. EXIT  
154 SECUI 2.145. Specific Alternative Flow  
155 SECUI 2.146. Poston  
156 SECUI 2.147. EXIT  
157 SECUI 2.148. Specific Alternative Flow  
158 SECUI 2.149. Poston  
159 SECUI 2.150. EXIT  
160 SECUI 2.151. Specific Alternative Flow  
161 SECUI 2.152. Poston  
162 SECUI 2.153. EXIT  
163 SECUI 2.154. Specific Alternative Flow  
164 SECUI 2.155. Poston  
165 SECUI 2.156. EXIT  
166 SECUI 2.157. Specific Alternative Flow  
167 SECUI 2.158. Poston  
168 SECUI 2.159. EXIT  
169 SECUI 2.160. Specific Alternative Flow  
170 SECUI 2.161. Poston  
171 SECUI 2.162. EXIT  
172 SECUI 2.163. Specific Alternative Flow  
173 SECUI 2.164. Poston  
174 SECUI 2.165. EXIT  
175 SECUI 2.166. Specific Alternative Flow  
176 SECUI 2.167. Poston  
177 SECUI 2.168. EXIT  
178 SECUI 2.169. Specific Alternative Flow  
179 SECUI 2.170. Poston  
180 SECUI 2.171. EXIT  
181 SECUI 2.172. Specific Alternative Flow  
182 SECUI 2.173. Poston  
183 SECUI 2.174. EXIT  
184 SECUI 2.175. Specific Alternative Flow  
185 SECUI 2.176. Poston  
186 SECUI 2.177. EXIT  
187 SECUI 2.178. Specific Alternative Flow  
188 SECUI 2.179. Poston  
189 SECUI 2.180. EXIT  
190 SECUI 2.181. Specific Alternative Flow  
191 SECUI 2.182. Poston  
192 SECUI 2.183. EXIT  
193 SECUI 2.184. Specific Alternative Flow  
194 SECUI 2.185. Poston  
195 SECUI 2.186. EXIT  
196 SECUI 2.187. Specific Alternative Flow  
197 SECUI 2.188. Poston  
198 SECUI 2.189. EXIT  
199 SECUI 2.190. Specific Alternative Flow  
200 SECUI 2.191. Poston  
201 SECUI 2.192. EXIT  
202 SECUI 2.193. Specific Alternative Flow  
203 SECUI 2.194. Poston  
204 SECUI 2.195. EXIT  
205 SECUI 2.196. Specific Alternative Flow  
206 SECUI 2.197. Poston  
207 SECUI 2.198. EXIT  
208 SECUI 2.199. Specific Alternative Flow  
209 SECUI 2.200. Poston  
210 SECUI 2.201. EXIT  
211 SECUI 2.202. Specific Alternative Flow  
212 SECUI 2.203. Poston  
213 SECUI 2.204. EXIT  
214 SECUI 2.205. Specific Alternative Flow  
215 SECUI 2.206. Poston  
216 SECUI 2.207. EXIT  
217 SECUI 2.208. Specific Alternative Flow  
218 SECUI 2.209. Poston  
219 SECUI 2.210. EXIT  
220 SECUI 2.211. Specific Alternative Flow  
221 SECUI 2.212. Poston  
222 SECUI 2.213. EXIT  
223 SECUI 2.214. Specific Alternative Flow  
224 SECUI 2.215. Poston  
225 SECUI 2.216. EXIT  
226 SECUI 2.217. Specific Alternative Flow  
227 SECUI 2.218. Poston  
228 SECUI 2.219. EXIT  
229 SECUI 2.220. Specific Alternative Flow  
230 SECUI 2.221. Poston  
231 SECUI 2.222. EXIT  
232 SECUI 2.223. Specific Alternative Flow  
233 SECUI 2.224. Poston  
234 SECUI 2.225. EXIT  
235 SECUI 2.226. Specific Alternative Flow  
236 SECUI 2.227. Poston  
237 SECUI 2.228. EXIT  
238 SECUI 2.229. Specific Alternative Flow  
239 SECUI 2.230. Poston  
240 SECUI 2.231. EXIT  
241 SECUI 2.232. Specific Alternative Flow  
242 SECUI 2.233. Poston  
243 SECUI 2.234. EXIT  
244 SECUI 2.235. Specific Alternative Flow  
245 SECUI 2.236. Poston  
246 SECUI 2.237. EXIT  
247 SECUI 2.238. Specific Alternative Flow  
248 SECUI 2.239. Poston  
249 SECUI 2.240. EXIT  
250 SECUI 2.241. Specific Alternative Flow  
251 SECUI 2.242. Poston  
252 SECUI 2.243. EXIT  
253 SECUI 2.244. Specific Alternative Flow  
254 SECUI 2.245. Poston  
255 SECUI 2.246. EXIT  
256 SECUI 2.247. Specific Alternative Flow  
257 SECUI 2.248. Poston  
258 SECUI 2.249. EXIT  
259 SECUI 2.250. Specific Alternative Flow  
260 SECUI 2.251. Poston  
261 SECUI 2.252. EXIT  
262 SECUI 2.253. Specific Alternative Flow  
263 SECUI 2.254. Poston  
264 SECUI 2.255. EXIT  
265 SECUI 2.256. Specific Alternative Flow  
266 SECUI 2.257. Poston  
267 SECUI 2.258. EXIT  
268 SECUI 2.259. Specific Alternative Flow  
269 SECUI 2.260. Poston  
270 SECUI 2.261. EXIT  
271 SECUI 2.262. Specific Alternative Flow  
272 SECUI 2.263. Poston  
273 SECUI 2.264. EXIT  
274 SECUI 2.265. Specific Alternative Flow  
275 SECUI 2.266. Poston  
276 SECUI 2.267. EXIT  
277 SECUI 2.268. Specific Alternative Flow  
278 SECUI 2.269. Poston  
279 SECUI 2.270. EXIT  
280 SECUI 2.271. Specific Alternative Flow  
281 SECUI 2.272. Poston  
282 SECUI 2.273. EXIT  
283 SECUI 2.274. Specific Alternative Flow  
284 SECUI 2.275. Poston  
285 SECUI 2.276. EXIT  
286 SECUI 2.277. Specific Alternative Flow  
287 SECUI 2.278. Poston  
288 SECUI 2.279. EXIT  
289 SECUI 2.280. Specific Alternative Flow  
290 SECUI 2.281. Poston  
291 SECUI 2.282. EXIT  
292 SECUI 2.283. Specific Alternative Flow  
293 SECUI 2.284. Poston  
294 SECUI 2.285. EXIT  
295 SECUI 2.286. Specific Alternative Flow  
296 SECUI 2.287. Poston  
297 SECUI 2.288. EXIT  
298 SECUI 2.289. Specific Alternative Flow  
299 SECUI 2.290. Poston  
300 SECUI 2.291. EXIT  
301 SECUI 2.292. Specific Alternative Flow  
302 SECUI 2.293. Poston  
303 SECUI 2.294. EXIT  
304 SECUI 2.295. Specific Alternative Flow  
305 SECUI 2.296. Poston  
306 SECUI 2.297. EXIT  
307 SECUI 2.298. Specific Alternative Flow  
308 SECUI 2.299. Poston  
309 SECUI 2.300. EXIT  
310 SECUI 2.301. Specific Alternative Flow  
311 SECUI 2.302. Poston  
312 SECUI 2.303. EXIT  
313 SECUI 2.304. Specific Alternative Flow  
314 SECUI 2.305. Poston  
315 SECUI 2.306. EXIT  
316 SECUI 2.307. Specific Alternative Flow  
317 SECUI 2.308. Poston  
318 SECUI 2.309. EXIT  
319 SECUI 2.310. Specific Alternative Flow  
320 SECUI 2.311. Poston  
321 SECUI 2.312. EXIT  
322 SECUI 2.313. Specific Alternative Flow  
323 SECUI 2.314. Poston  
324 SECUI 2.315. EXIT  
325 SECUI 2.316. Specific Alternative Flow  
326 SECUI 2.317. Poston  
327 SECUI 2.318. EXIT  
328 SECUI 2.319. Specific Alternative Flow  
329 SECUI 2.320. Poston  
330 SECUI 2.321. EXIT  
331 SECUI 2.322. Specific Alternative Flow  
332 SECUI 2.323. Poston  
333 SECUI 2.324. EXIT  
334 SECUI 2.325. Specific Alternative Flow  
335 SECUI 2.326. Poston  
336 SECUI 2.327. EXIT  
337 SECUI 2.328. Specific Alternative Flow  
338 SECUI 2.329. Poston  
339 SECUI 2.330. EXIT  
340 SECUI 2.331. Specific Alternative Flow  
341 SECUI 2.332. Poston  
342 SECUI 2.333. EXIT  
343 SECUI 2.334. Specific Alternative Flow  
344 SECUI 2.335. Poston  
345 SECUI 2.336. EXIT  
346 SECUI 2.337. Specific Alternative Flow  
347 SECUI 2.338. Poston  
348 SECUI 2.339. EXIT  
349 SECUI 2.340. Specific Alternative Flow  
350 SECUI 2.341. Poston  
351 SECUI 2.342. EXIT  
352 SECUI 2.343. Specific Alternative Flow  
353 SECUI 2.344. Poston  
354 SECUI 2.345. EXIT  
355 SECUI 2.346. Specific Alternative Flow  
356 SECUI 2.347. Poston  
357 SECUI 2.348. EXIT  
358 SECUI 2.349. Specific Alternative Flow  
359 SECUI 2.350. Poston  
360 SECUI 2.351. EXIT  
361 SECUI 2.352. Specific Alternative Flow  
362 SECUI 2.353. Poston  
363 SECUI 2.354. EXIT  
364 SECUI 2.355. Specific Alternative Flow  
365 SECUI 2.356. Poston  
366 SECUI 2.357. EXIT  
367 SECUI 2.358. Specific Alternative Flow  
368 SECUI 2.359. Poston  
369 SECUI 2.360. EXIT  
370 SECUI 2.361. Specific Alternative Flow  
371 SECUI 2.362. Poston  
372 SECUI 2.363. EXIT  
373 SECUI 2.364. Specific Alternative Flow  
374 SECUI 2.365. Poston  
375 SECUI 2.366. EXIT  
376 SECUI 2.367. Specific Alternative Flow  
377 SECUI 2.368. Poston  
378 SECUI 2.369. EXIT  
379 SECUI 2.370. Specific Alternative Flow  
380 SECUI 2.371. Poston  
381 SECUI 2.372. EXIT  
382 SECUI 2.373. Specific Alternative Flow  
383 SECUI 2.374. Poston  
384 SECUI 2.375. EXIT  
385 SECUI 2.376. Specific Alternative Flow  
386 SECUI 2.377. Poston  
387 SECUI 2.378. EXIT  
388 SECUI 2.379. Specific Alternative Flow  
389 SECUI 2.380. Poston  
390 SECUI 2.381. EXIT  
391 SECUI 2.382. Specific Alternative Flow  
392 SECUI 2.383. Poston  
393 SECUI 2.384. EXIT  
394 SECUI 2.385. Specific Alternative Flow  
395 SECUI 2.386. Poston  
396 SECUI 2.387. EXIT  
397 SECUI 2.388. Specific Alternative Flow  
398 SECUI 2.389. Poston  
399 SECUI 2.390. EXIT  
400 SECUI 2.391. Specific Alternative Flow  
401 SECUI 2.392. Poston  
402 SECUI 2.393. EXIT  
403 SECUI 2.394. Specific Alternative Flow  
404 SECUI 2.395. Poston  
405 SECUI 2.396. EXIT  
406 SECUI 2.397. Specific Alternative Flow  
407 SECUI 2.398. Poston  
408 SECUI 2.399. EXIT  
409 SECUI 2.400. Specific Alternative Flow  
410 SECUI 2.401. Poston  
411 SECUI 2.402. EXIT  
412 SECUI 2.403. Specific Alternative Flow  
413 SECUI 2.404. Poston  
414 SECUI 2.405. EXIT  
415 SECUI 2.406. Specific Alternative Flow  
416 SECUI 2.407. Poston  
417 SECUI 2.408. EXIT  
418 SECUI 2.409. Specific Alternative Flow  
419 SECUI 2.410. Poston  
420 SECUI 2.411. EXIT  
421 SECUI 2.412. Specific Alternative Flow  
422 SECUI 2.413. Poston  
423 SECUI 2.414. EXIT  
424 SECUI 2.415. Specific Alternative Flow  
425 SECUI 2.416. Poston  
426 SECUI 2.417. EXIT  
427 SECUI 2.418. Specific Alternative Flow  
428 SECUI 2.419. Poston  
429 SECUI 2.420. EXIT  
430 SECUI 2.421. Specific Alternative Flow  
431 SECUI 2.422. Poston  
432 SECUI 2.423. EXIT  
433 SECUI 2.424. Specific Alternative Flow  
434 SECUI 2.425. Poston  
435 SECUI 2.426. EXIT  
436 SECUI 2.427. Specific Alternative Flow  
437 SECUI 2.428. Poston  
438 SECUI 2.429. EXIT  
439 SECUI 2.430. Specific Alternative Flow  
440 SECUI 2.431. Poston  
441 SECUI 2.432. EXIT  
442 SECUI 2.433. Specific Alternative Flow  
443 SECUI 2.434. Poston  
444 SECUI 2.435. EXIT  
445 SECUI 2.436. Specific Alternative Flow  
446 SECUI 2.437. Poston  
447 SECUI 2.438. EXIT  
448 SECUI 2.439. Specific Alternative Flow  
449 SECUI 2.440. Poston  
450 SECUI 2.441. EXIT  
451 SECUI 2.442. Specific Alternative Flow  
452 SECUI 2.443. Poston  
453 SECUI 2.444. EXIT  
454 SECUI 2.445. Specific Alternative Flow  
455 SECUI 2.446. Poston  
456 SECUI 2.447. EXIT  
457 SECUI 2.448. Specific Alternative Flow  
458 SECUI 2.449. Poston  
459 SECUI 2.450. EXIT  
460 SECUI 2.451. Specific Alternative Flow  
461 SECUI 2.452. Poston  
462 SECUI 2.453. EXIT  
463 SECUI 2.454. Specific Alternative Flow  
464 SECUI 2.455. Poston  
465 SECUI 2.456. EXIT  
466 SECUI 2.457. Specific Alternative Flow  
467 SECUI 2.458. Poston  
468 SECUI 2.459. EXIT  
469 SECUI 2.460. Specific Alternative Flow  
470 SECUI 2.461. Poston  
471 SECUI 2.462. EXIT  
472 SEC
```

# Summary



# The Complexity of our World

- Many applications, diversity of contexts, and types of NL requirements
- Variety of very different working assumptions
  - Form of requirements, e.g., RUCM
  - Change information
  - Modeling practice, e.g., domain models
  - Scale, e.g., embedded automotive versus satellite ground control systems

# The Road Ahead

- Practical solutions are possible based on combining advanced NLP and (often) machine learning.
- We must account for practicality and scalability at the outset, not as an afterthought.
- We need more (reported) industrial experiences, as working assumptions play a key role.

# Automated Analysis of Natural-Language Requirements: Industrial Needs and Opportunities

AIRE'2018 @ RE'2018

Lionel Briand  
Interdisciplinary Centre for ICT Security, Reliability, and Trust (SnT)  
University of Luxembourg, Luxembourg

# References

## Analysis of Natural Language Requirements

- [TSE 2017] C. Arora et al., Automated Extraction and Clustering of Requirements Glossary Terms
- [MODELS 2016] C. Arora et al., Extracting Domain Models from Natural-Language Requirements: Approach and Industrial Evaluation
- [RE 2015] C. Arora et al., Change Impact Analysis for Natural Language Requirements: An NLP Approach
- [TSE 2015] C. Arora et al., Automated Checking of Conformance to Requirements Templates using Natural Language Processing

## Requirements-Driven (Security) Testing

- [ISSTA 2015] C. Wang et al., Automatic Generation of System Test Cases from Use Case Specifications
- [ICST 2017] C. Wang et al., System Testing of Timing Requirements based on Use Cases and Timed Automata
- [ICST 2018] C. Wang et al., Automated Generation of Constraints from Use Case Specifications to Support System Testing
- [ISSRE 2018] P. Mai et al., A Natural Language Programming Approach for Requirements-based Security Testing

# References

## Product Families and Configuration

- [SoSYM 2018] I. Hajri et al., Configuring Use Case Models in Product Families
- [JSS 2018] I. Hajri et al., Change Impact Analysis for Evolving Configuration Decisions in Product Line Use Case Models

## Impact Analysis

- [FSE 2016] S. Nejati et al., Automated Change Impact Analysis between SysML Models of Requirements and Design
- [RE 2015] C. Arora et al., Change Impact Analysis for Natural Language Requirements: An NLP Approach