

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**

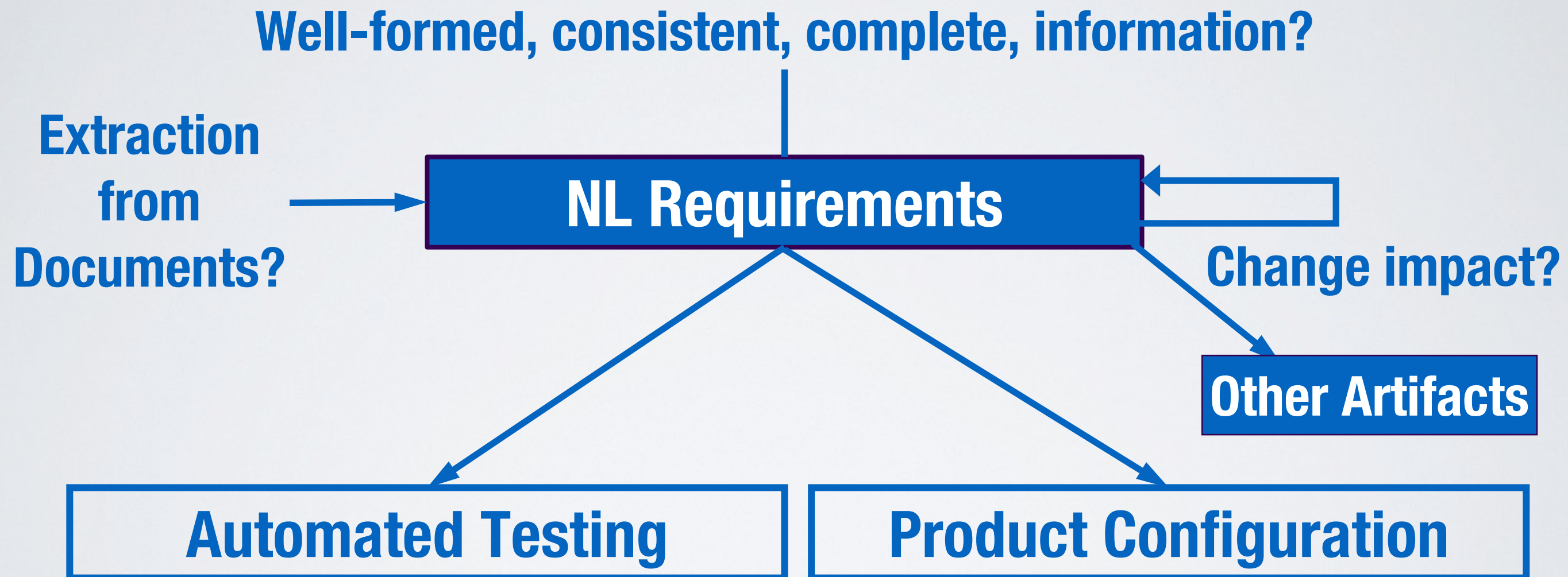
Observations

- 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

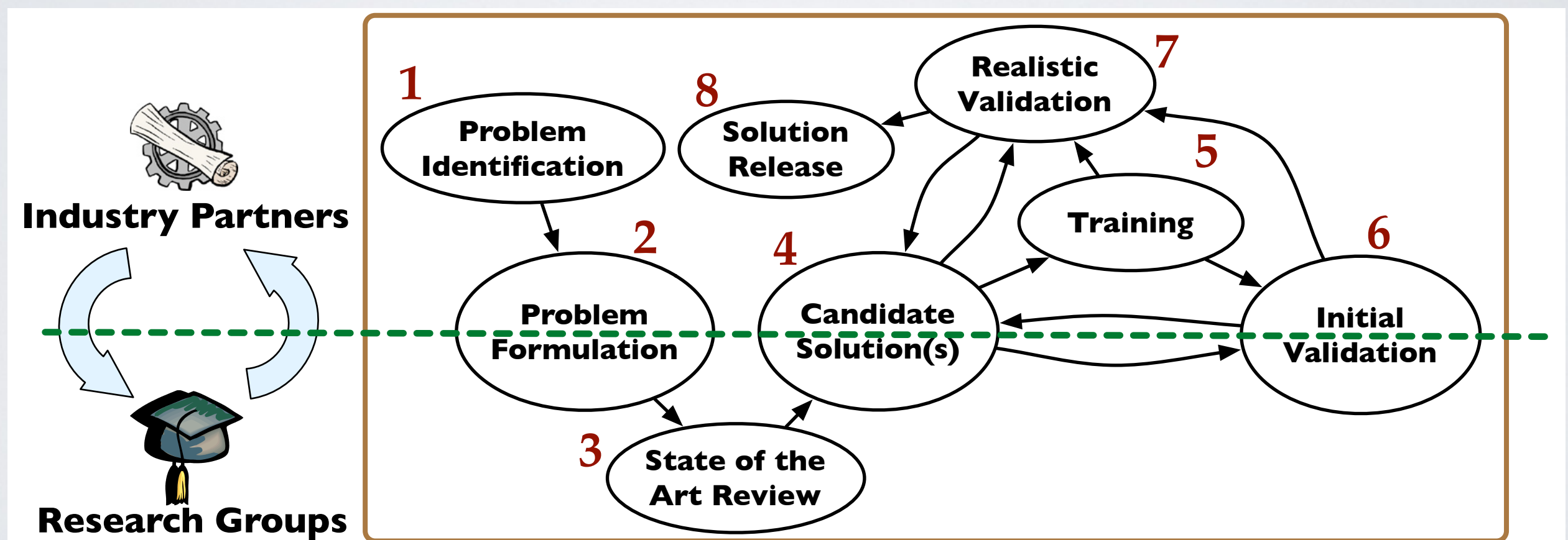
- **Report on a variety of research projects**
- **Collaborations with industry**
- **Various objectives and applications**
- **Examples from automotive and satellite**
- **Lessons learned and reflections**

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

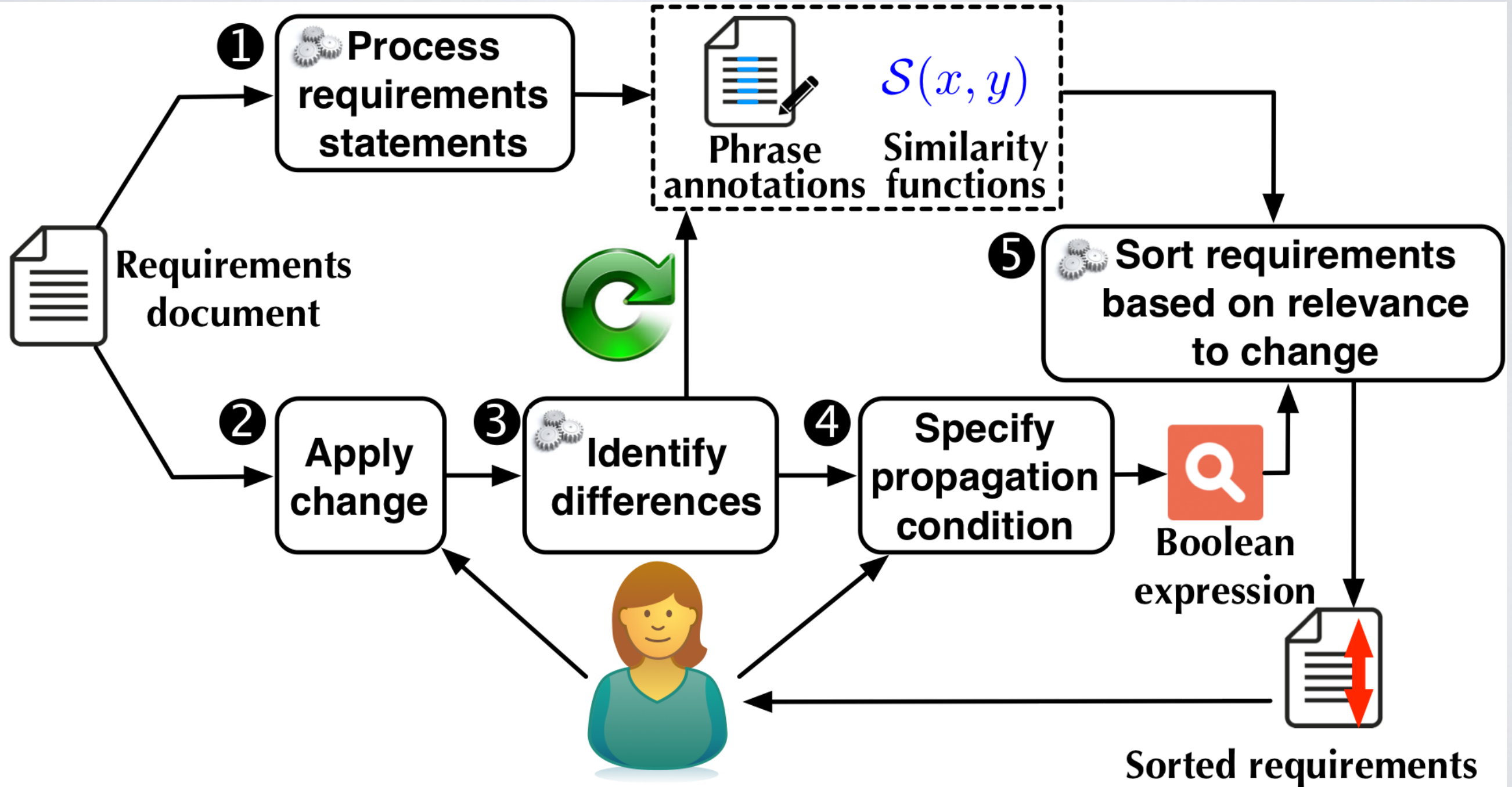
- Account for the **phrasal structure of requirements**

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



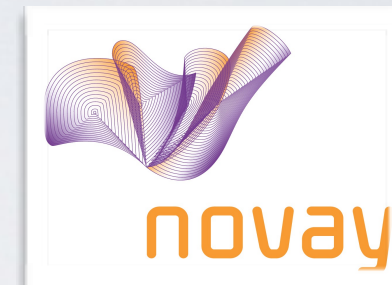
Evaluation



1 case study



**158 Requirements
9 change scenarios**

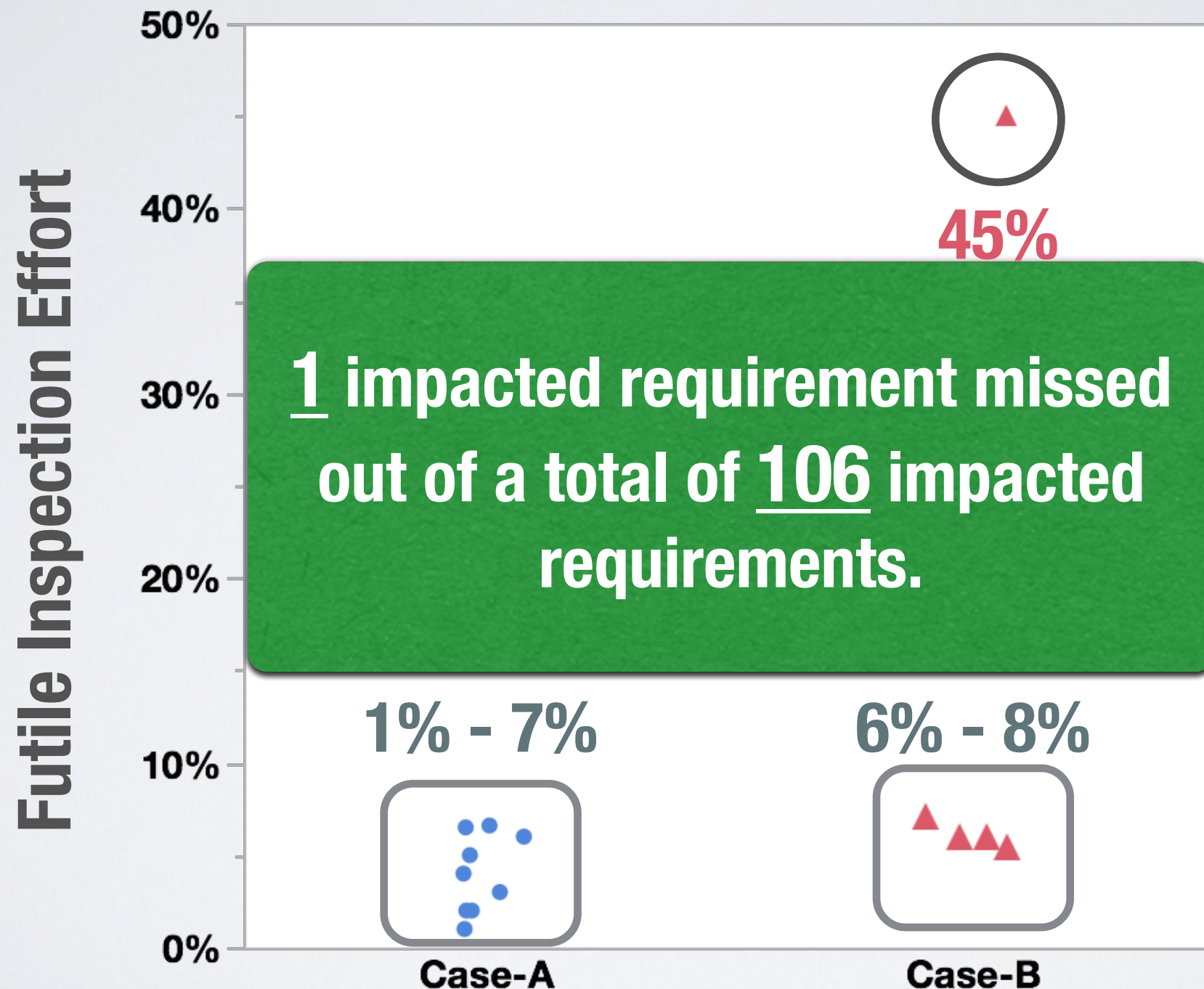


1 case study

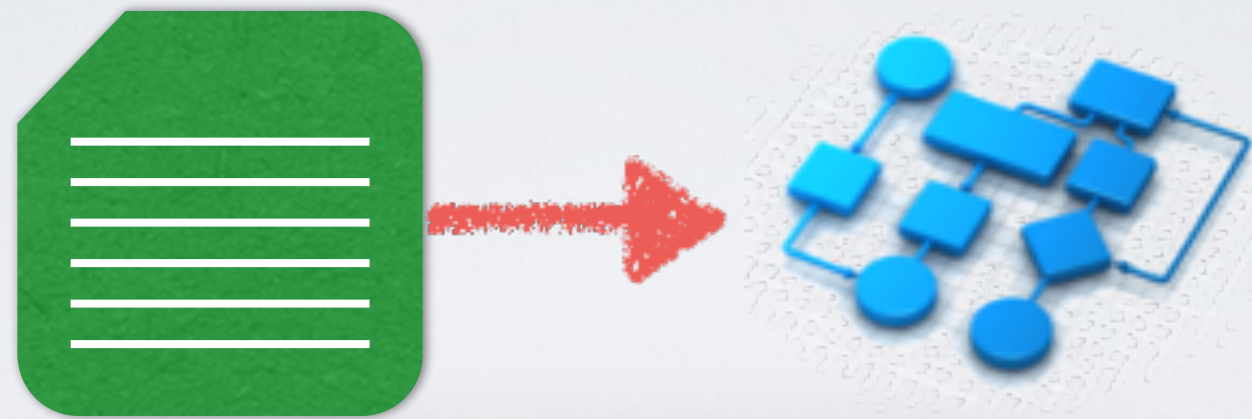


**72 Requirements
5 Change
Scenarios**

Effectiveness of Our Approach



Requirements to Design

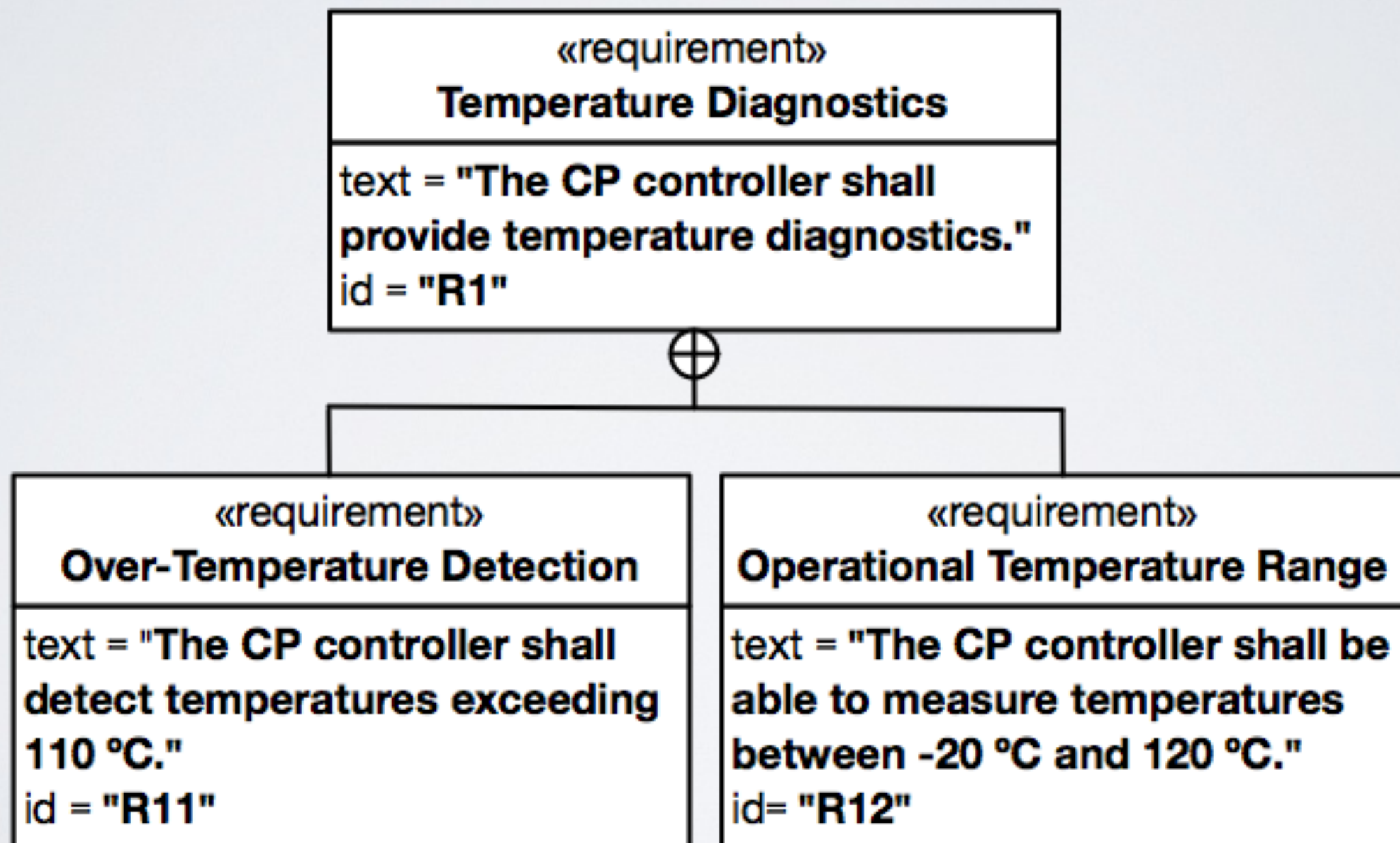


**Requirements-to-Design
Change Impact Analysis**

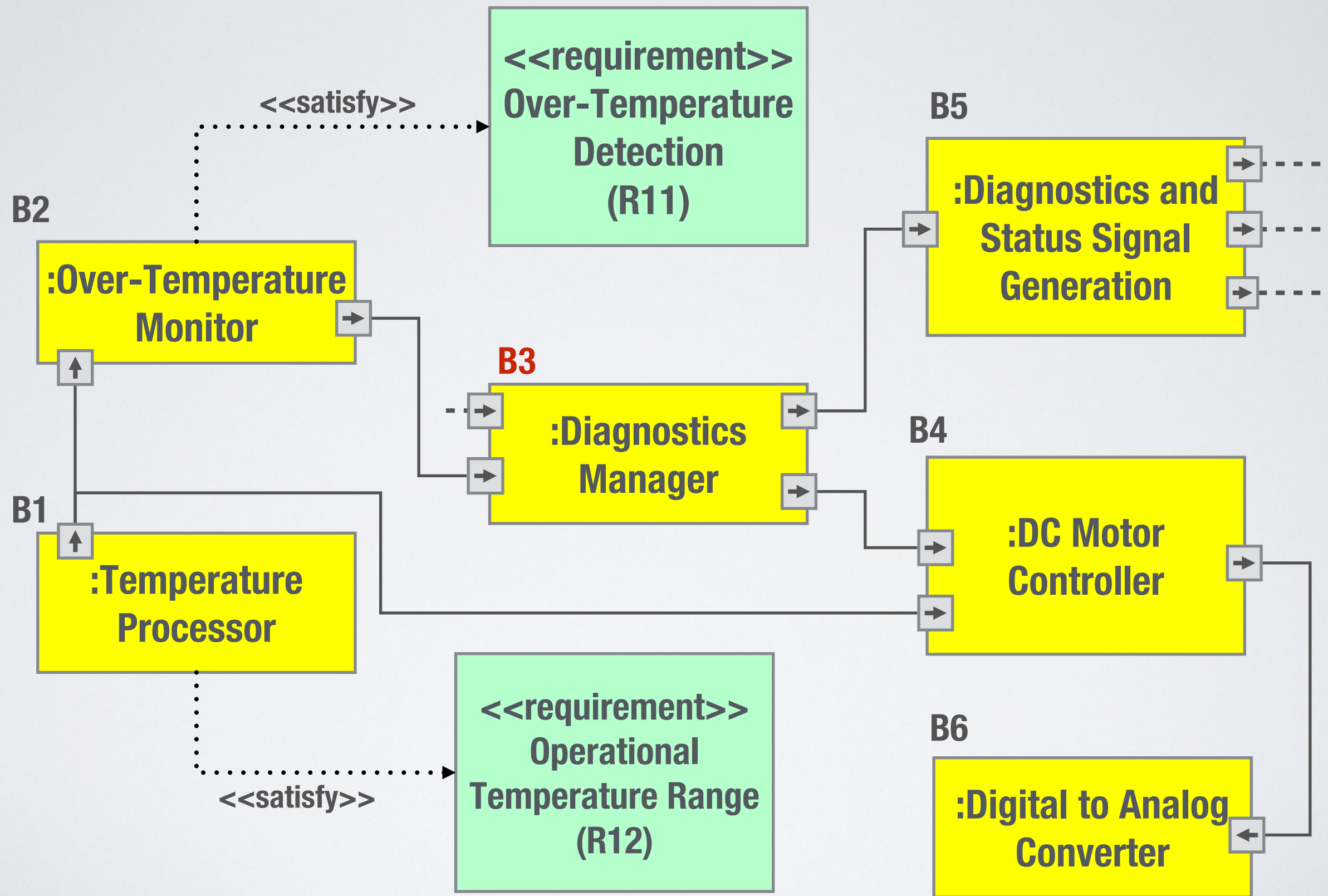
Motivations

- **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



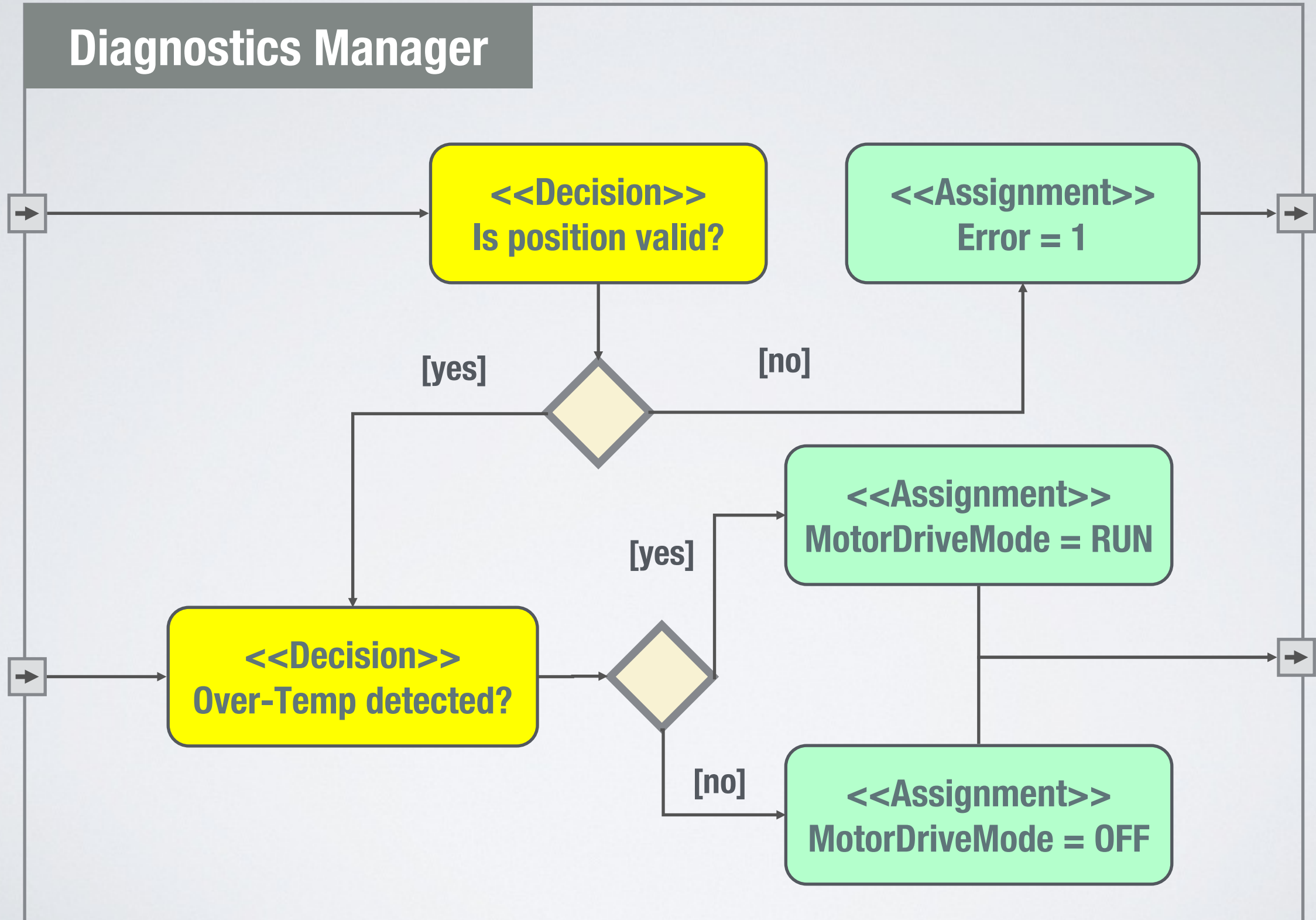
Structural Diagram



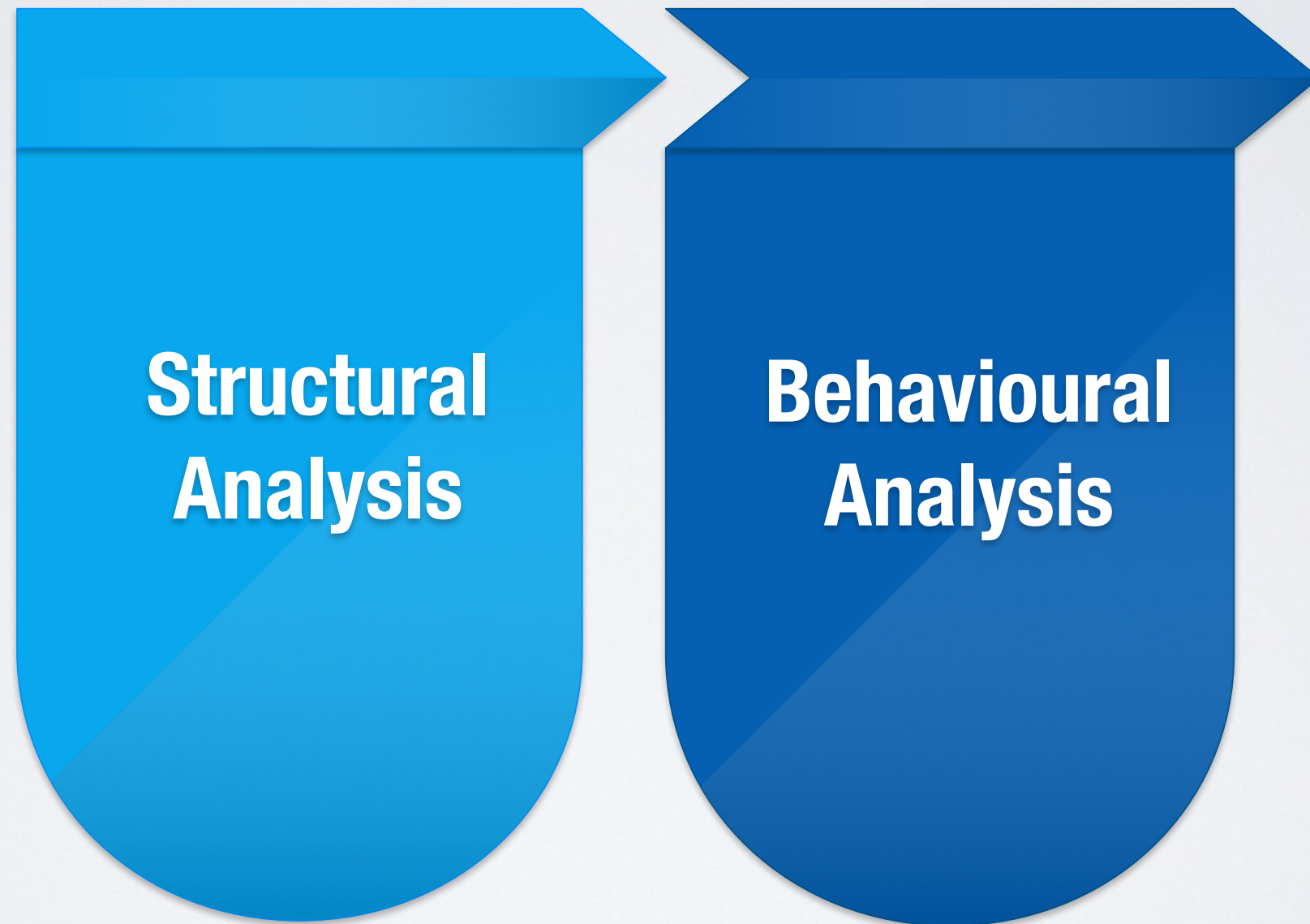
Behavioural Diagram

B3

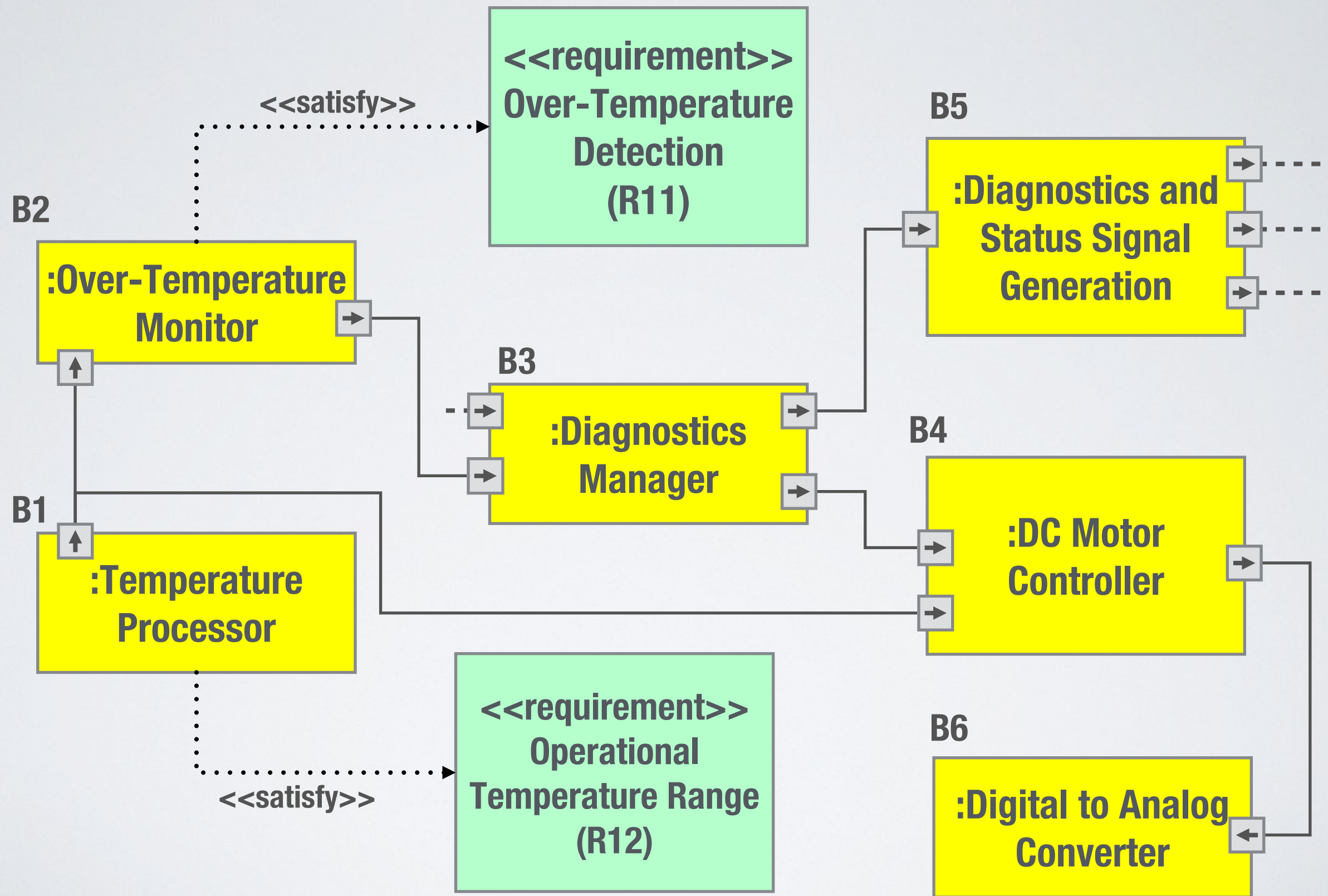
Diagnostics Manager



Compute Impacted Elements



Structural Diagram

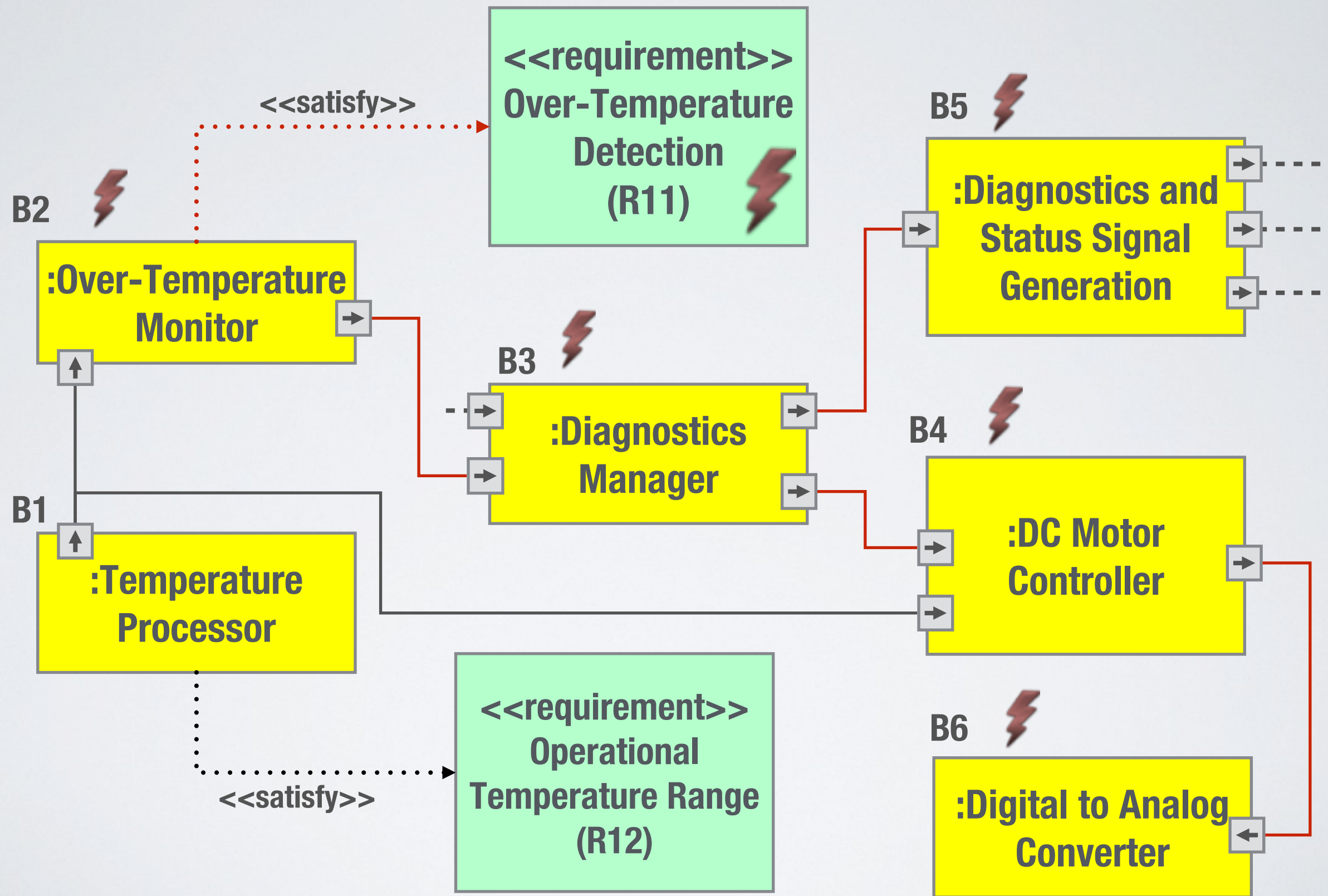


Structural Diagram

<<requirement>>

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

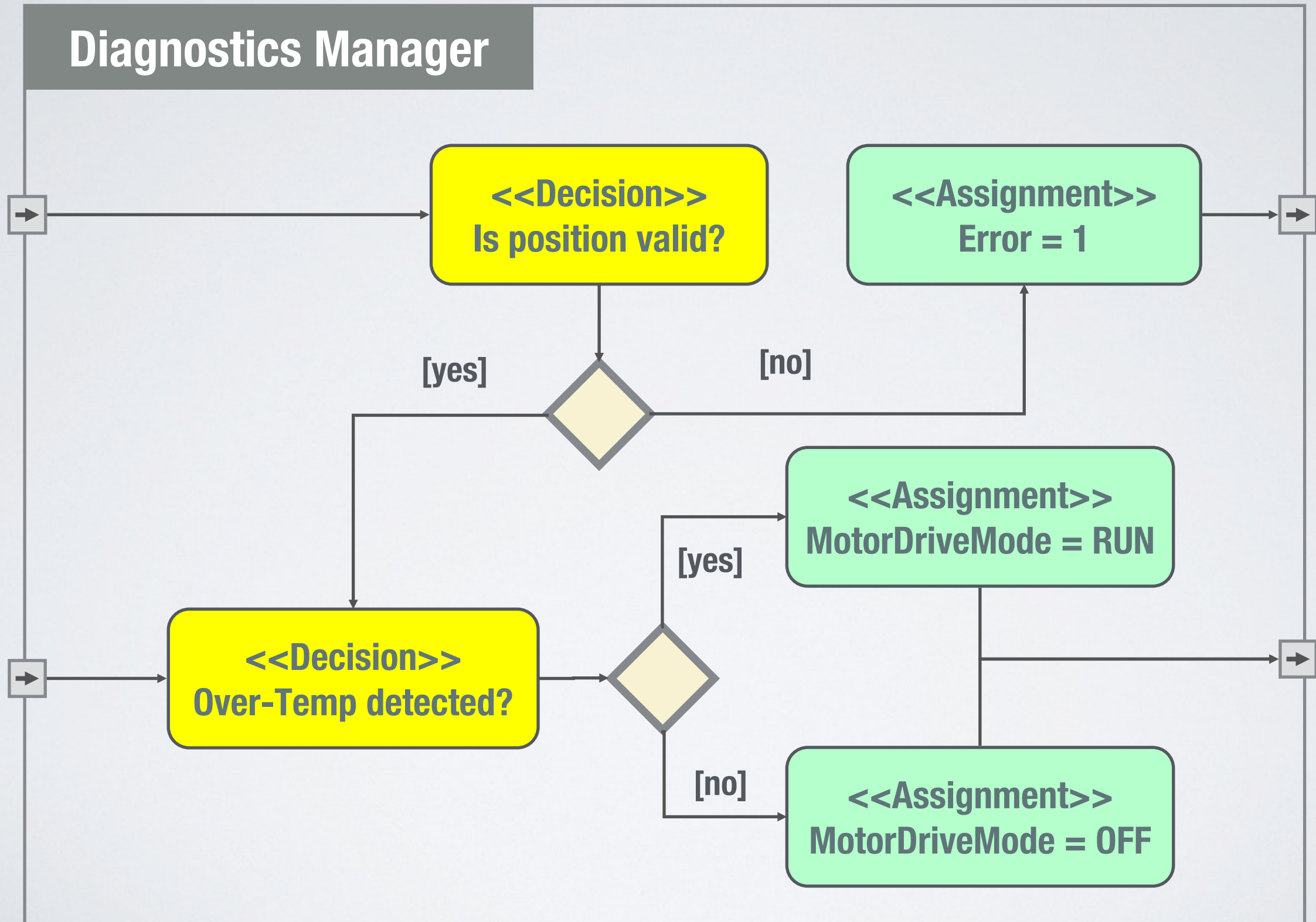
Structural Diagram



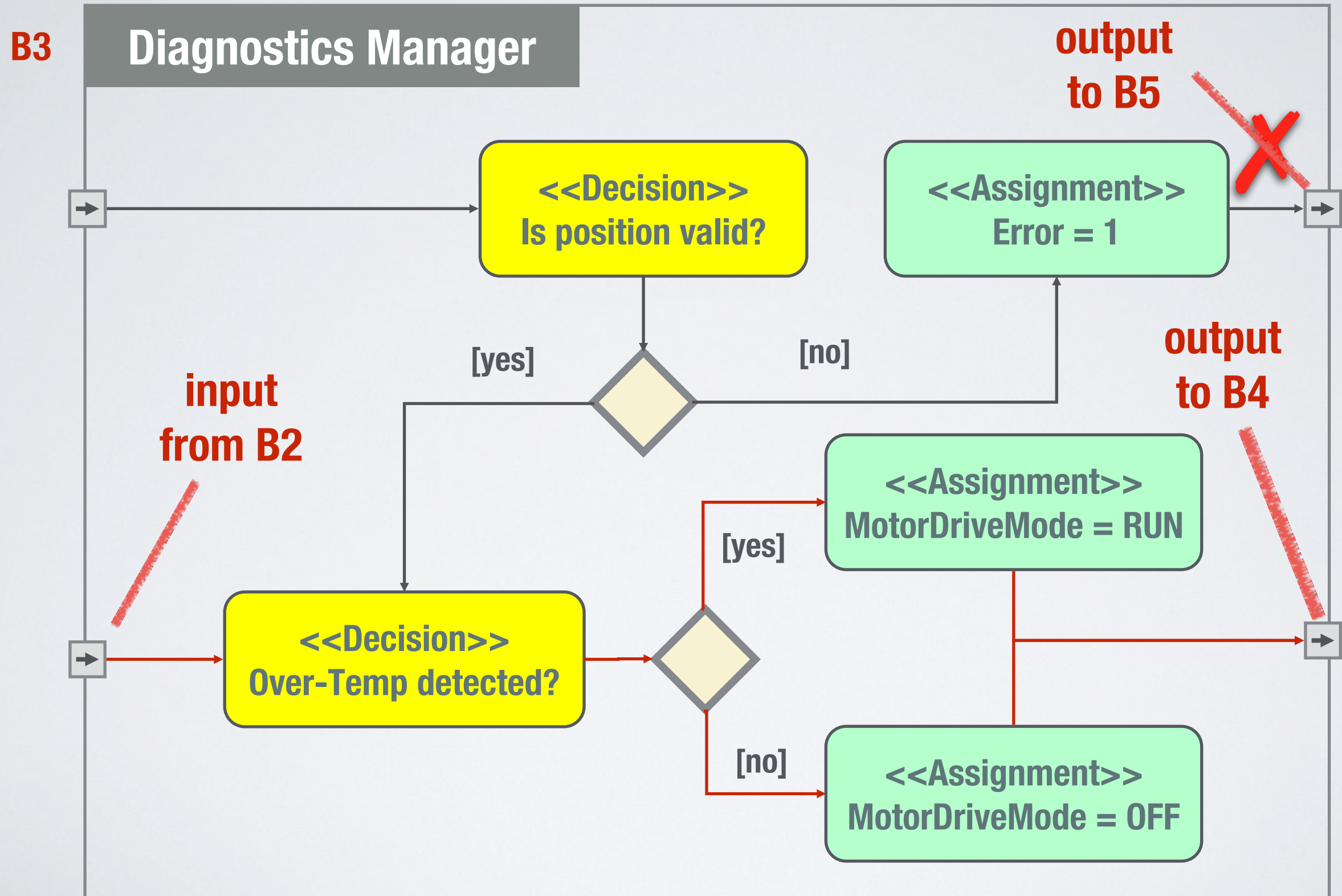
Behavioural Diagram

B3

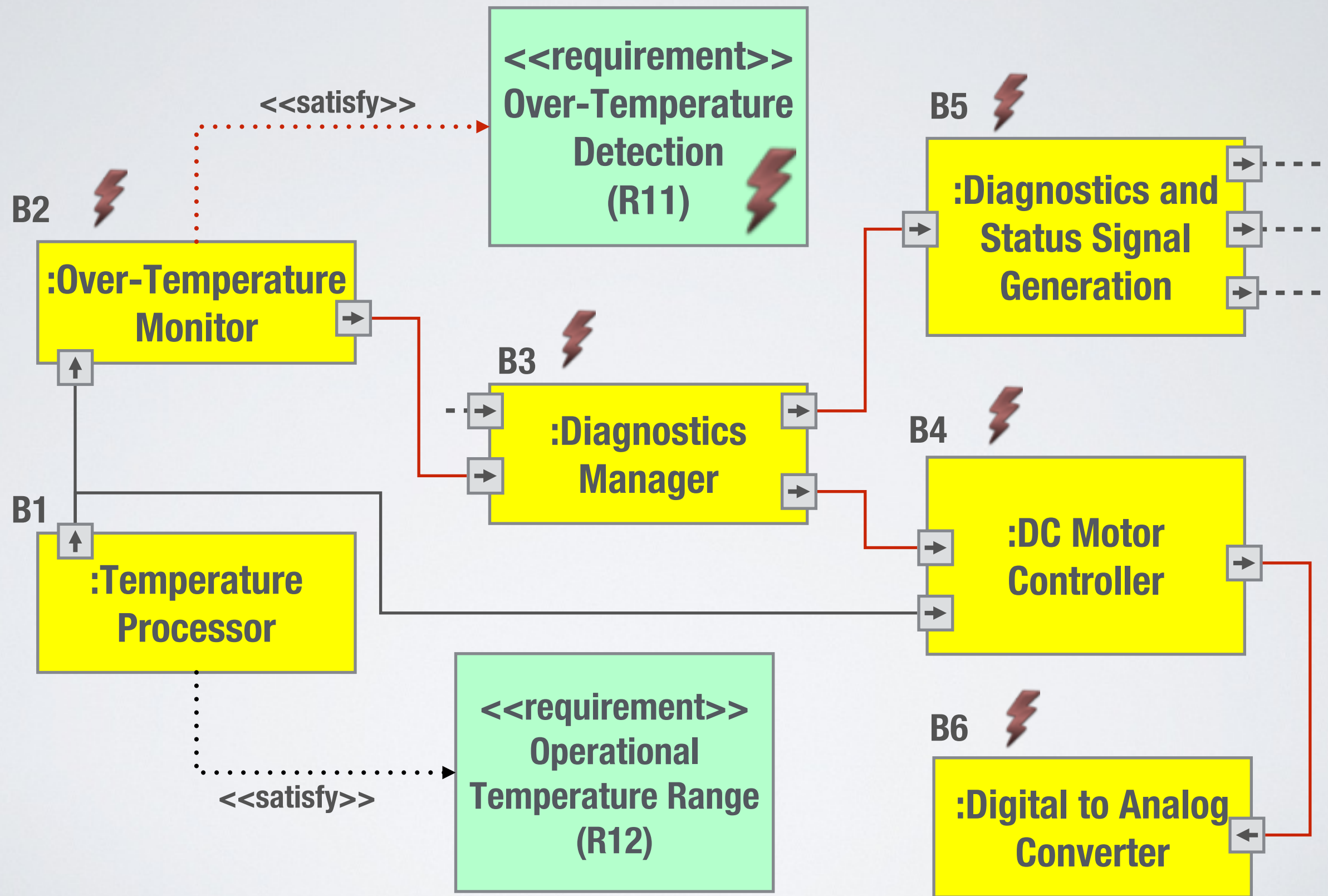
Diagnostics Manager



Behavioural Diagram



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

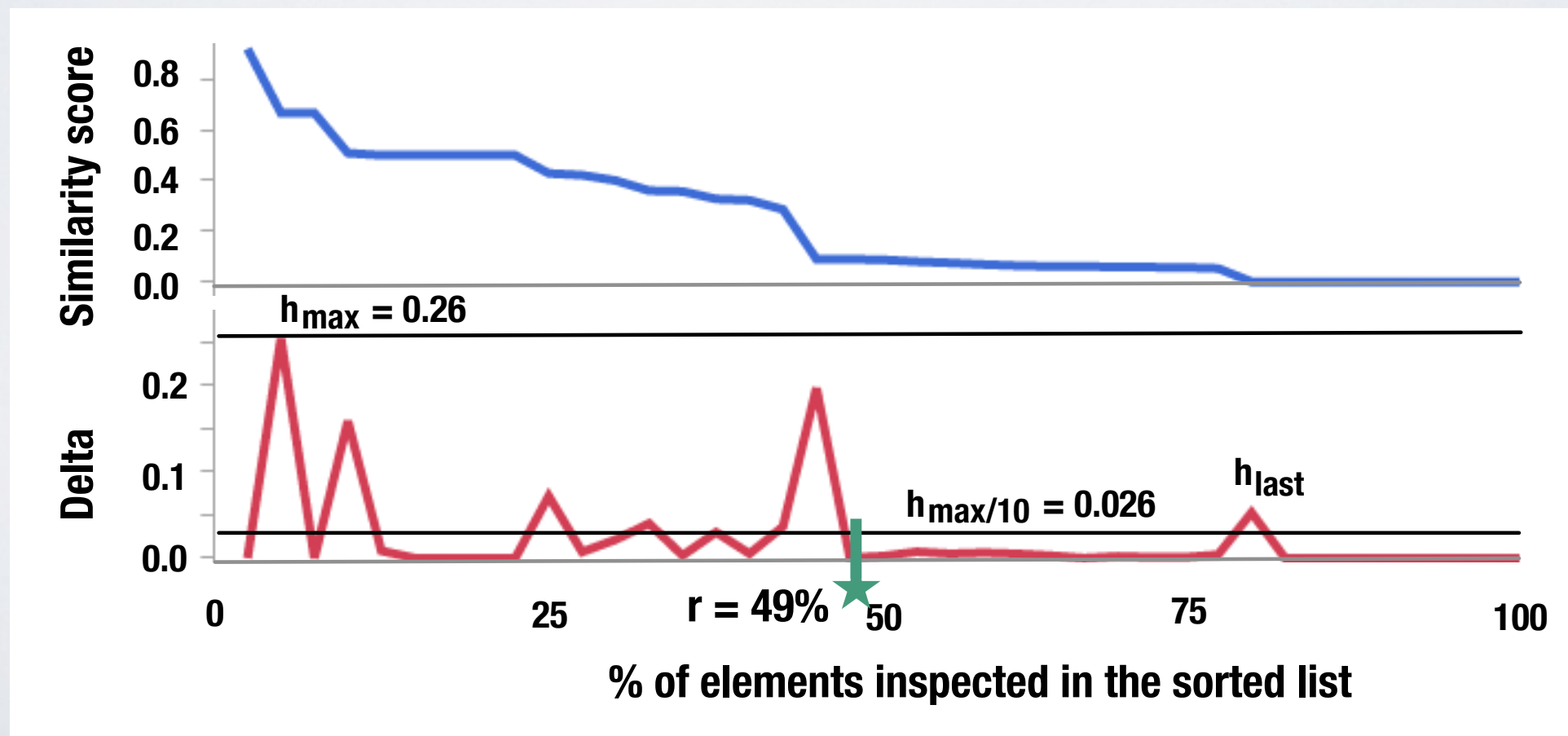
- **Informal inputs** from systems engineers regarding impact of changes
- **Example:** “Temperature lookup tables and voltage converters need to be adjusted”

Natural Language Processing

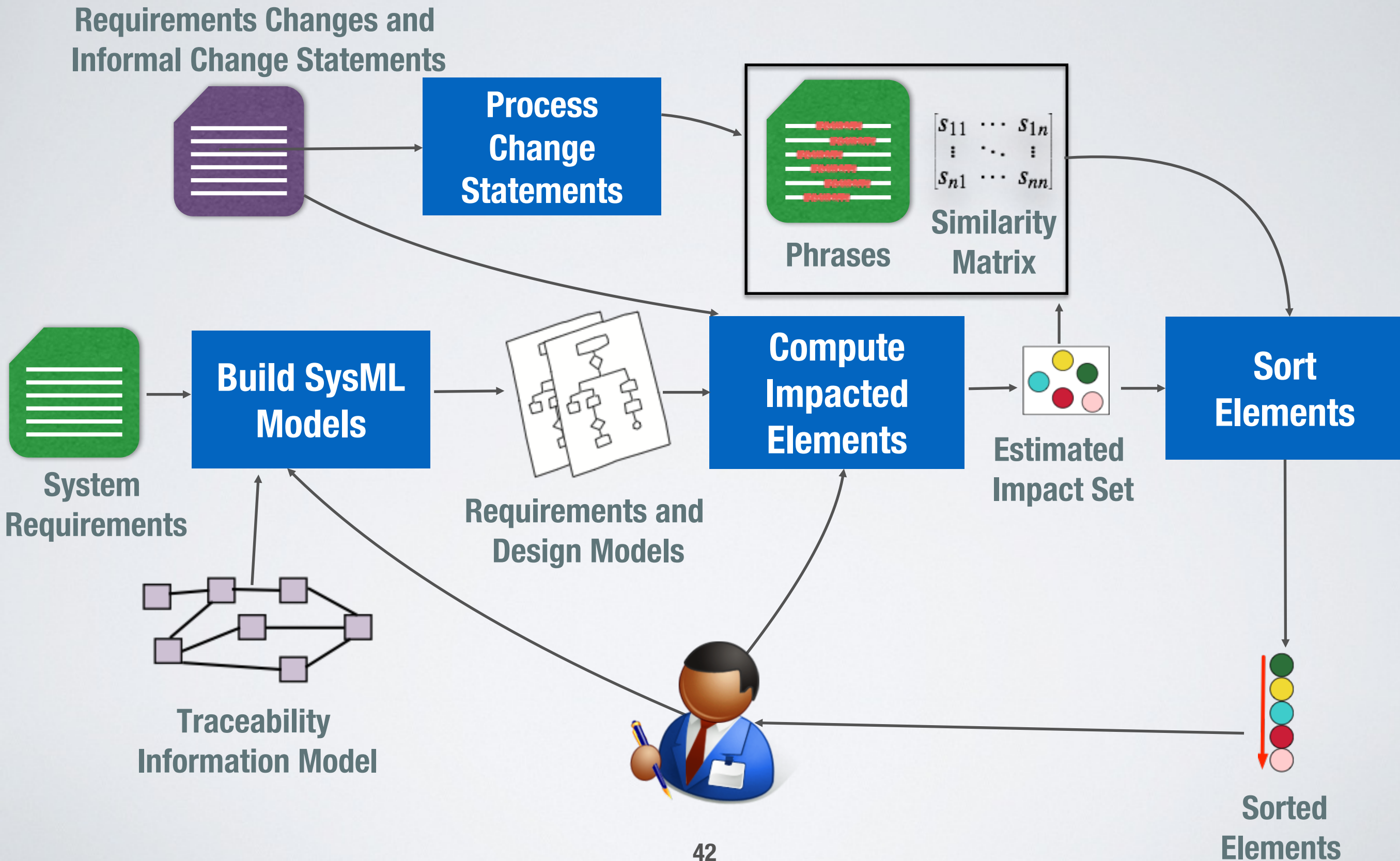
- Compute **similarity scores** (syntactic and semantic) **between model elements labels and change statements**
- Experimented with **pairwise combinations** of syntactic and semantic measures
- **Sort** the design elements obtained after structural and behavioral analysis based on the similarity scores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

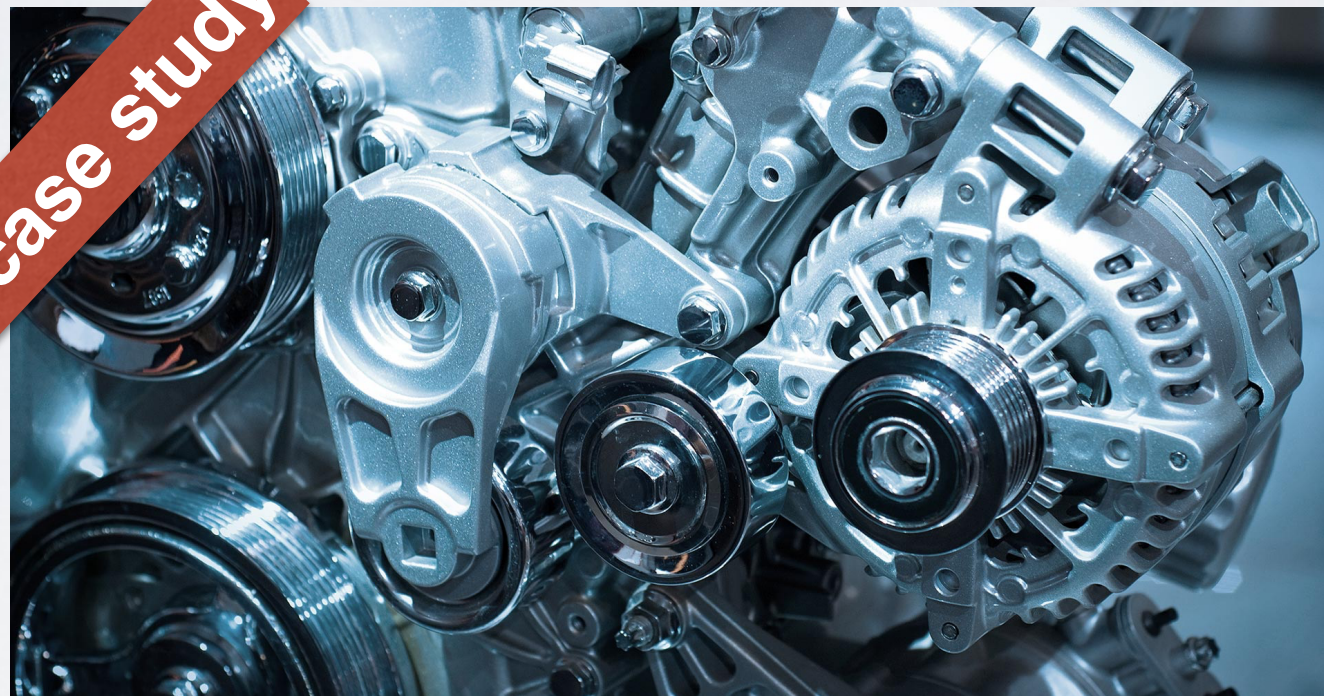


Evaluation

DELPHI

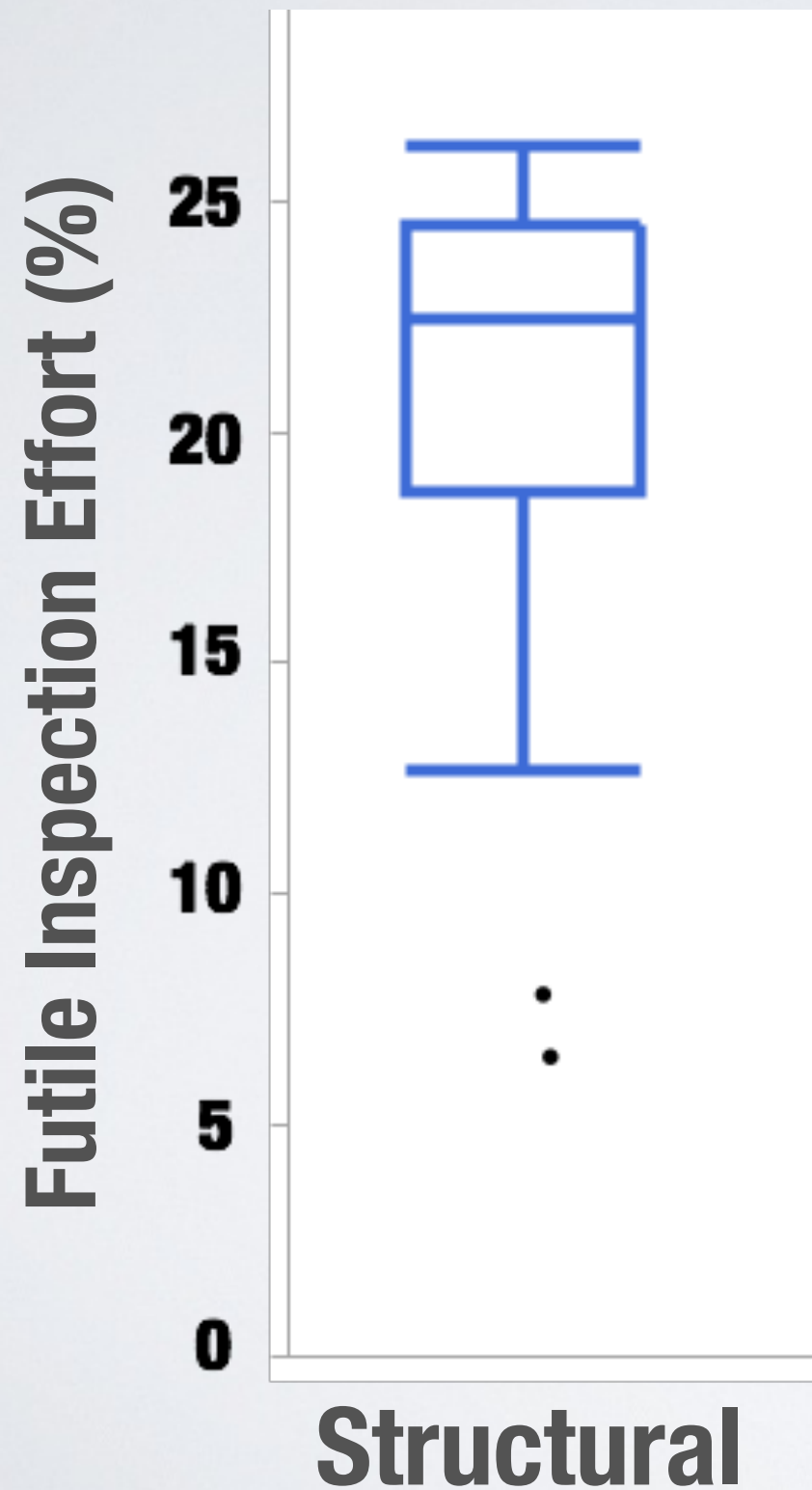
Innovation for the Real World

1 case study

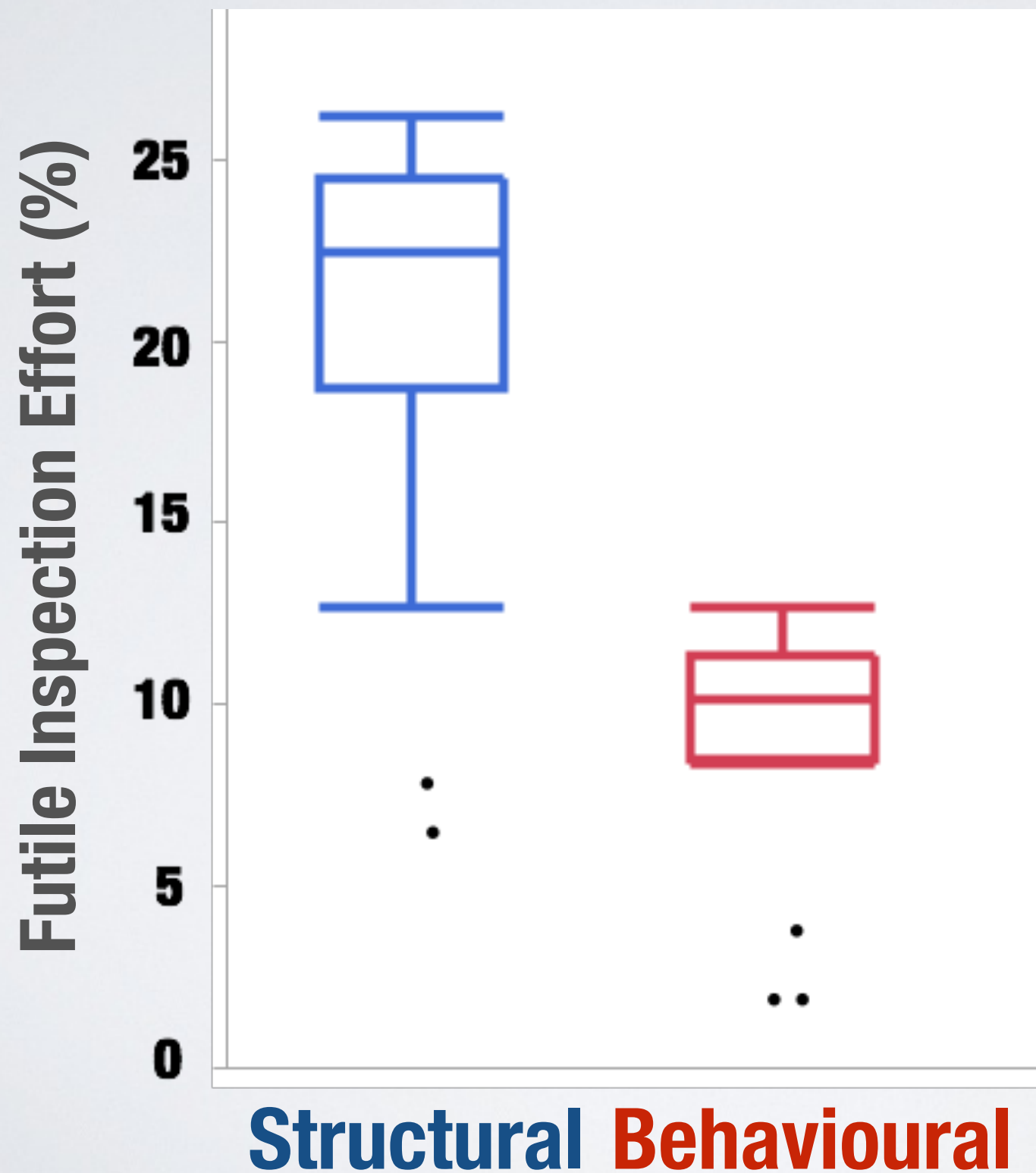


370 elements
16 change scenarios

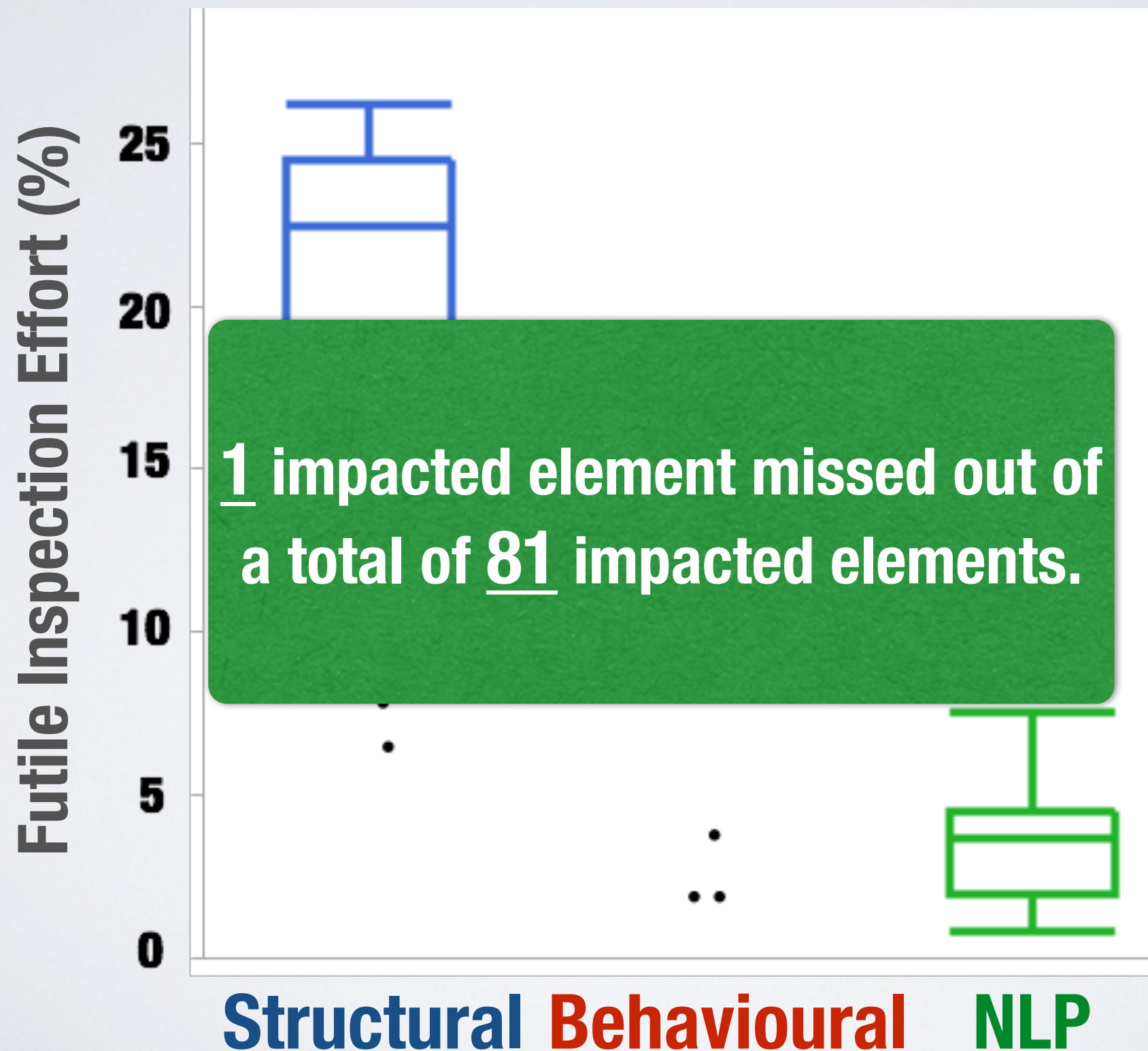
Effectiveness of Our Approach



Effectiveness of Our Approach



Effectiveness of Our Approach



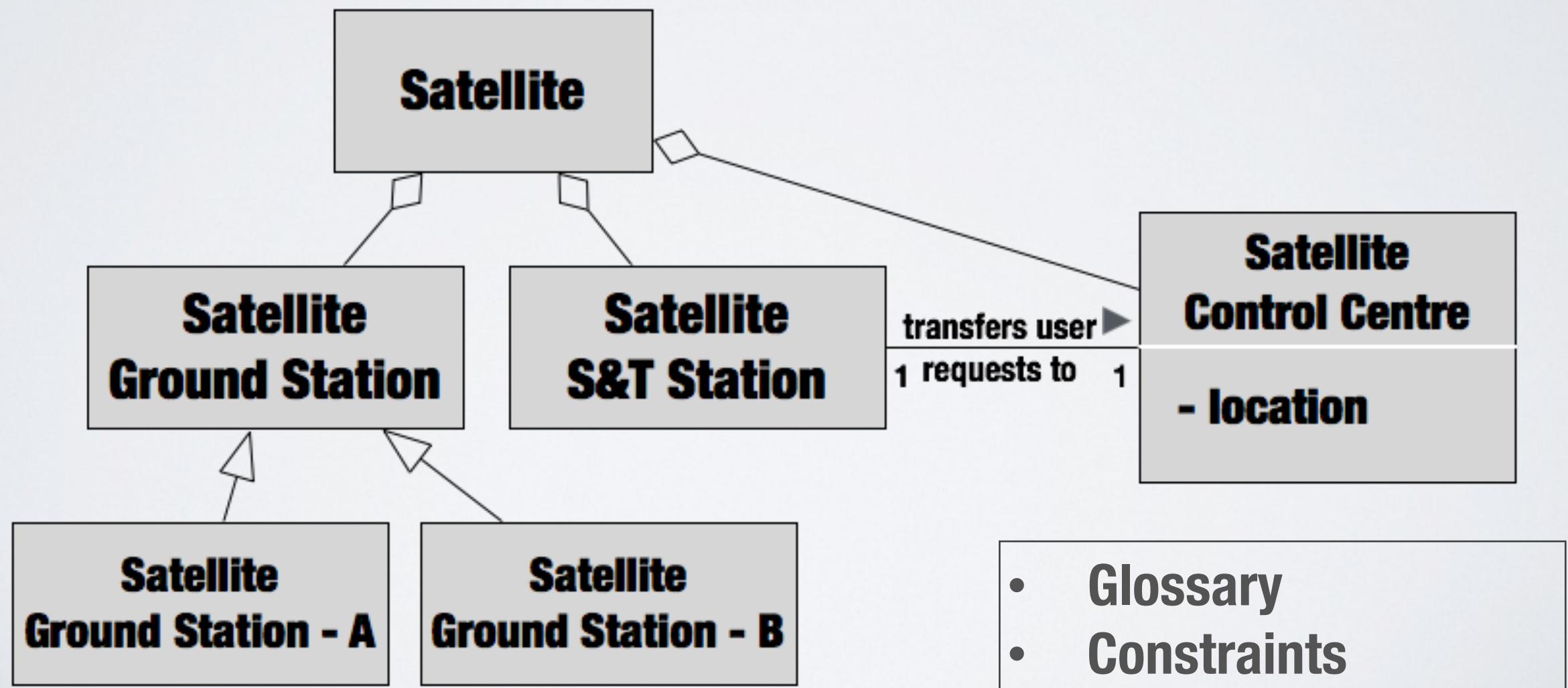
Extracting Domain Knowledge

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

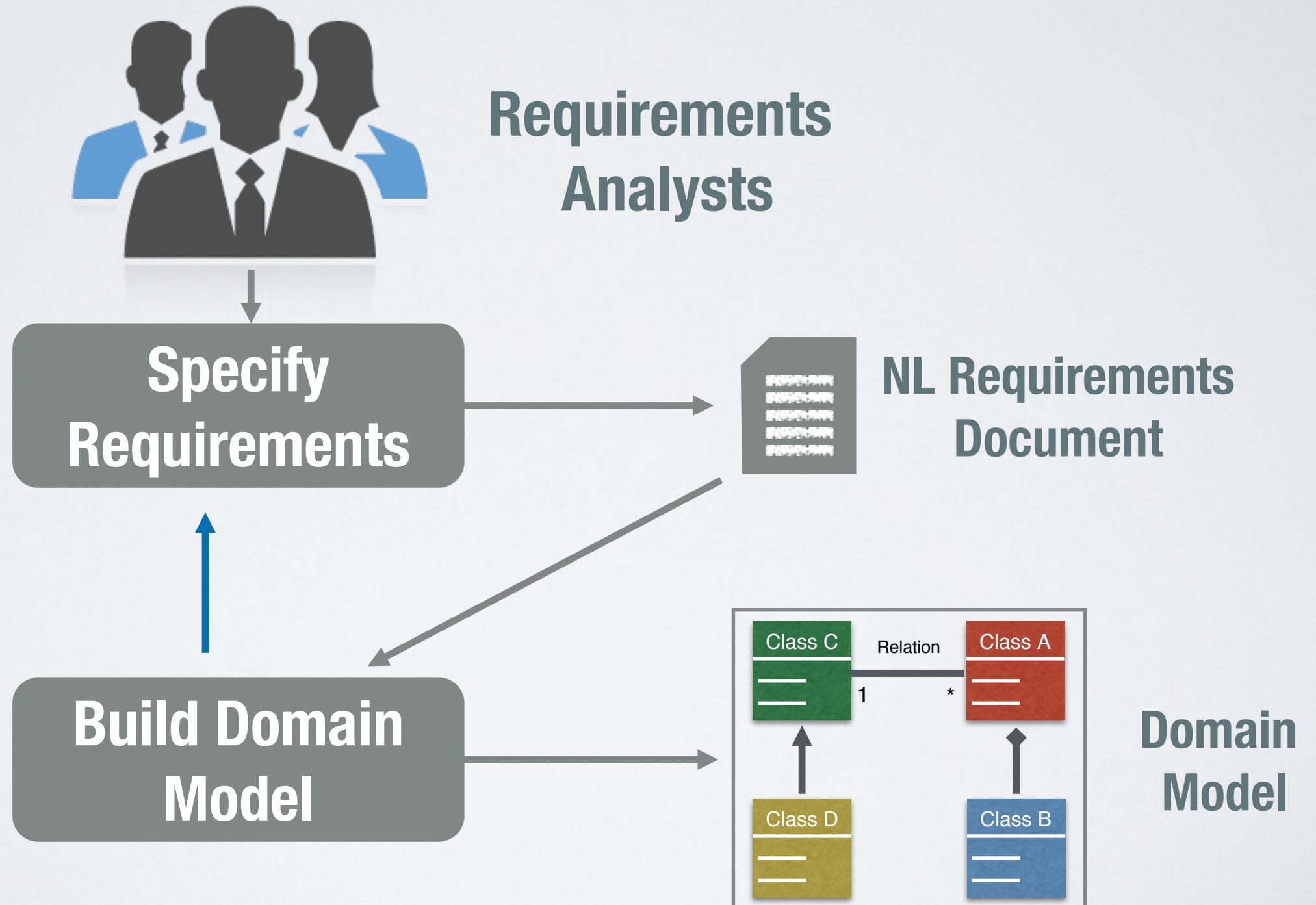
A domain model is a representation of conceptual entities or real-world objects in a domain of interest



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

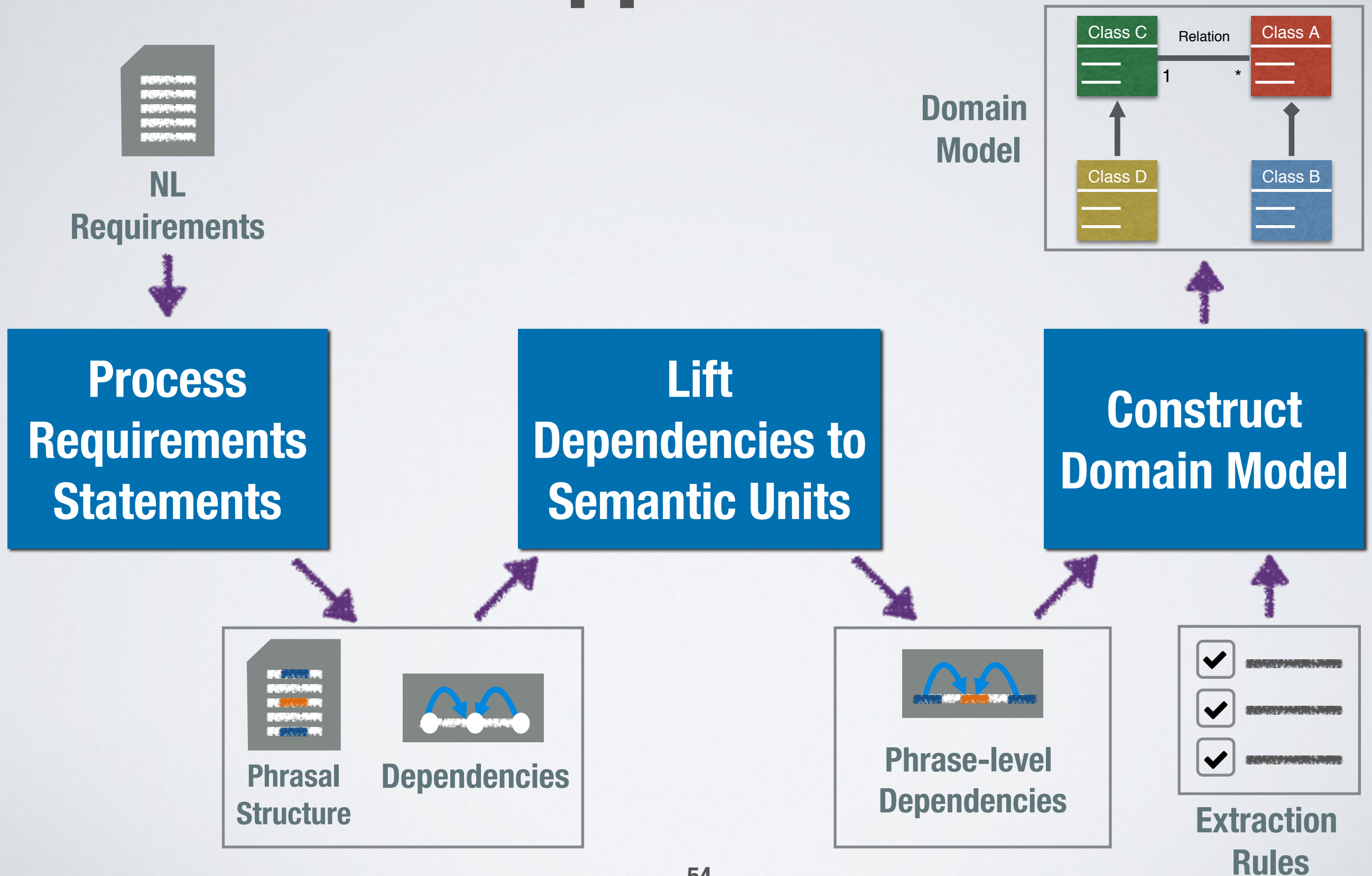


- Manually building domain models is **laborious**
- **Automated support is required** for building domain models

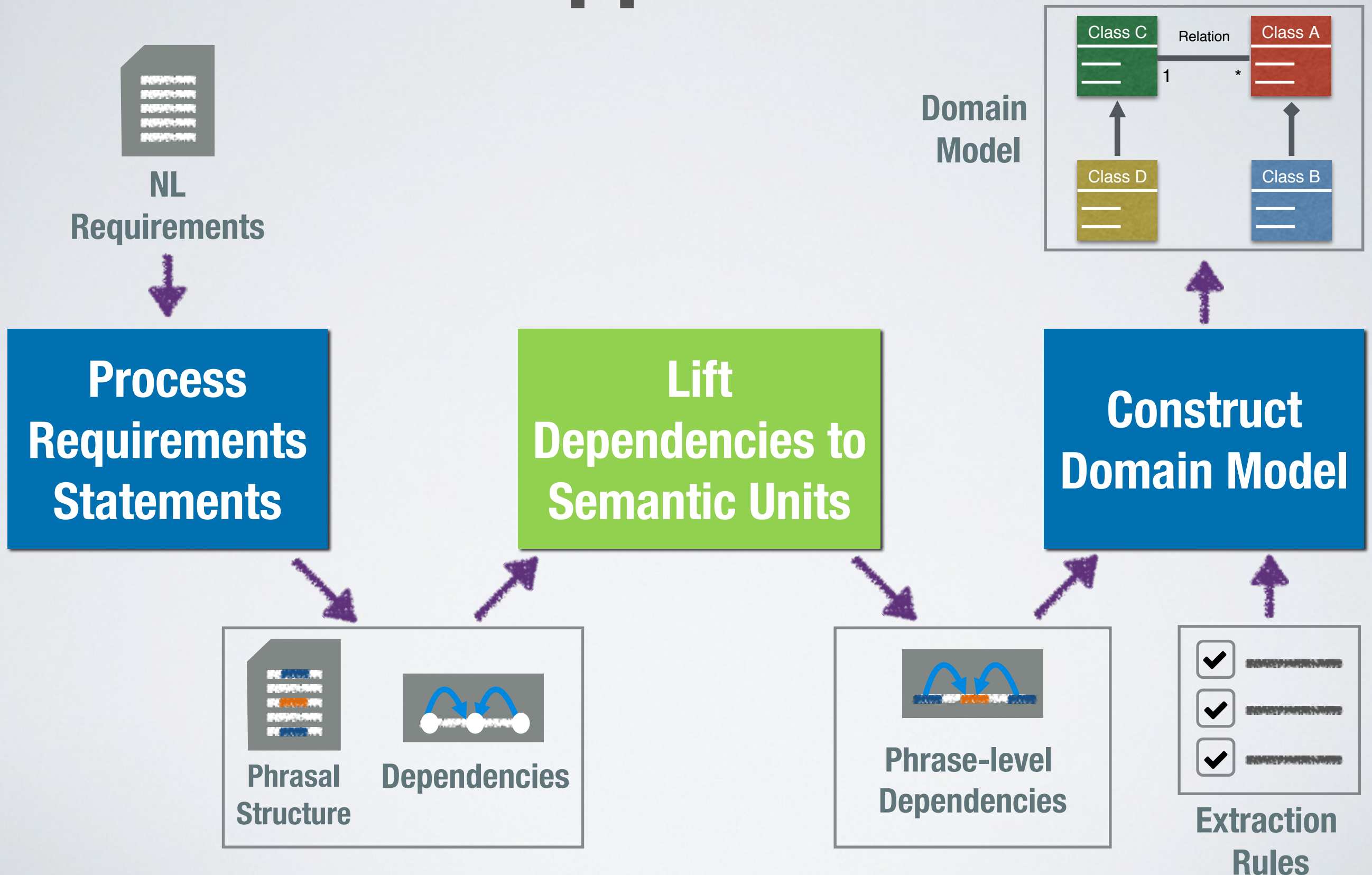
State of the Art

- **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

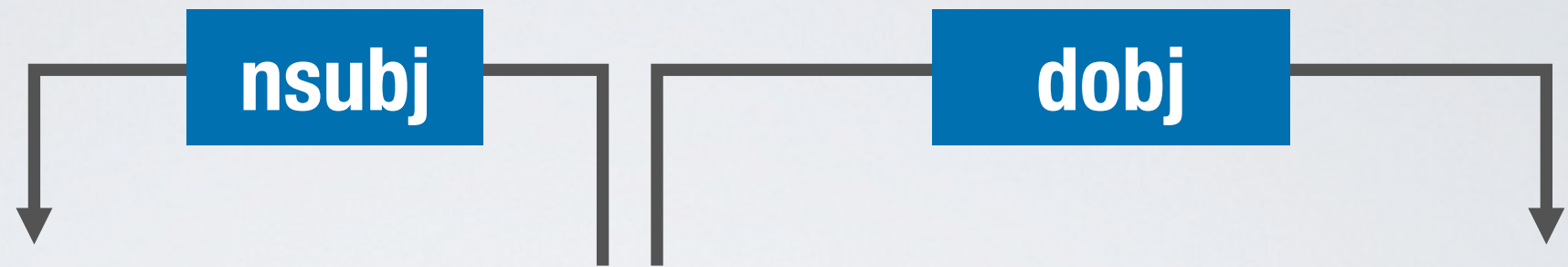
Approach



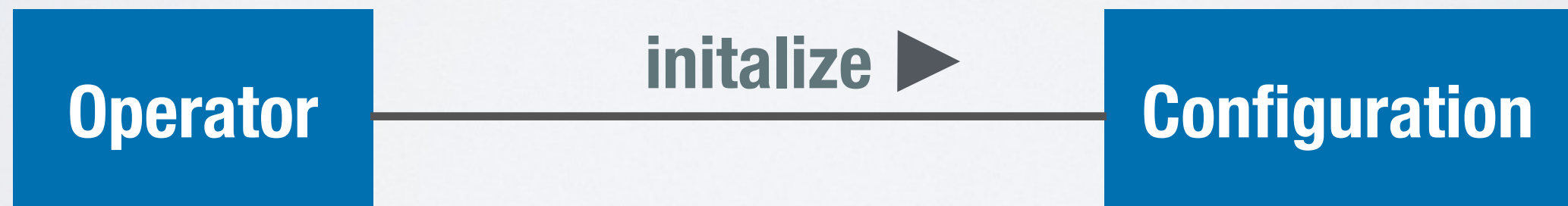
Approach



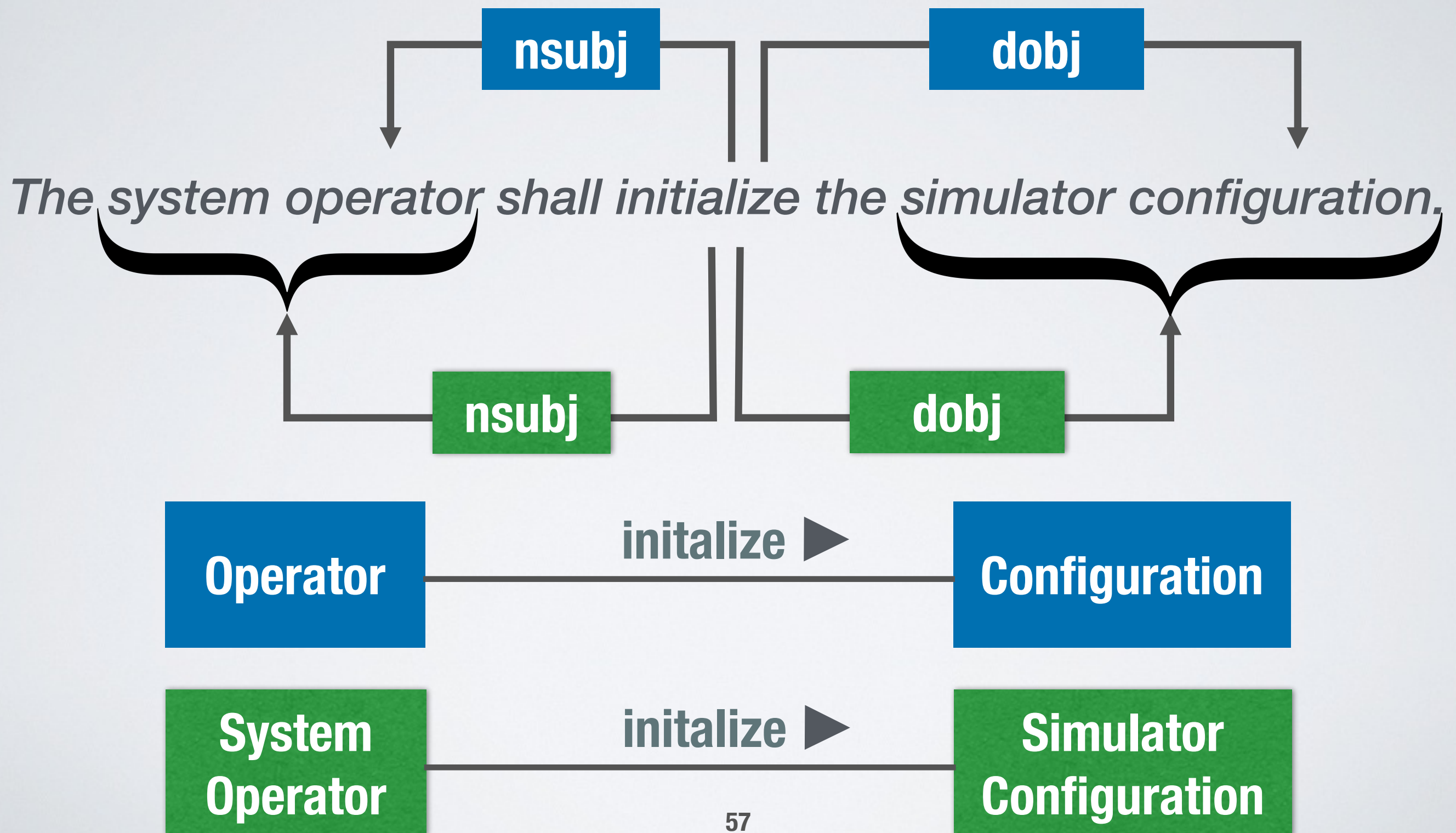
Grammatical Dependencies



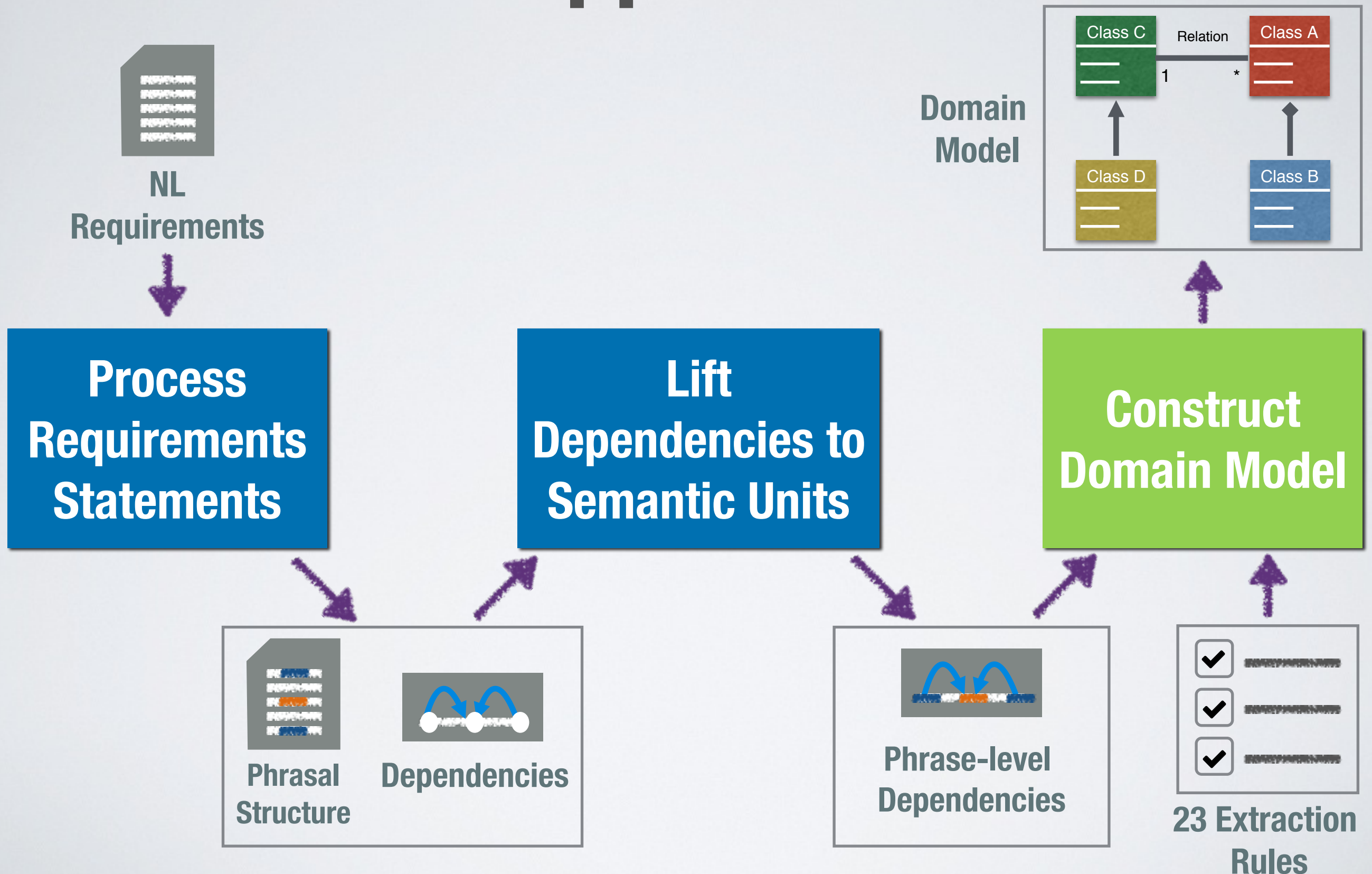
The system operator shall initialize the simulator configuration.



Lift Dependencies to Semantic Units

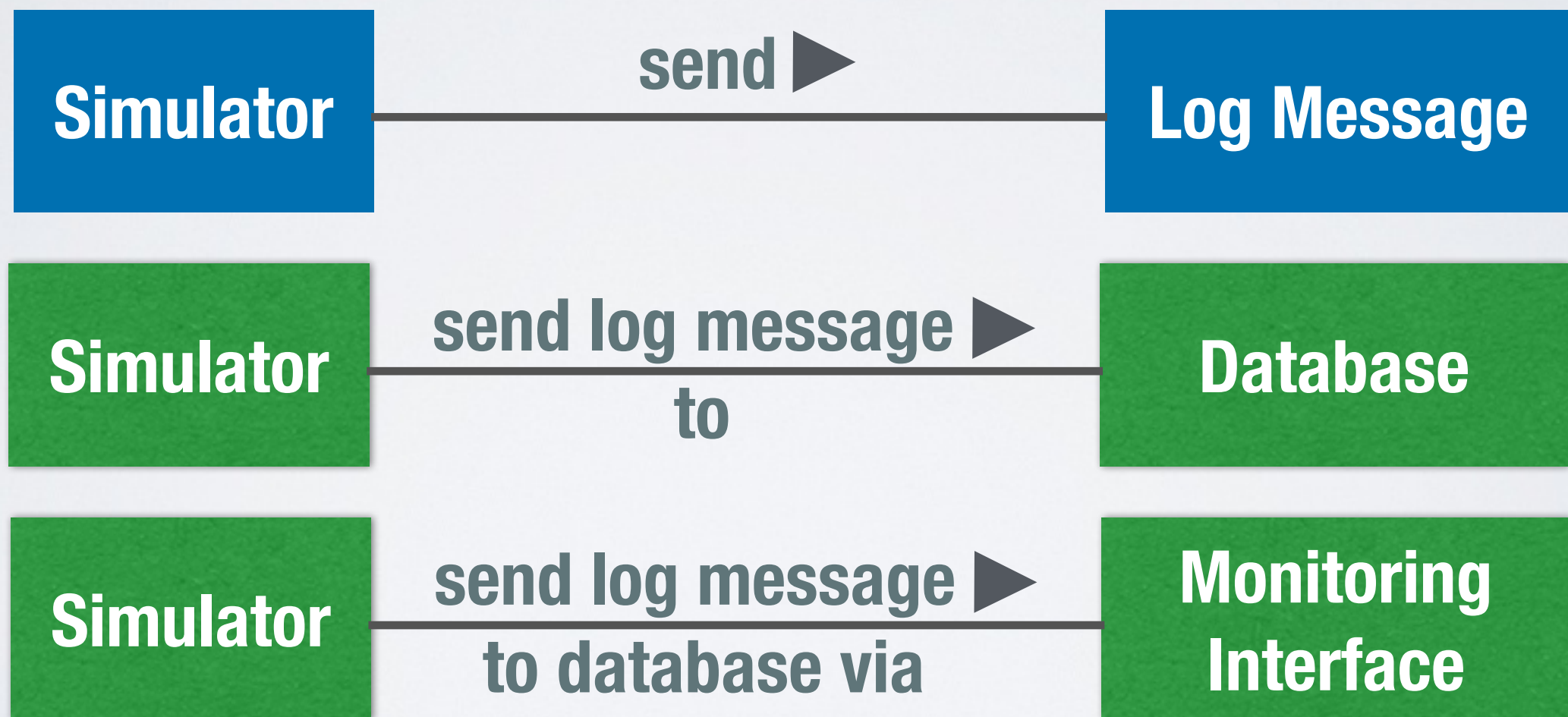


Approach



Link Paths – Indirect Relations

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



How useful is our approach?

SES[▲]
your satellite company

1 case study



**50 Requirements
213 Relations**

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

Results

Correctness: 88% (avg.)

**Incorrectness largely
explained by NLP
errors**

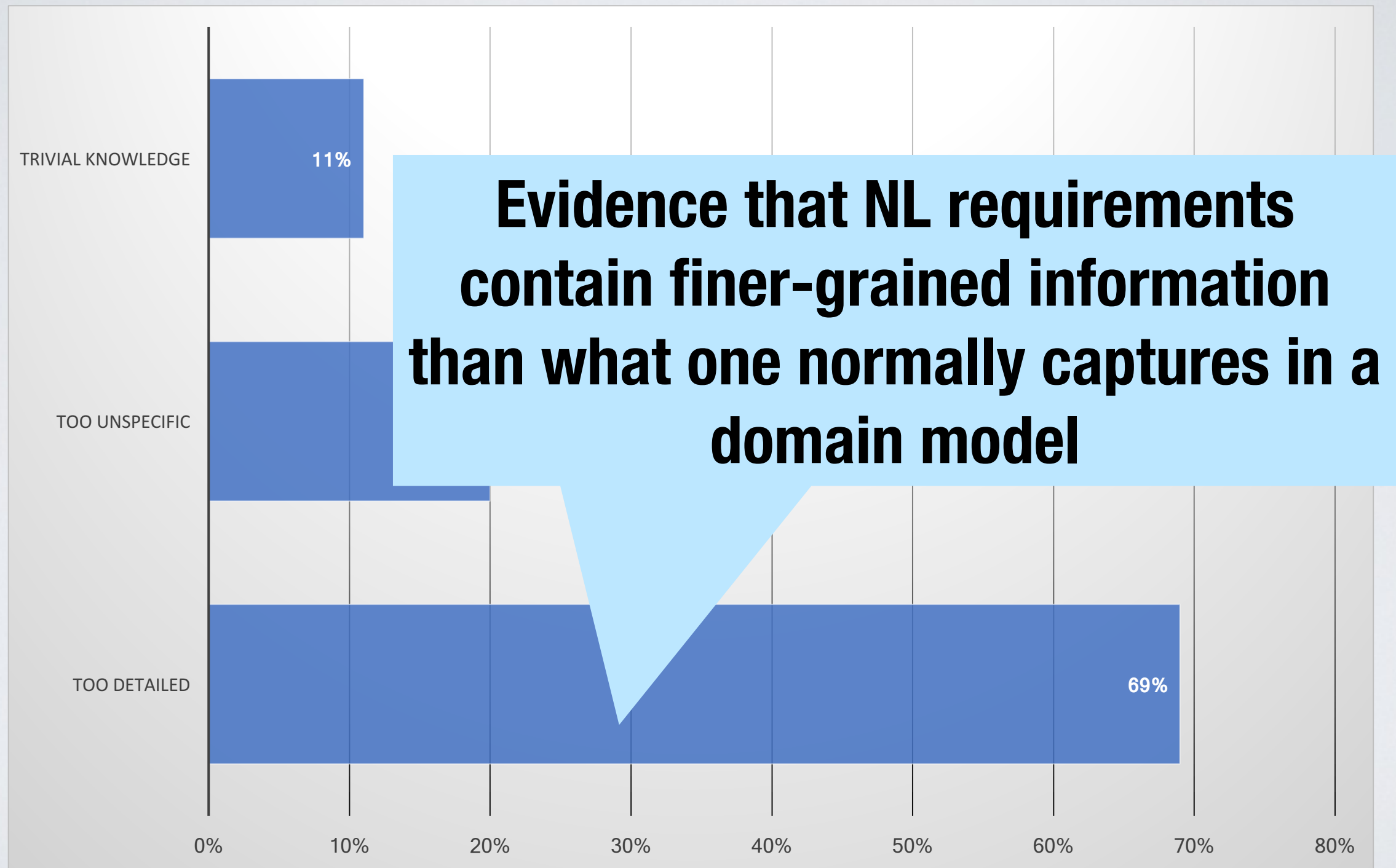
Relevance: 37% (avg.)

**Breakdown of the
remaining 63%**

Missed: 10% (avg.)

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

Statistics for Superfluouslyness

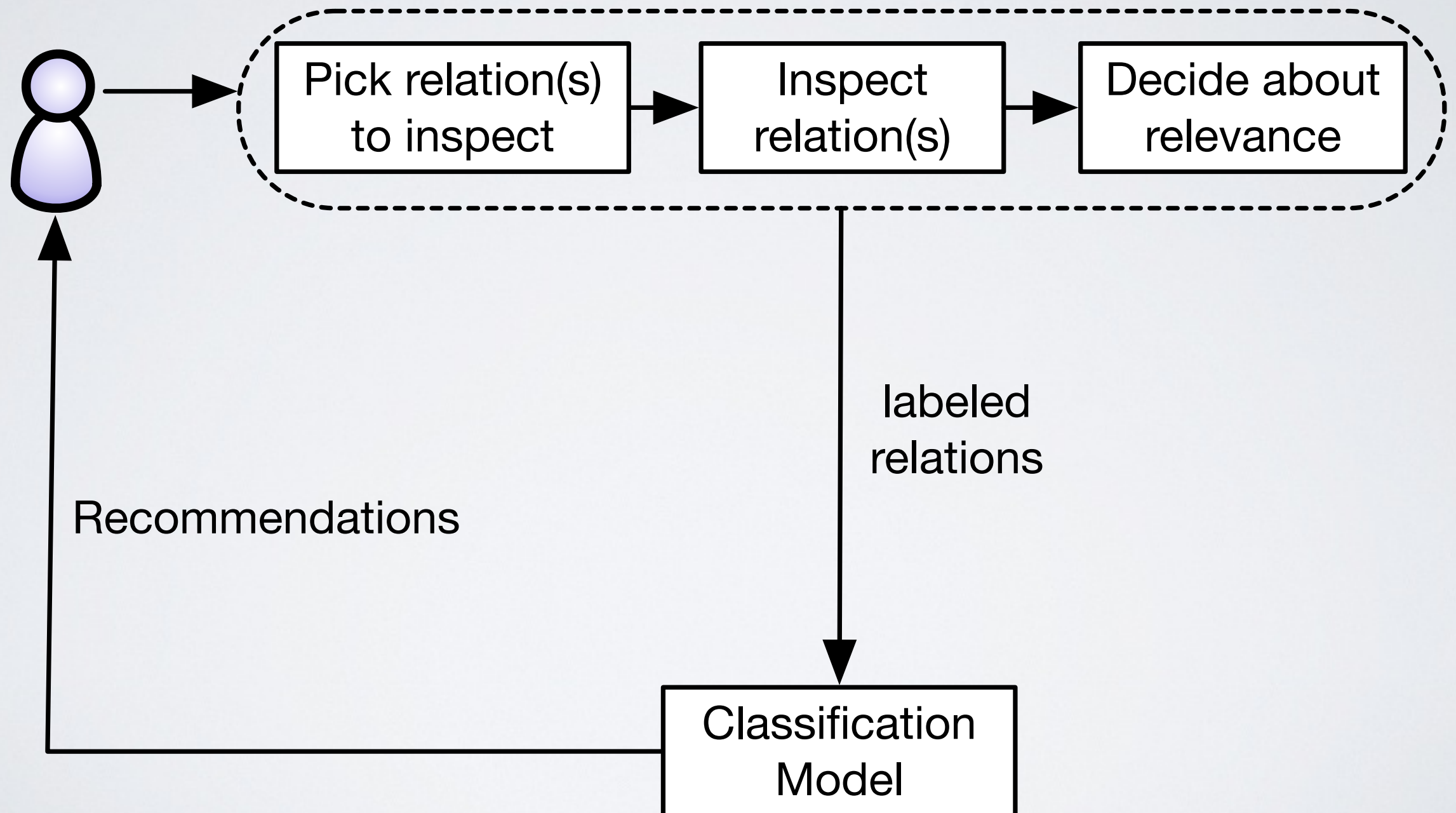


**Can we improve the relevance of
model extraction results?**

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

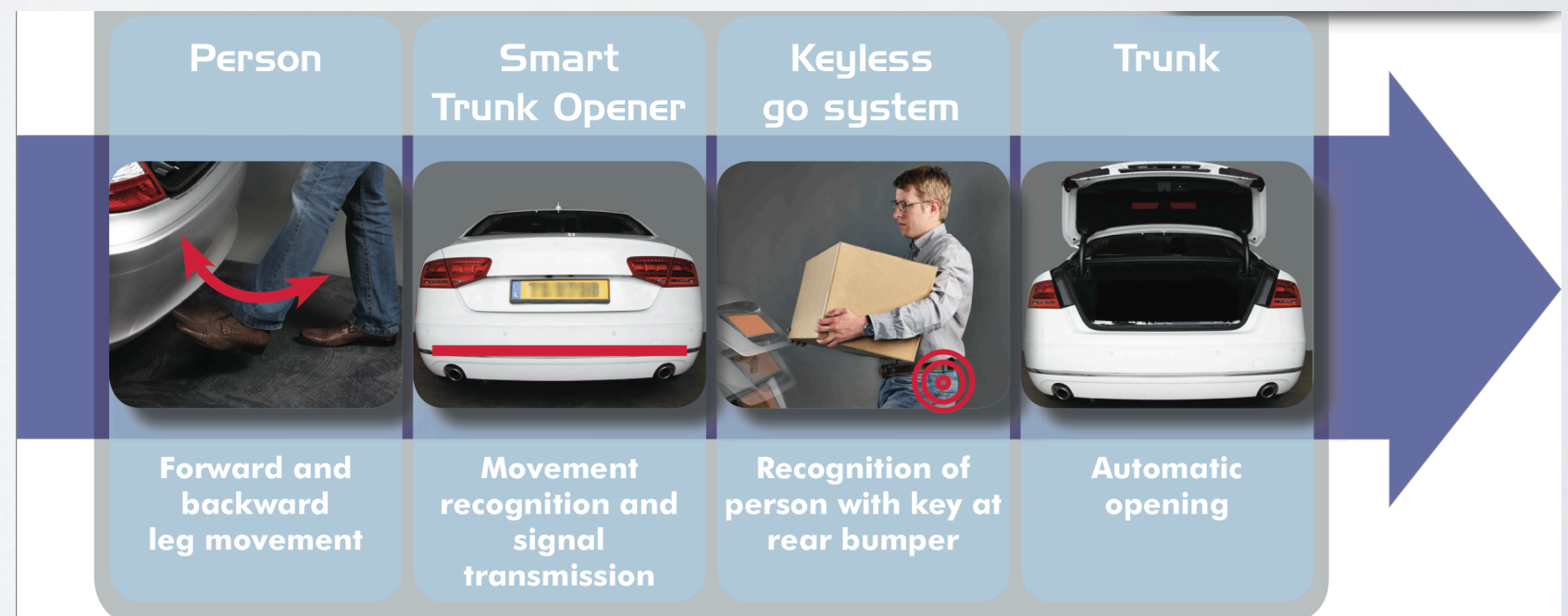
Context

IEE develops real-time embedded systems:

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



International Electronics
& Engineering (IEE)



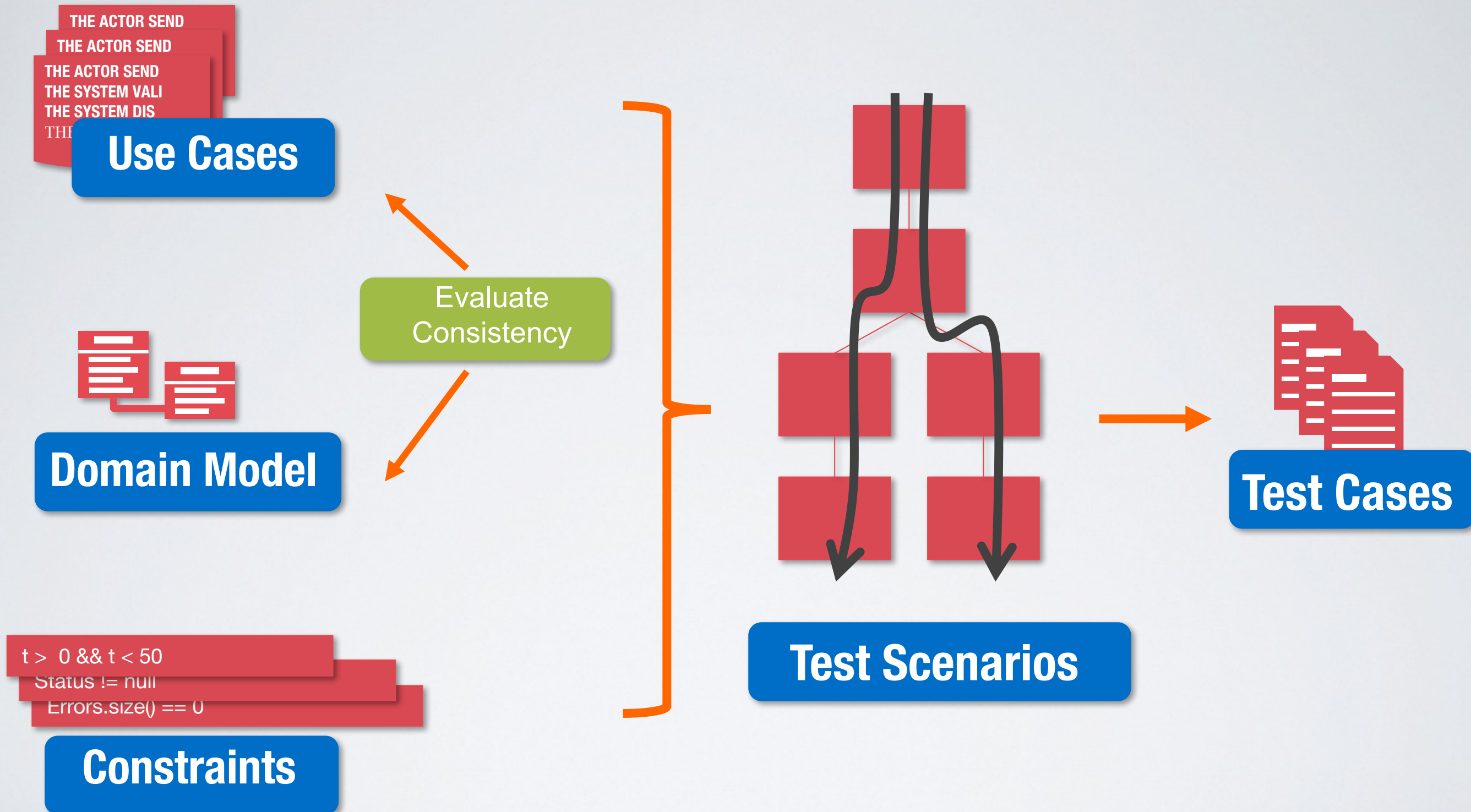
Objectives

- 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

RUCM

[Yue et al. TOSEM'13]

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

2

Model the Domain

3

Evaluate Consistency

5

Specify Constraints

4

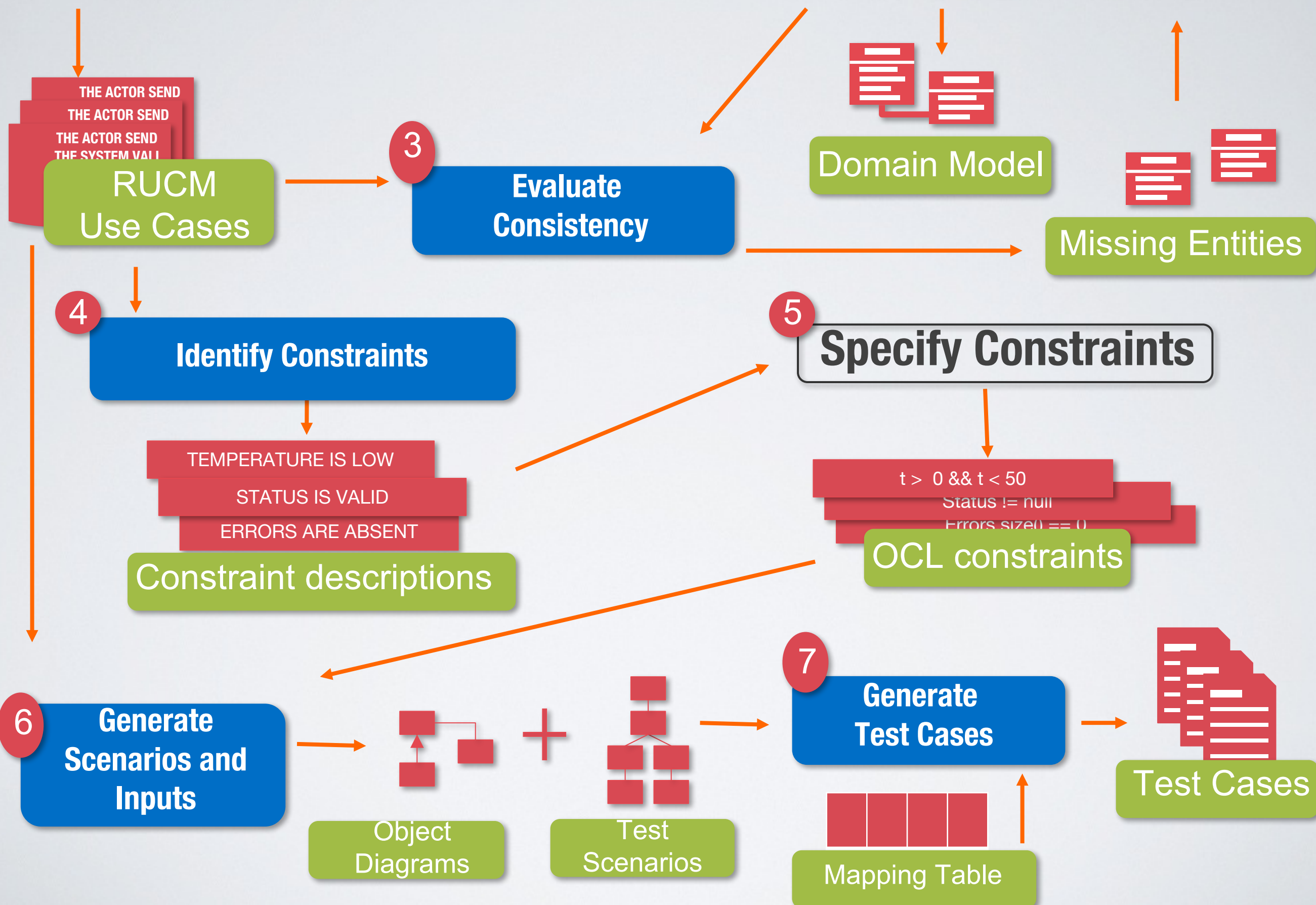
Identify Constraints

7

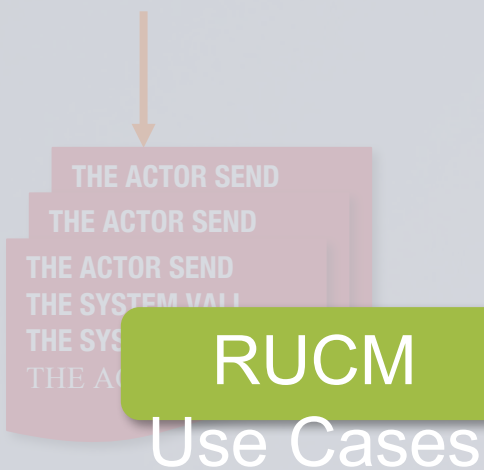
Generate Test Cases

6

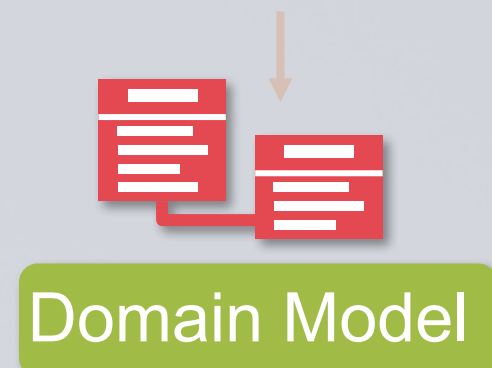
Generate Scenarios and Inputs



1 Elicit Use Cases

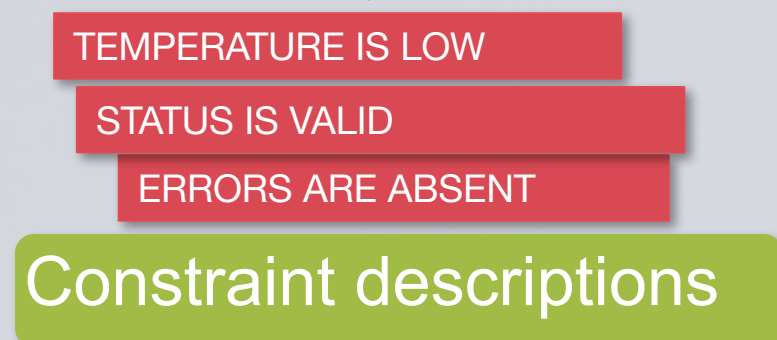


2 Model the Domain

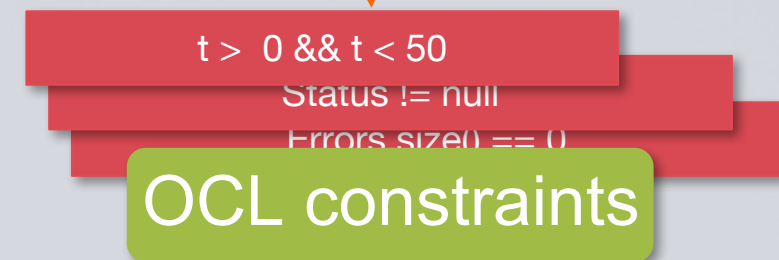


3 Evaluate Consistency

4 Identify Constraints



5 Specify 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

“no error has been detected”

OCCL Constraint

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

“the occupant class for airbag control was derived.”

OCCL Constraint

```
BodySenseSystem.allInstances() → forAll( b |  
  b.OccupantClassForAirbag = Child  
  OR b.OccupantClassForAirbag = Adult )
```

UseCaseStart

Input

Include

Condition

Condition

Output

Exit

DomainEntity

OccupancyStatus

OCCL Constraint

...

Condition

Exit

Output

Exit

DomainEntity

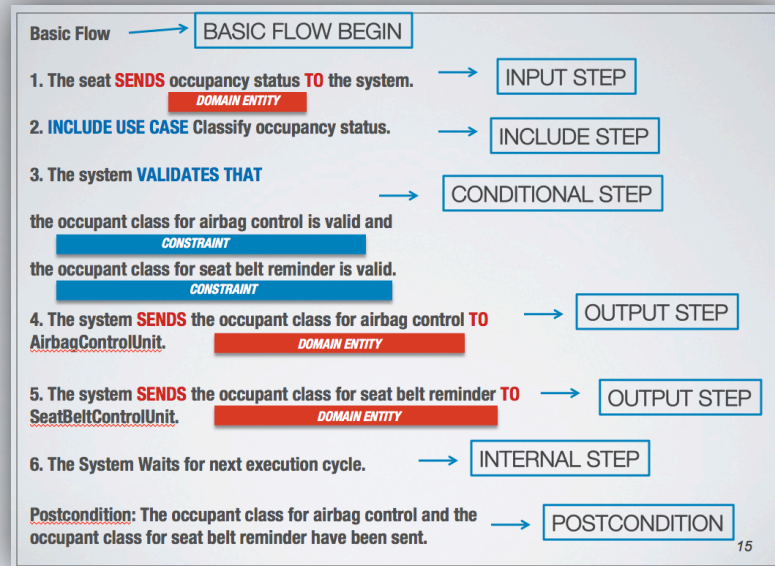
...

DomainEntity

...

Constraint solving: UML2Alloy

Evaluate Model Consistency



Airbag Control ✓

Classification Filter ?

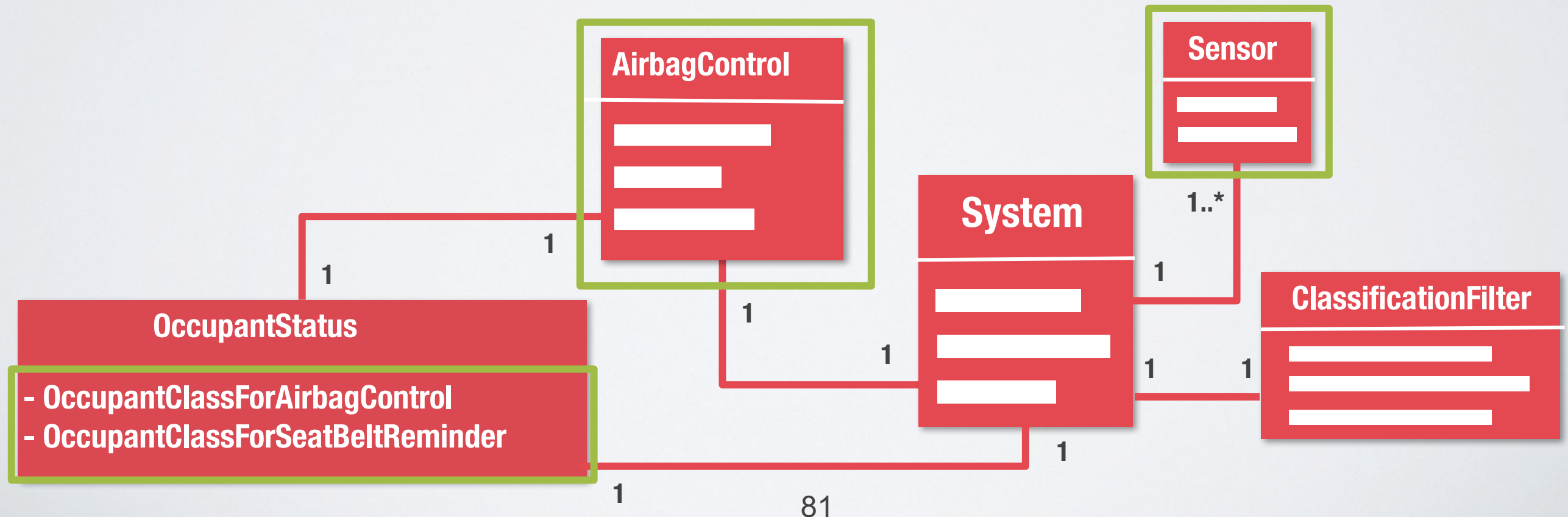
Sensor ✓

Occupant Class for Airbag Control ✓

Occupant Class for Seat Belt Reminder ✓

Tagged Use Case

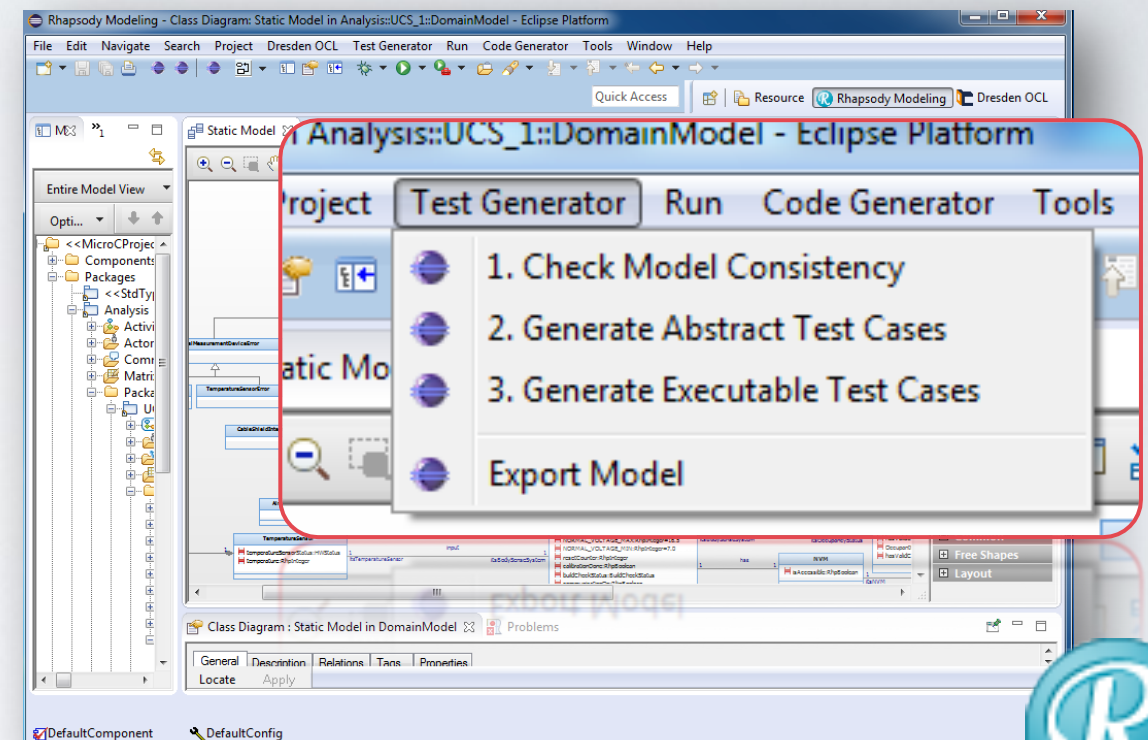
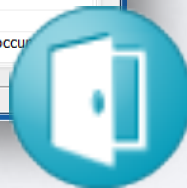
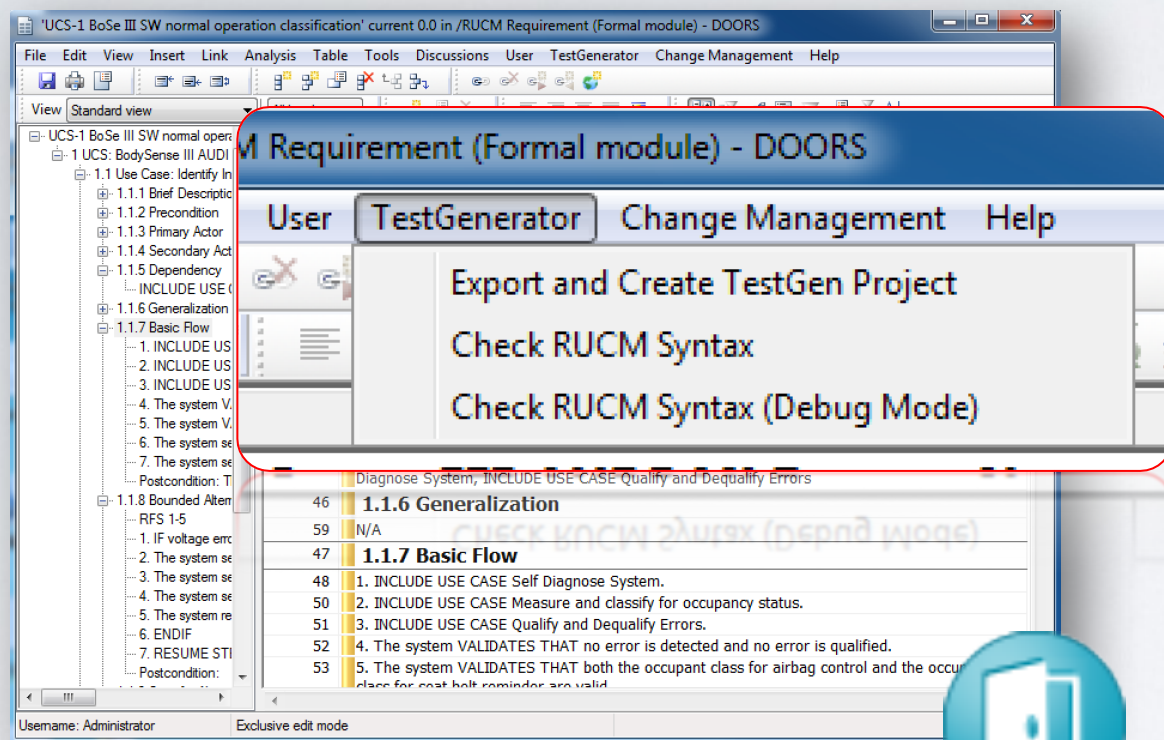
Domain Entities





TG

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**

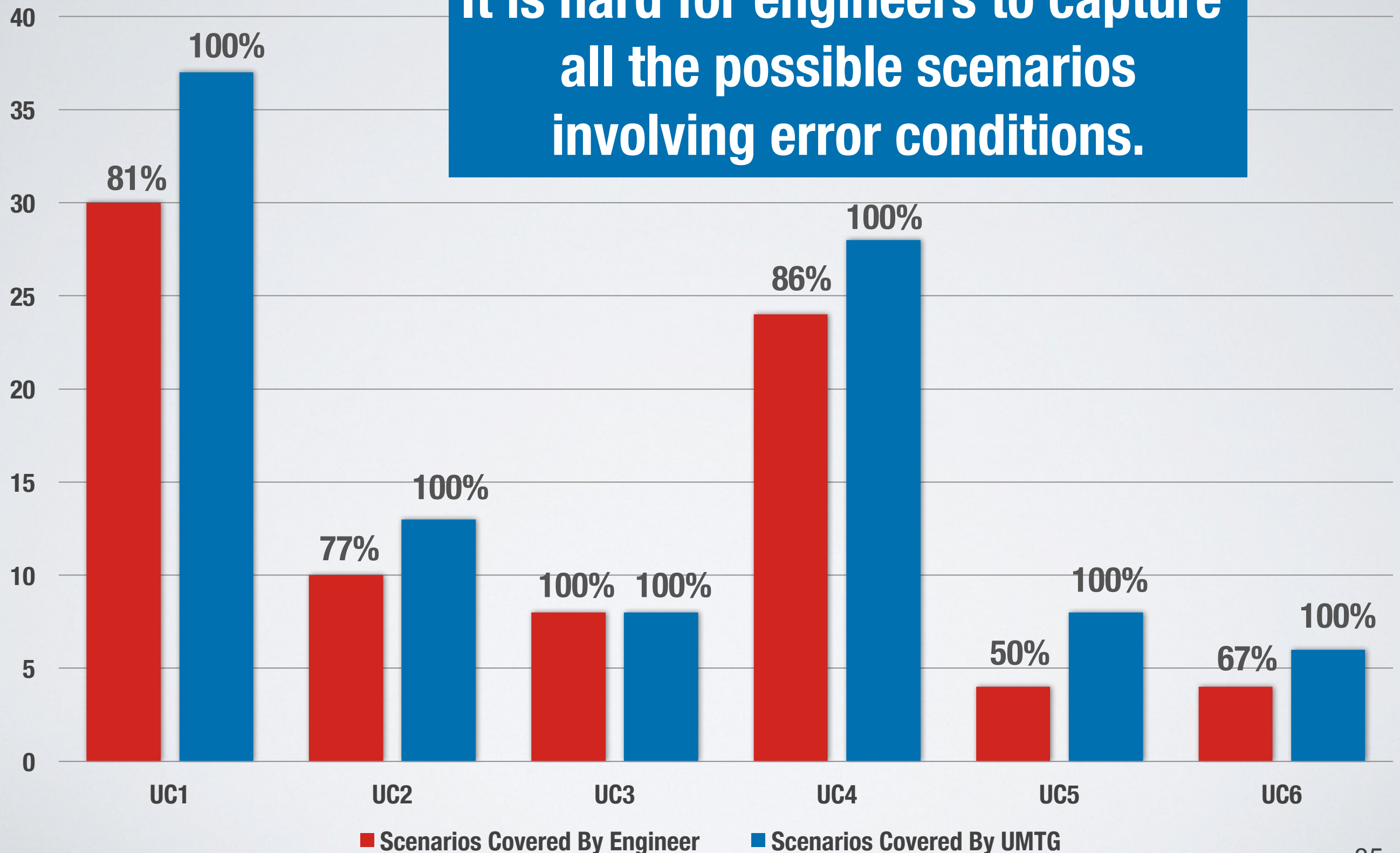
Costs of Additional Modeling

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

It is hard for engineers to capture all the possible scenarios involving error conditions.



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
5  P 1. MISUSE CASE Bypass Authorization Schema
6  Description The MALICIOUS user accesses resources that are dedicated to
7  a user with a different role.
8
9  B 1. PREcondition For each role available on the system, the MALICIOUS user
10 has a list of different of users with that role, plus a list function/resources
11 that can be accessed with that role.
12
13 B 2. Basic Test Flow
14 1. FORACH role
15 2. The MALICIOUS user sends username and password to the system through
16 the login page
17 3. FORACH resource
18 4. The MALICIOUS user requests the resource from the system.
19 5. The system sends a response page to the MALICIOUS user.
20 6. The MALICIOUS user EXPLOITS the system using the response page and
21 the roles.
22
23 E 1. END FORACH
24 12. END FORACH
25
26 P 2. Postcondition: The MALICIOUS user has executed a function dedicated to
27 another user with different role.
28
29 B 3. Specific Alternative Test Flow (SAT1)
30 1. IF role
31 16. I IF user
32 17. I IF role
33 18. I IF role
34 19. I IF role
35 20. I IF role
36 21. I IF role
37 22. I IF role
38 23. I IF role
39 24. I IF role
40 25. I IF role
41 26. I IF role
42 27. I IF role
43 28. I IF role
44 29. I IF role
45 30. I IF role
46 31. I IF role
47 32. I IF role
48 33. I IF role
49 34. I IF role
50 35. I IF role
51 36. I IF role
52 37. I IF role
53 38. I IF role
54 39. I IF role
55 40. I IF role
56 41. I IF role
57 42. I IF role
58 43. I IF role
59 44. I IF role
60 45. I IF role
61 46. I IF role
62 47. I IF role
63 48. I IF role
64 49. I IF role
65 50. I IF role
66 51. I IF role
67 52. I IF role
68 53. I IF role
69 54. I IF role
70 55. I IF role
71 56. I IF role
72 57. I IF role
73 58. I IF role
74 59. I IF role
75 60. I IF role
76 61. I IF role
77 62. I IF role
78 63. I IF role
79 64. I IF role
80 65. I IF role
81 66. I IF role
82 67. I IF role
83 68. I IF role
84 69. I IF role
85 70. I IF role
86 71. I IF role
87 72. I IF role
88 73. I IF role
89 74. I IF role
90 75. I IF role
91 76. I IF role
92 77. I IF role
93 78. I IF role
94 79. I IF role
95 80. I IF role
96 81. I IF role
97 82. I IF role
98 83. I IF role
99 84. I IF role
100 85. I IF role
101 86. I IF role
102 87. I IF role
103 88. I IF role
104 89. I IF role
105 90. I IF role
106 91. I IF role
107 92. I IF role
108 93. I IF role
109 94. I IF role
110 95. I IF role
111 96. I IF role
112 97. I IF role
113 98. I IF role
114 99. I IF role
115 100. I IF role
116 101. I IF role
117 102. I IF role
118 103. I IF role
119 104. I IF role
120 105. I IF role
121 106. I IF role
122 107. I IF role
123 108. I IF role
124 109. I IF role
125 110. I IF role
126 111. I IF role
127 112. I IF role
128 113. I IF role
129 114. I IF role
130 115. I IF role
131 116. I IF role
132 117. I IF role
133 118. I IF role
134 119. I IF role
135 120. I IF role
136 121. I IF role
137 122. I IF role
138 123. I IF role
139 124. I IF role
140 125. I IF role
141 126. I IF role
142 127. I IF role
143 128. I IF role
144 129. I IF role
145 130. I IF role
146 131. I IF role
147 132. I IF role
148 133. I IF role
149 134. I IF role
150 135. I IF role
151 136. I IF role
152 137. I IF role
153 138. I IF role
154 139. I IF role
155 140. I IF role
156 141. I IF role
157 142. I IF role
158 143. I IF role
159 144. I IF role
160 145. I IF role
161 146. I IF role
162 147. I IF role
163 148. I IF role
164 149. I IF role
165 150. I IF role
166 151. I IF role
167 152. I IF role
168 153. I IF role
169 154. I IF role
170 155. I IF role
171 156. I IF role
172 157. I IF role
173 158. I IF role
174 159. I IF role
175 160. I IF role
176 161. I IF role
177 162. I IF role
178 163. I IF role
179 164. I IF role
180 165. I IF role
181 166. I IF role
182 167. I IF role
183 168. I IF role
184 169. I IF role
185 170. I IF role
186 171. I IF role
187 172. I IF role
188 173. I IF role
189 174. I IF role
190 175. I IF role
191 176. I IF role
192 177. I IF role
193 178. I IF role
194 179. I IF role
195 180. I IF role
196 181. I IF role
197 182. I IF role
198 183. I IF role
199 184. I IF role
200 185. I IF role
201 186. I IF role
202 187. I IF role
203 188. I IF role
204 189. I IF role
205 190. I IF role
206 191. I IF role
207 192. I IF role
208 193. I IF role
209 194. I IF role
210 195. I IF role
211 196. I IF role
212 197. I IF role
213 198. I IF role
214 199. I IF role
215 200. I IF role
216 201. I IF role
217 202. I IF role
218 203. I IF role
219 204. I IF role
220 205. I IF role
221 206. I IF role
222 207. I IF role
223 208. I IF role
224 209. I IF role
225 210. I IF role
226 211. I IF role
227 212. I IF role
228 213. I IF role
229 214. I IF role
230 215. I IF role
231 216. I IF role
232 217. I IF role
233 218. I IF role
234 219. I IF role
235 220. I IF role
236 221. I IF role
237 222. I IF role
238 223. I IF role
239 224. I IF role
240 225. I IF role
241 226. I IF role
242 227. I IF role
243 228. I IF role
244 229. I IF role
245 230. I IF role
246 231. I IF role
247 232. I IF role
248 233. I IF role
249 234. I IF role
250 235. I IF role
251 236. I IF role
252 237. I IF role
253 238. I IF role
254 239. I IF role
255 240. I IF role
256 241. I IF role
257 242. I IF role
258 243. I IF role
259 244. I IF role
260 245. I IF role
261 246. I IF role
262 247. I IF role
263 248. I IF role
264 249. I IF role
265 250. I IF role
266 251. I IF role
267 252. I IF role
268 253. I IF role
269 254. I IF role
270 255. I IF role
271 256. I IF role
272 257. I IF role
273 258. I IF role
274 259. I IF role
275 260. I IF role
276 261. I IF role
277 262. I IF role
278 263. I IF role
279 264. I IF role
280 265. I IF role
281 266. I IF role
282 267. I IF role
283 268. I IF role
284 269. I IF role
285 270. I IF role
286 271. I IF role
287 272. I IF role
288 273. I IF role
289 274. I IF role
290 275. I IF role
291 276. I IF role
292 277. I IF role
293 278. I IF role
294 279. I IF role
295 280. I IF role
296 281. I IF role
297 282. I IF role
298 283. I IF role
299 284. I IF role
300 285. I IF role
301 286. I IF role
302 287. I IF role
303 288. I IF role
304 289. I IF role
305 290. I IF role
306 291. I IF role
307 292. I IF role
308 293. I IF role
309 294. I IF role
310 295. I IF role
311 296. I IF role
312 297. I IF role
313 298. I IF role
314 299. I IF role
315 300. I IF role
316 301. I IF role
317 302. I IF role
318 303. I IF role
319 304. I IF role
320 305. I IF role
321 306. I IF role
322 307. I IF role
323 308. I IF role
324 309. I IF role
325 310. I IF role
326 311. I IF role
327 312. I IF role
328 313. I IF role
329 314. I IF role
330 315. I IF role
331 316. I IF role
332 317. I IF role
333 318. I IF role
334 319. I IF role
335 320. I IF role
336 321. I IF role
337 322. I IF role
338 323. I IF role
339 324. I IF role
340 325. I IF role
341 326. I IF role
342 327. I IF role
343 328. I IF role
344 329. I IF role
345 330. I IF role
346 331. I IF role
347 332. I IF role
348 333. I IF role
349 334. I IF role
350 335. I IF role
351 336. I IF role
352 337. I IF role
353 338. I IF role
354 339. I IF role
```

```

SECURITY USE CASE: Anonymous user data
1. The system VALIDATES THAT the user agreed to show his name
Postcondition: the user data can be visualized.
Specific Security Use Case: Anonymous user data
RFS 1. 1. The system VALIDATES THAT the user agreed to show his name
The rfs Postcondition: the user data can be visualized.
EXIT:
Postcon SECURITY 1. The system VALIDATES THAT the user agreed to show his name
SPECIF 1. The rfs Postcondition: the user data can be visualized.
Precon EXIT Specific Alternative Flow
Group: POSTCON RFS 1.
Basic Flow 1. The system replaces the user name with the internal user id.
1. The group EXIT.
2. The group EXIT.
3. The group EXIT.
4. The group EXIT.
5. The group EXIT.
6. The group EXIT.
7. The group EXIT.
8. The group EXIT.
9. The group EXIT.
10. The group EXIT.
11. The group EXIT.
12. The group EXIT.
13. The group EXIT.
14. The group EXIT.
15. The group EXIT.
16. The group EXIT.
17. The group EXIT.
18. The group EXIT.
19. The group EXIT.
20. The group EXIT.
21. The group EXIT.
22. The group EXIT.
23. The group EXIT.
24. The group EXIT.
25. The group EXIT.
26. The group EXIT.
27. The group EXIT.
28. The group EXIT.
29. The group EXIT.
30. The group EXIT.
31. The group EXIT.
32. The group EXIT.
33. The group EXIT.
34. The group EXIT.
35. The group EXIT.
36. The group EXIT.
37. The group EXIT.
38. The group EXIT.
39. The group EXIT.
40. The group EXIT.
41. The group EXIT.
42. The group EXIT.
43. The group EXIT.
44. The group EXIT.
45. The group EXIT.
46. The group EXIT.
47. The group EXIT.
48. The group EXIT.
49. The group EXIT.
50. The group EXIT.
51. The group EXIT.
52. The group EXIT.
53. The group EXIT.
54. The group EXIT.
55. The group EXIT.
56. The group EXIT.
57. The group EXIT.
58. The group EXIT.
59. The group EXIT.
60. The group EXIT.
61. The group EXIT.
62. The group EXIT.
63. The group EXIT.
64. The group EXIT.
65. The group EXIT.
66. The group EXIT.
67. The group EXIT.
68. The group EXIT.
69. The group EXIT.
70. The group EXIT.
71. The group EXIT.
72. The group EXIT.
73. The group EXIT.
74. The group EXIT.
75. The group EXIT.
76. The group EXIT.
77. The group EXIT.
78. The group EXIT.
79. The group EXIT.
80. The group EXIT.
81. The group EXIT.
82. The group EXIT.
83. The group EXIT.
84. The group EXIT.
85. The group EXIT.
86. The group EXIT.
87. The group EXIT.
88. The group EXIT.
89. The group EXIT.
90. The group EXIT.
91. The group EXIT.
92. The group EXIT.
93. The group EXIT.
94. The group EXIT.
95. The group EXIT.
96. The group EXIT.
97. The group EXIT.
98. The group EXIT.
99. The group EXIT.
100. The group EXIT.
101. The group EXIT.
102. The group EXIT.
103. The group EXIT.
104. The group EXIT.
105. The group EXIT.
106. The group EXIT.
107. The group EXIT.
108. The group EXIT.
109. The group EXIT.
110. The group EXIT.
111. The group EXIT.
112. The group EXIT.
113. The group EXIT.
114. The group EXIT.
115. The group EXIT.
116. The group EXIT.
117. The group EXIT.
118. The group EXIT.
119. The group EXIT.
120. The group EXIT.
121. The group EXIT.
122. The group EXIT.
123. The group EXIT.
124. The group EXIT.
125. The group EXIT.
126. The group EXIT.
127. The group EXIT.
128. The group EXIT.
129. The group EXIT.
130. The group EXIT.
131. The group EXIT.
132. The group EXIT.
133. The group EXIT.
134. The group EXIT.
135. The group EXIT.
136. The group EXIT.
137. The group EXIT.
138. The group EXIT.
139. The group EXIT.
140. The group EXIT.
141. The group EXIT.
142. The group EXIT.
143. The group EXIT.
144. The group EXIT.
145. The group EXIT.
146. The group EXIT.
147. The group EXIT.
148. The group EXIT.
149. The group EXIT.
150. The group EXIT.
151. The group EXIT.
152. The group EXIT.
153. The group EXIT.
154. The group EXIT.
155. The group EXIT.
156. The group EXIT.
157. The group EXIT.
158. The group EXIT.
159. The group EXIT.
160. The group EXIT.
161. The group EXIT.
162. The group EXIT.
163. The group EXIT.
164. The group EXIT.
165. The group EXIT.
166. The group EXIT.
167. The group EXIT.
168. The group EXIT.
169. The group EXIT.
170. The group EXIT.
171. The group EXIT.
172. The group EXIT.
173. The group EXIT.
174. The group EXIT.
175. The group EXIT.
176. The group EXIT.
177. The group EXIT.
178. The group EXIT.
179. The group EXIT.
180. The group EXIT.
181. The group EXIT.
182. The group EXIT.
183. The group EXIT.
184. The group EXIT.
185. The group EXIT.
186. The group EXIT.
187. The group EXIT.
188. The group EXIT.
189. The group EXIT.
190. The group EXIT.
191. The group EXIT.
192. The group EXIT.
193. The group EXIT.
194. The group EXIT.
195. The group EXIT.
196. The group EXIT.
197. The group EXIT.
198. The group EXIT.
199. The group EXIT.
200. The group EXIT.
201. The group EXIT.
202. The group EXIT.
203. The group EXIT.
204. The group EXIT.
205. The group EXIT.
206. The group EXIT.
207. The group EXIT.
208. The group EXIT.
209. The group EXIT.
210. The group EXIT.
211. The group EXIT.
212. The group EXIT.
213. The group EXIT.
214. The group EXIT.
215. The group EXIT.
216. The group EXIT.
217. The group EXIT.
218. The group EXIT.
219. The group EXIT.
220. The group EXIT.
221. The group EXIT.
222. The group EXIT.
223. The group EXIT.
224. The group EXIT.
225. The group EXIT.
226. The group EXIT.
227. The group EXIT.
228. The group EXIT.
229. The group EXIT.
230. The group EXIT.
231. The group EXIT.
232. The group EXIT.
233. The group EXIT.
234. The group EXIT.
235. The group EXIT.
236. The group EXIT.
237. The group EXIT.
238. The group EXIT.
239. The group EXIT.
240. The group EXIT.
241. The group EXIT.
242. The group EXIT.
243. The group EXIT.
244. The group EXIT.
245. The group EXIT.
246. The group EXIT.
247. The group EXIT.
248. The group EXIT.
249. The group EXIT.
250. The group EXIT.
251. The group EXIT.
252. The group EXIT.
253. The group EXIT.
254. The group EXIT.
255. The group EXIT.
256. The group EXIT.
257. The group EXIT.
258. The group EXIT.
259. The group EXIT.
260. The group EXIT.
261. The group EXIT.
262. The group EXIT.
263. The group EXIT.
264. The group EXIT.
265. The group EXIT.
266. The group EXIT.
267. The group EXIT.
268. The group EXIT
```

Specifications

Specifications

Automated Generation of Executable Test Cases

Security Vulnerability Testing

Addressing the identification of system vulnerabilities

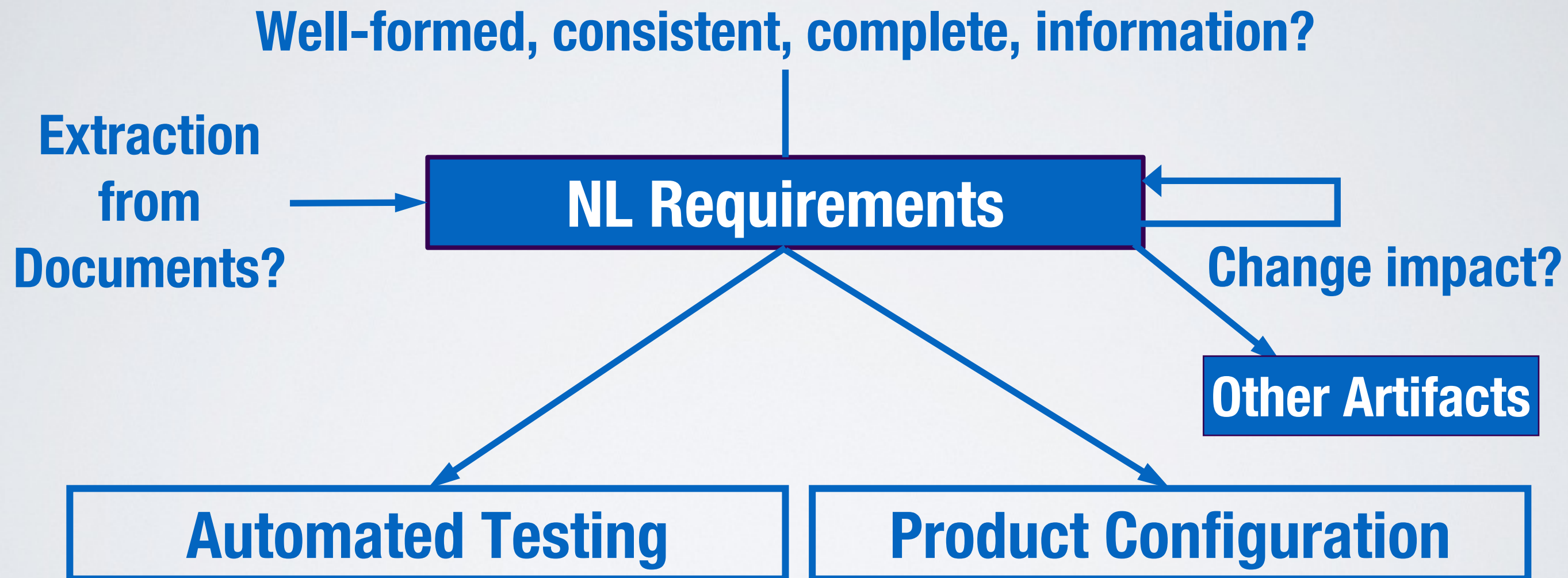
Security Functional Testing

Validating whether the specified security properties are implemented correctly

Benefits of automated generation:

- **Automated generation reduces development costs**
- **Ensures coverage and traceability**
- **Compliance with standards and regulations**

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