2018 Annual INCOSE
Great Lakes Regional Conference
SYSTEMS AT THE CROSSROADS
17 - 20 October 2018 | Indianapolis, Indiana

Using Cognitive Computing to Elevate Requirements Quality and Test Case Accuracy

Bob Parro, VP Business Development & Product Development, River North Solutions
Steve Denman, Principal Consultant, Stephen D. Denman Consulting, LLC
Kevin McHugh, Senior Managing Consultant, IBM Watson IoT Lab Services

www.incose.org/glrc2018

Copyright © 2018 by Bob Parro, Steve Denman, Kevin McHugh.
Permission granted to INCOSE to publish and use.

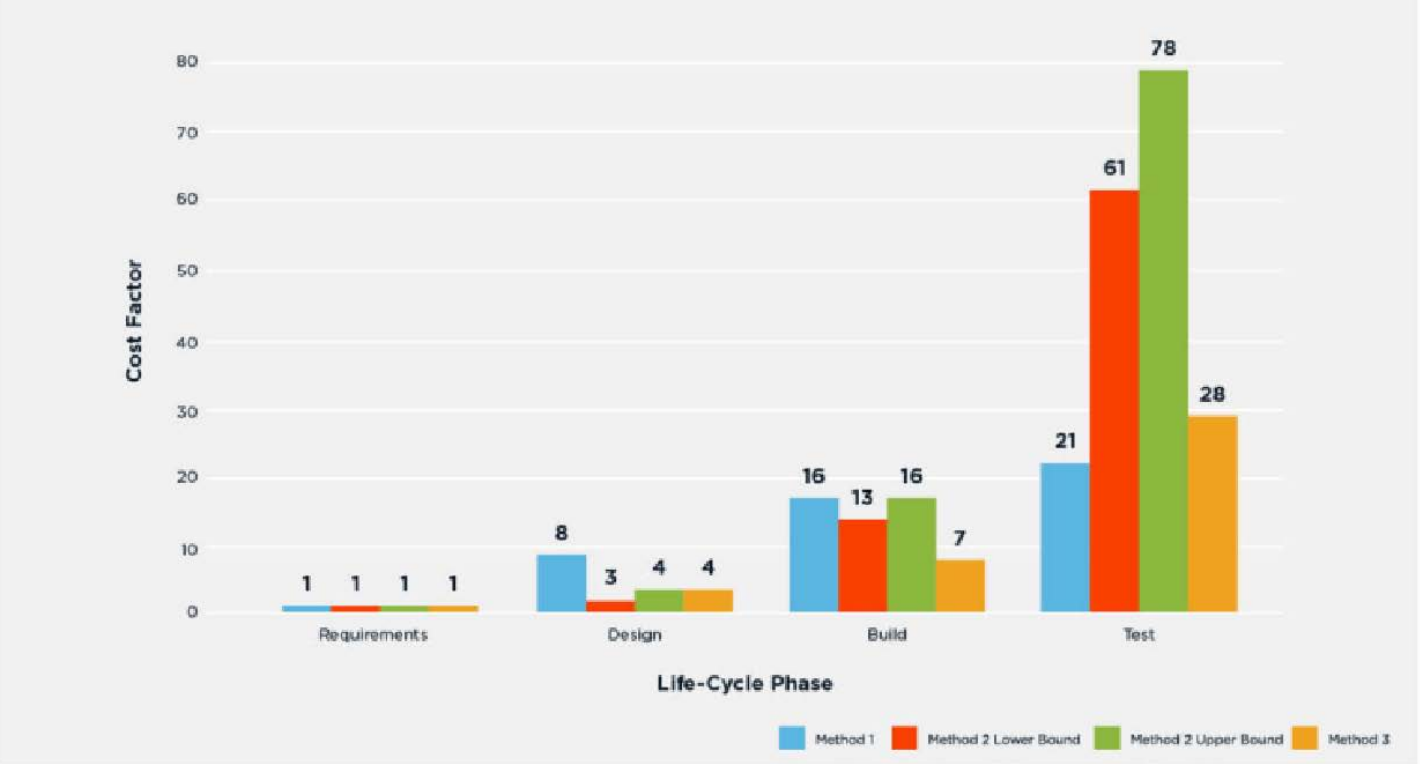
Agenda

- The problem
- Cognitive approaches
- Cognitive engineering
- Cognitive requirements and test offerings
- Benefits of cognitive tools
- Future of cognitive engineering
- Summary & challenge
- Q&A

The problem

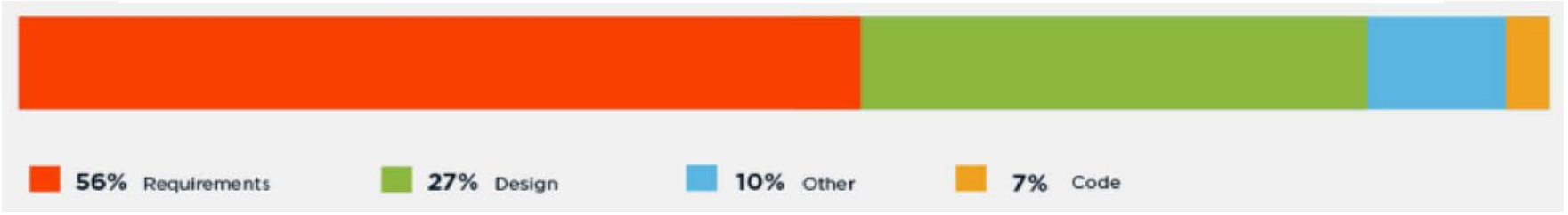
- Poor requirements are widely understood to be the cause of numerous project/product failures
  - Schedule delays, cost overruns, poor quality, going to market with reduced functionality compared to the original plan, going to market with the wrong product or solution
  - These impacts have been studied and long understood

The problem



Comparison of System Cost Factors – Excluding Operations [Jonette]

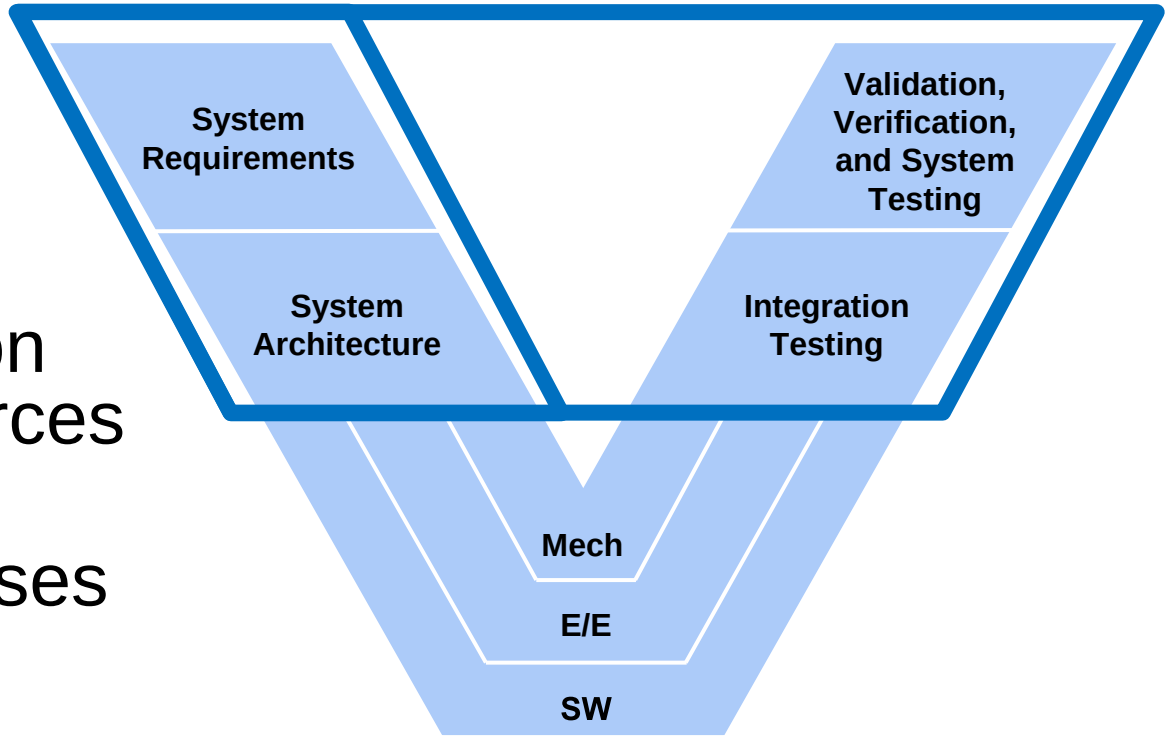
Cost to Fix Errors Increases Exponentially the Later They are Found in the Development Lifecycle



Distribution of defects in software projects by development phase (Martin)

The problem

- But, there's more to this picture
  - Poor validation & verification practices also lead to failures
  - Test development and execution typically consume lots of resources and schedule
    - Only amplified when test cases are written poorly
  - Poor requirements result in poor test cases
    - Compounded by poor test case development practices



The problem underlying the problem

- Human language, aka natural language (NL), ***still*** the primary engineering specification language
  - Ambiguous by nature, even for humans
  - Computer comprehension still far behind human
- 60+ years developing computer comprehension of NL
  - Procedural
  - Formal specification
  - Cognitive computing
  - Combinations of above

Procedural approaches

- Conventional programming and information models to process natural language
  - Text parsed based on rules-based and/or pattern recognition techniques
    - Rules/patterns based on linguistics knowledge
  - Graph of text parts and relationships created and used to “understand” current and new text
  - Works better when requirements written within constraints
    - The <entity> shall <action verb> <object> <per some measure>

Formal specification

- Avoid issues with natural language by not using it
  - Formal languages
    - Structured, rigorous textual expressions
    - Range from sets of rules/constraints to highly formal expressions
    - Difficult to write - even more difficult to read
  - Modeling (graphical languages)
    - Graphical expression of structure, relationships, behavior, constraints, etc., e.g., SysML
    - Valuable but limited ability to express richness available in NL
    - Use of modeling + NL specifications expanding rapidly

Cognitive computing & deep learning

- Cognitive computing
  - Technology apps/services/platforms based on Artificial Intelligence (AI) techniques, including deep learning
- Deep learning
  - A non-procedural computing approach based on neural network model of human brain
  - Training needed to “teach” the network to recognize patterns
    - Supervised learning stage: Guided by SMEs
    - Requires input corpus of known “good” and “bad” patterns
    - Semi-supervised and unsupervised learning

Natural Language Understanding and Processing (NLU/NLP)

- NLU
  - Machine comprehension of parts of speech, interrelationships, and meaning in a body of text
  - Based on deep learning over a corpus of knowledge
- NLP: Apps and services that utilize NLU
  - Ex: Digital assistants: Apple Siri, Amazon Alexa
  - Ex: Sentiment analysis: IBM Watson, MS Azure
  - Language translation: Google Translate

Cognitive engineering

- Cognitive computing in engineering tool chains
- Tools can *finally* begin to *assist* engineers
  - Improving quality
  - Reducing errors, oversights, time, cost, ...
  - Requirements are getting most of focus currently
  - Testing is next logical choice
    - Also based heavily on NL
    - Derived directly from requirements
- Cross-domain, cross-lifecycle traceability next

Cognitive engineering

Improving quality of requirements & test

- Requirements
  - Numerically score quality – Rationale behind it
  - Recommendations – Rationale behind them
- Test
  - Numerically score quality – Rationale behind it
  - Recommendations – Rationale behind them

Cognitive engineering
Cognitive requirements and test offerings

A sample of the available offerings:

- QRA Corp QVscribe
- Argosim STIMULUS
- Qualicen Scout
- The Reuse Company offerings

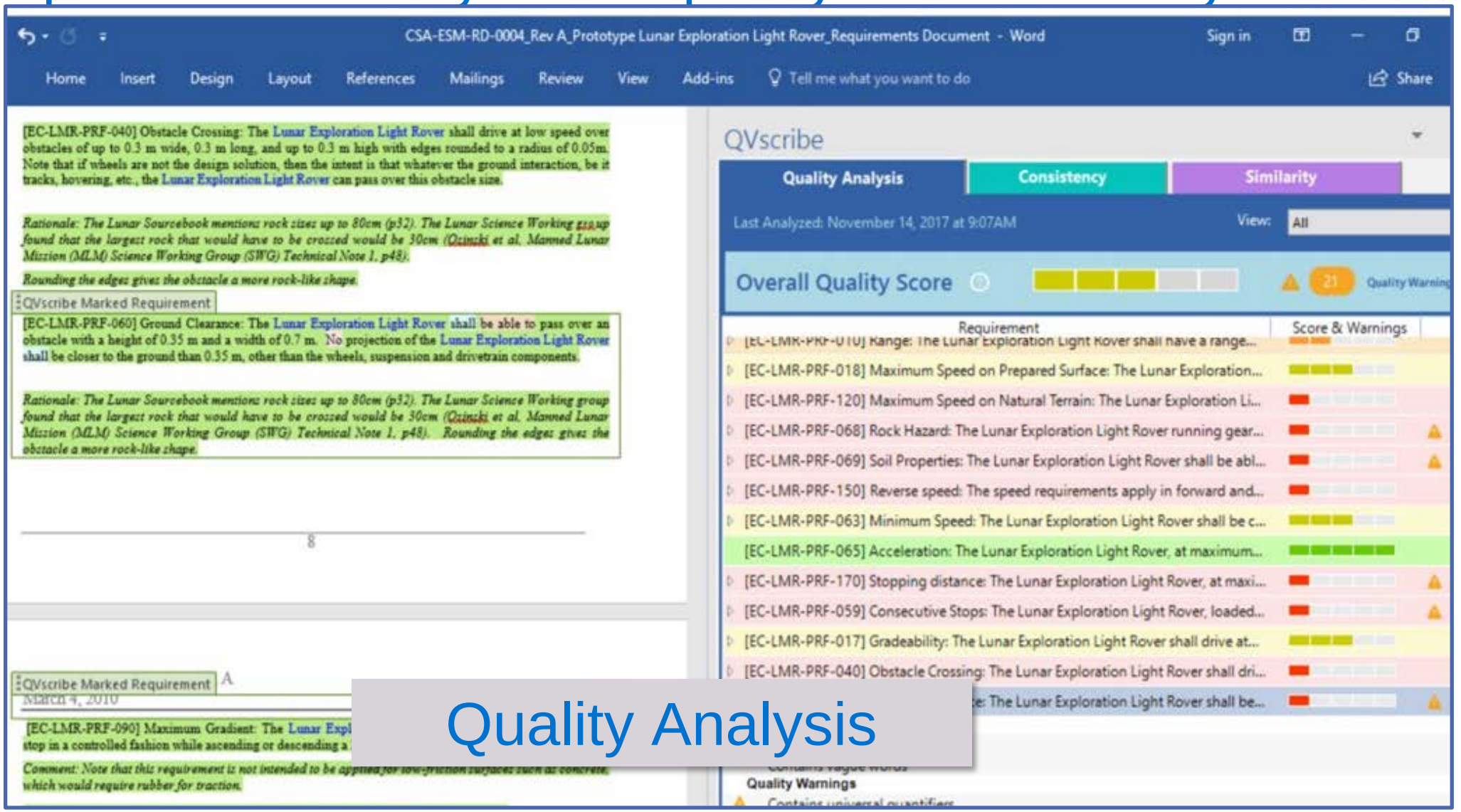
QVscribe from QRA Corp
Requirements analysis for quality & consistency

Uses Natural Language Processing (NLP)

- Auto-detects requirements
- IDs and corrects ambiguities
- IDs duplicate or conflicting requirements
- Checks terminology and units consistency
- Quality Indicators
  - Imperatives (e.g. shall), Options (e.g. can, may), Vagueness (e.g. pronouns), Weaknesses (e.g. adequate), Subjectiveness (e.g. larger), Continuances (e.g. below), Universal quantifiers (all)
- Produces quality scorecard

QVscribe from QRA Corp

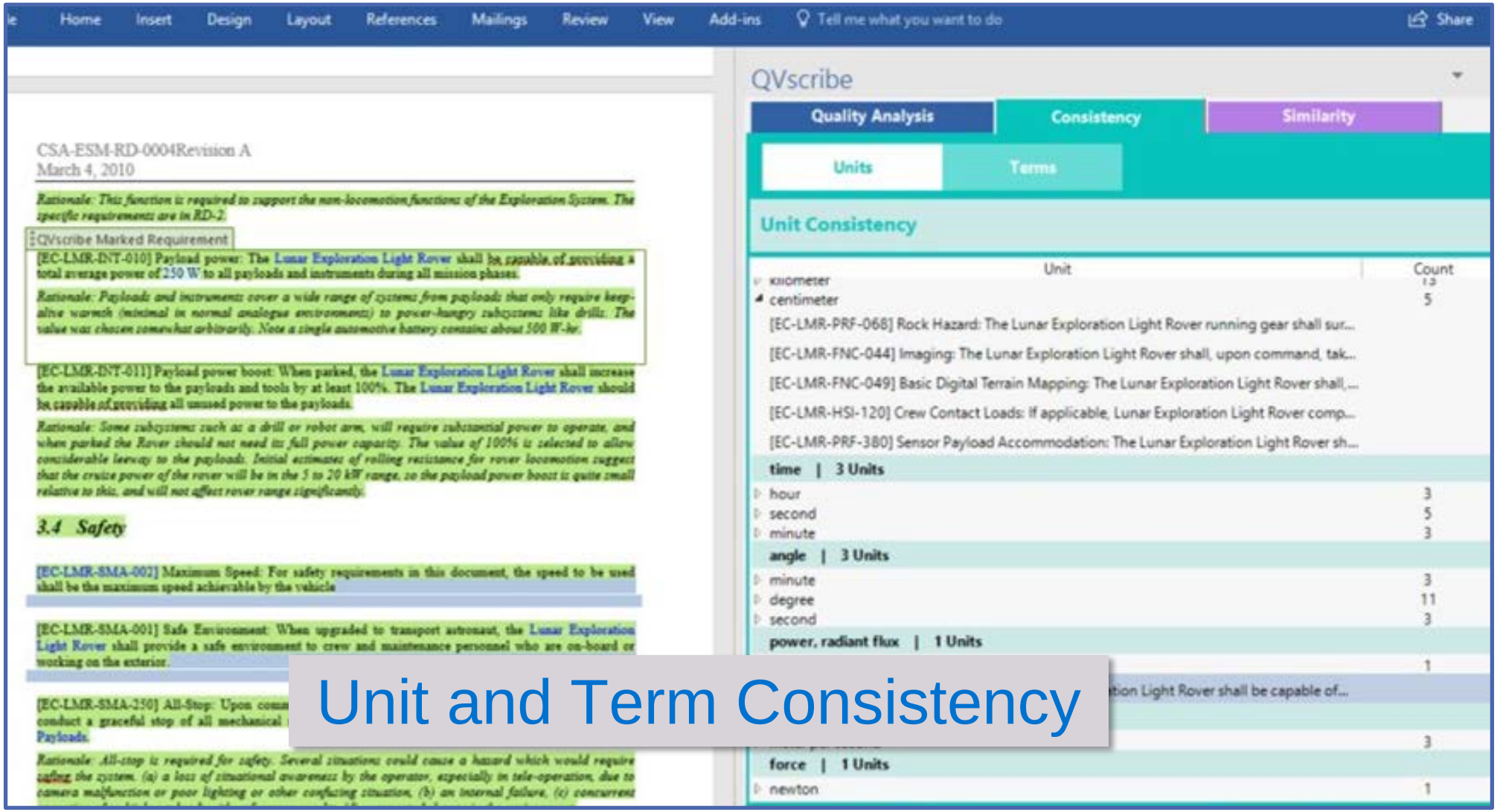
Requirements analysis for quality & consistency



Quality Analysis

QVscribe from QRA Corp

Requirements analysis for quality & consistency

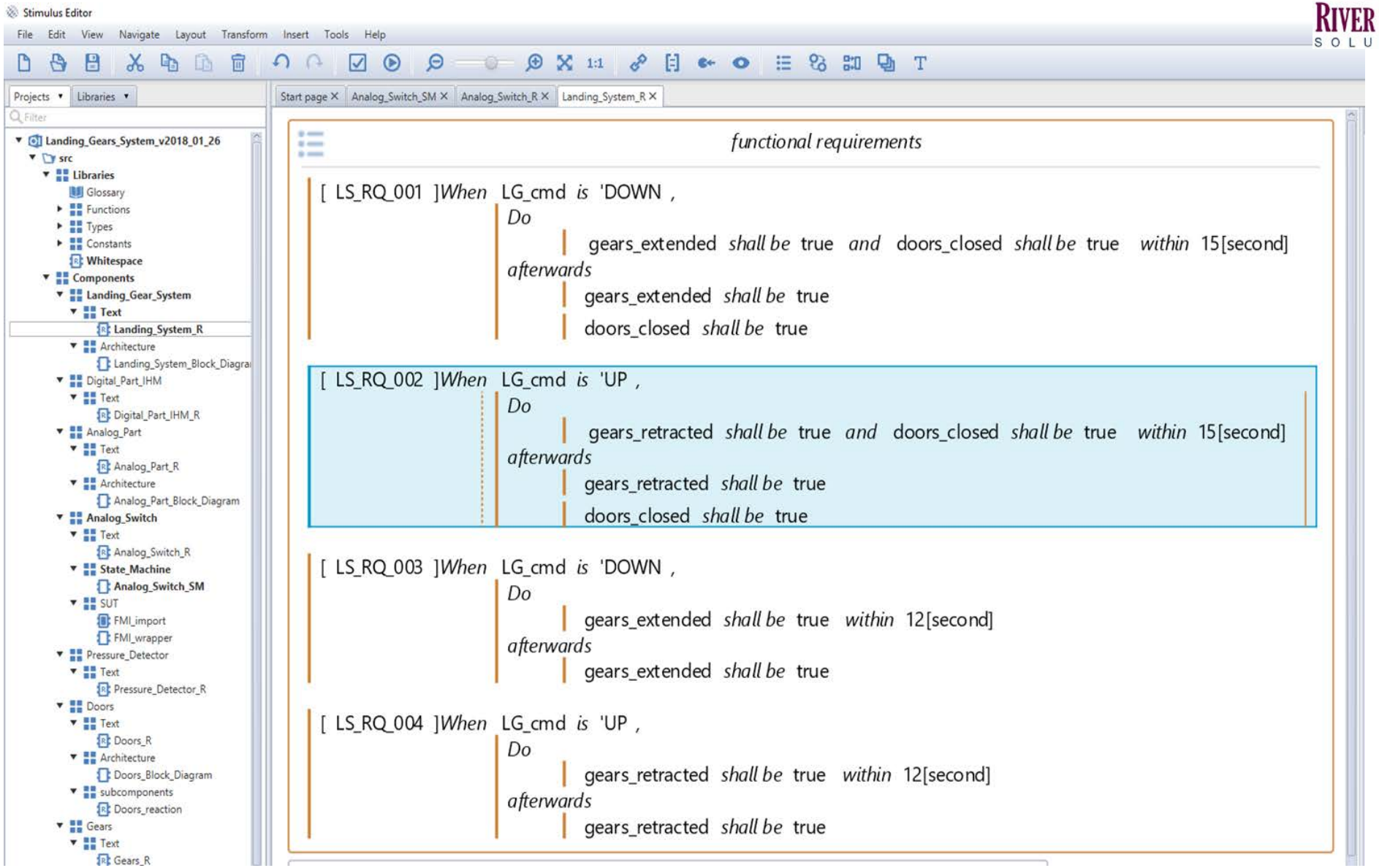


Unit and Term Consistency

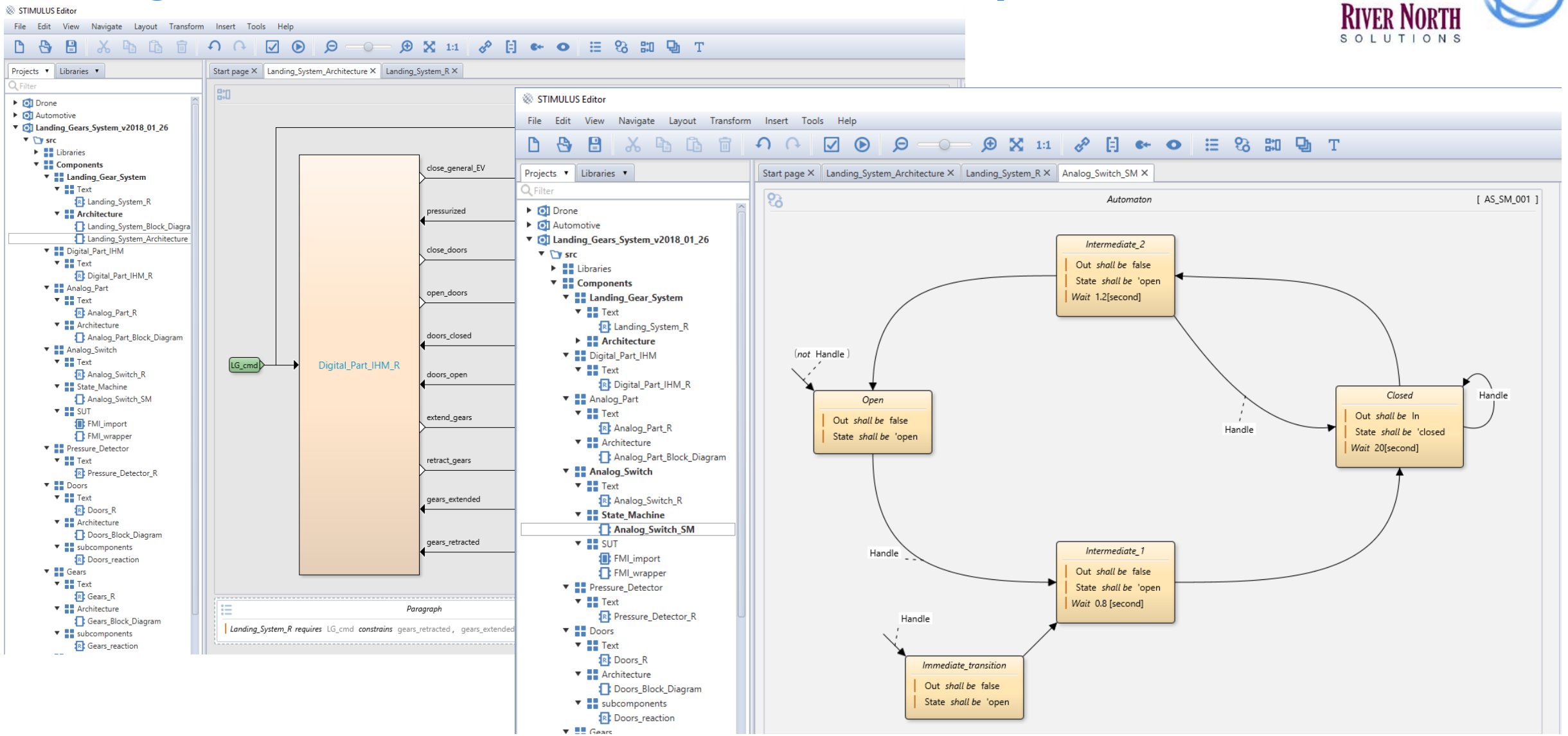
Argosim STIMULUS

- *Requirements-in-the-loop*
  - Executable requirements and system specs
- Automated requirements validation and verification
  - Based on formal requirements specification language and system models
  - Conflicting/missing requirements and functional coverage
- Formal language and system models automatically converted to software-in-the-loop (SIL) validation tests
  - Based on conversion to executable SIL test code via the Functional Mockup Interface (FMI) standard

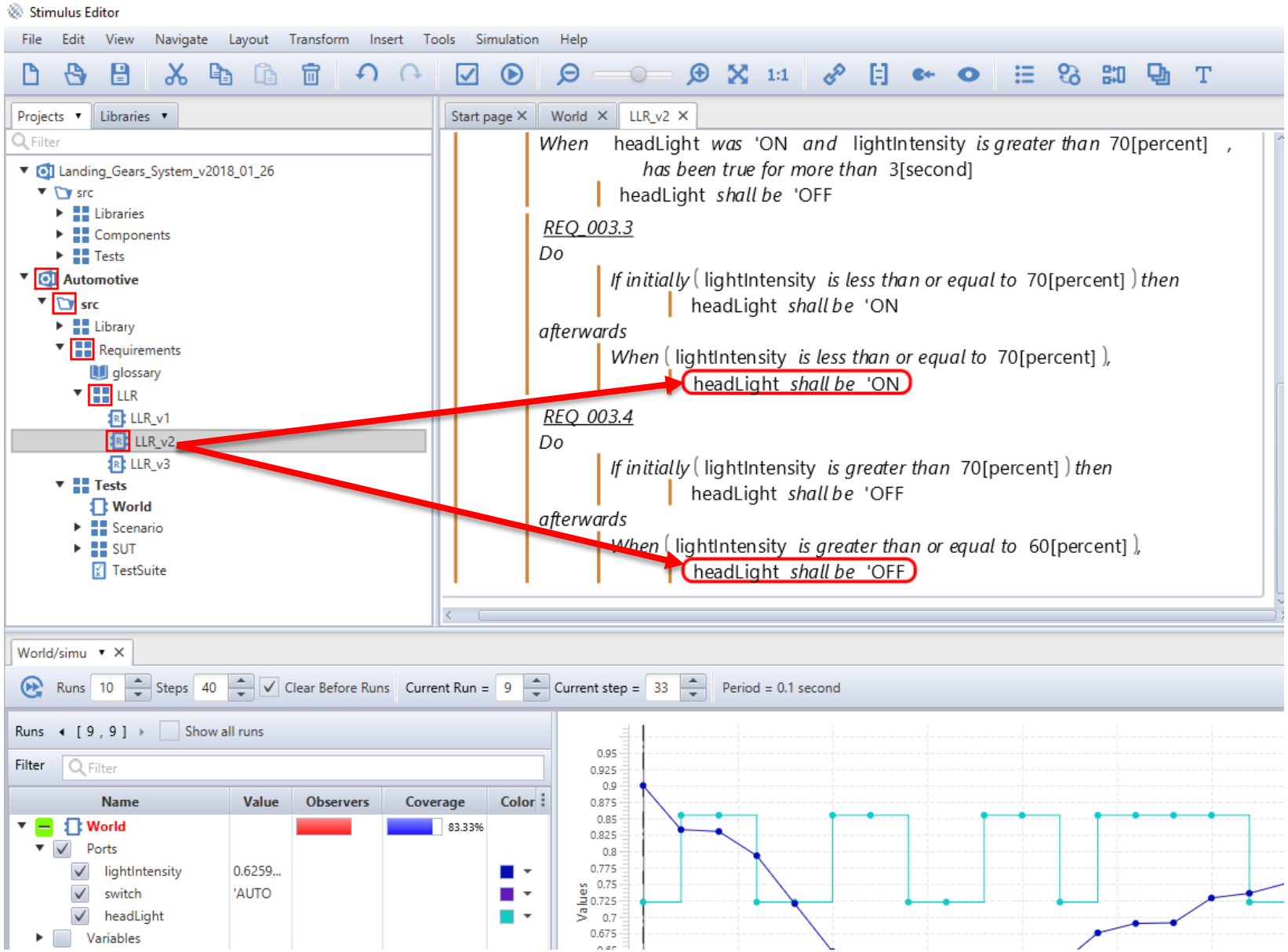
Argosim STIMULUS for Requirements



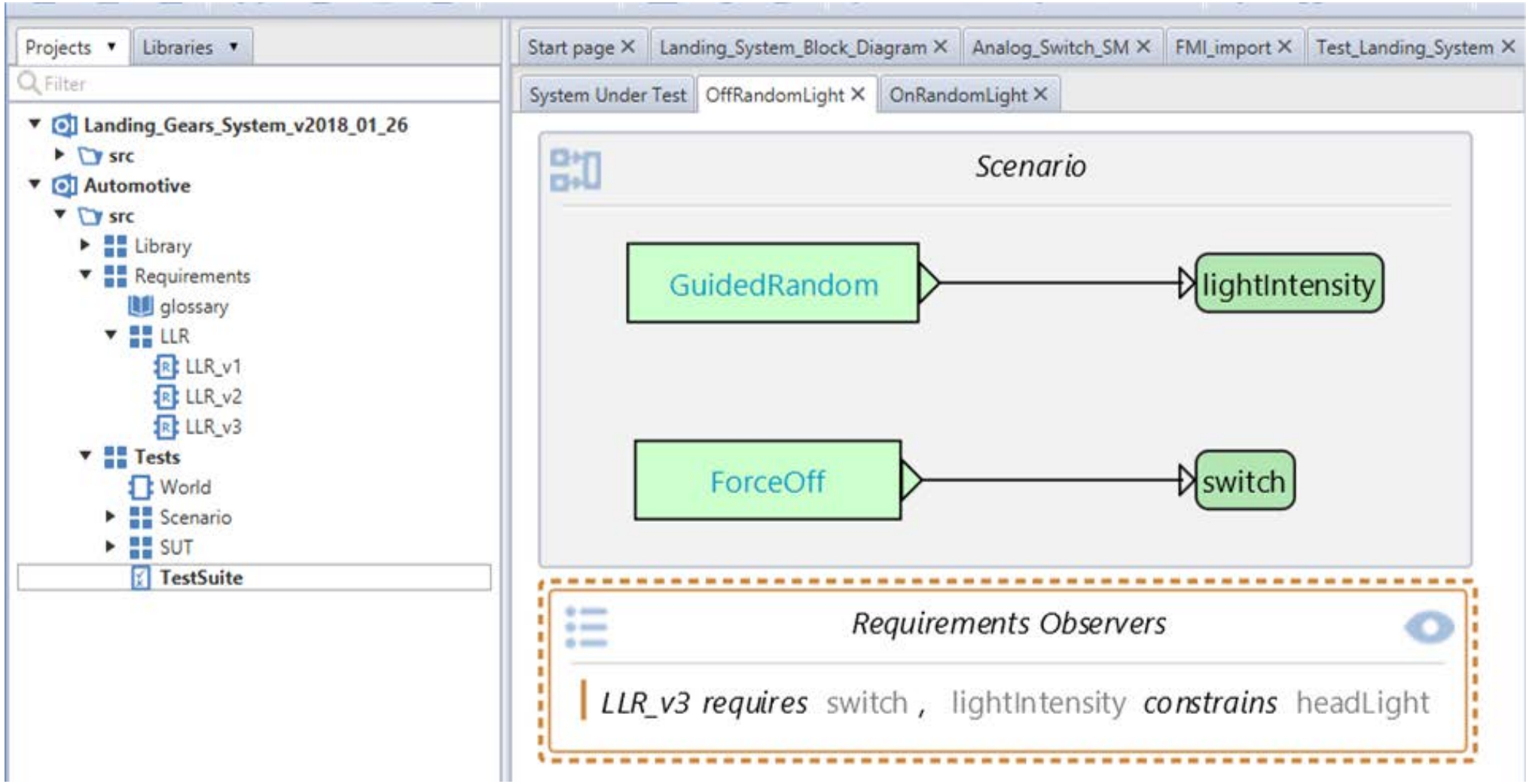
Argosim STIMULUS for Requirements



Argosim STIMULUS for Requirements



Argosim STIMULUS for Tests



Scout from Qualicen

Rapid user feedback provides strongest learning

Integrations

- Word, DOORS, PTC Integrity

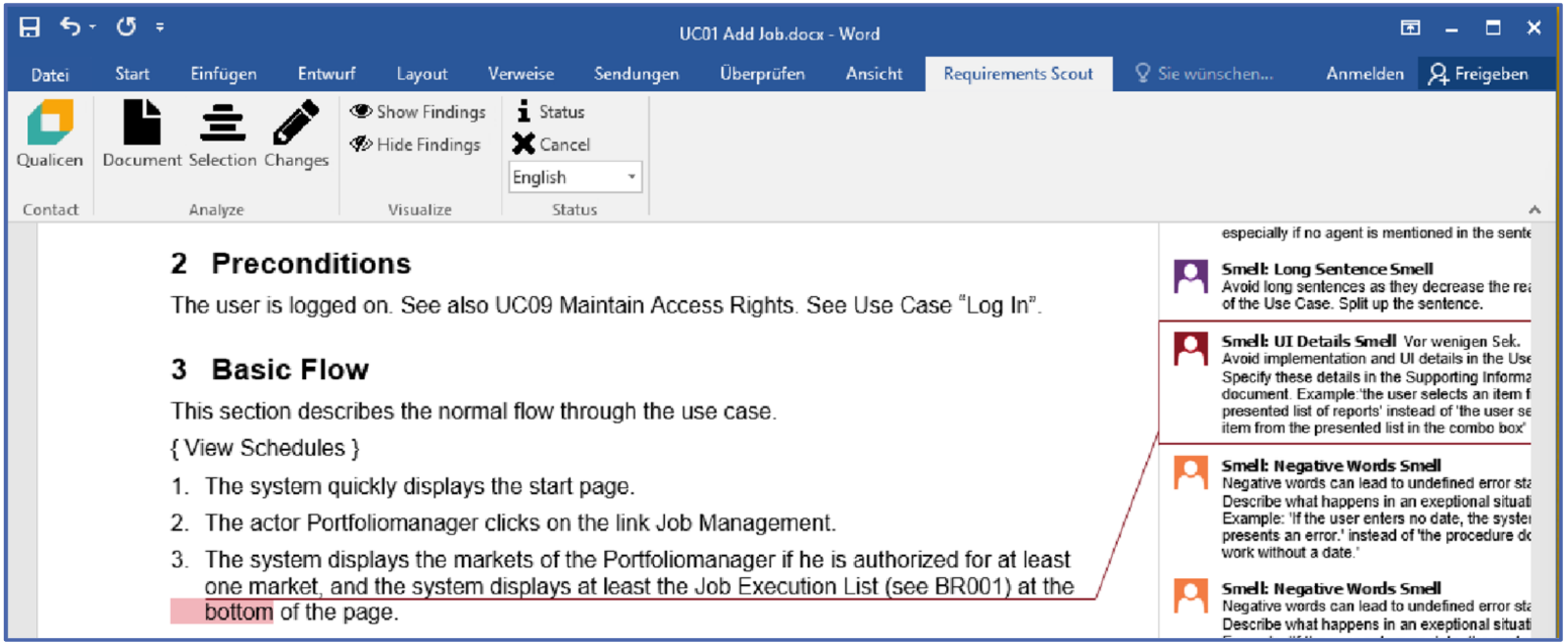
Trend analysis key capability

- Users want quality improvement over time

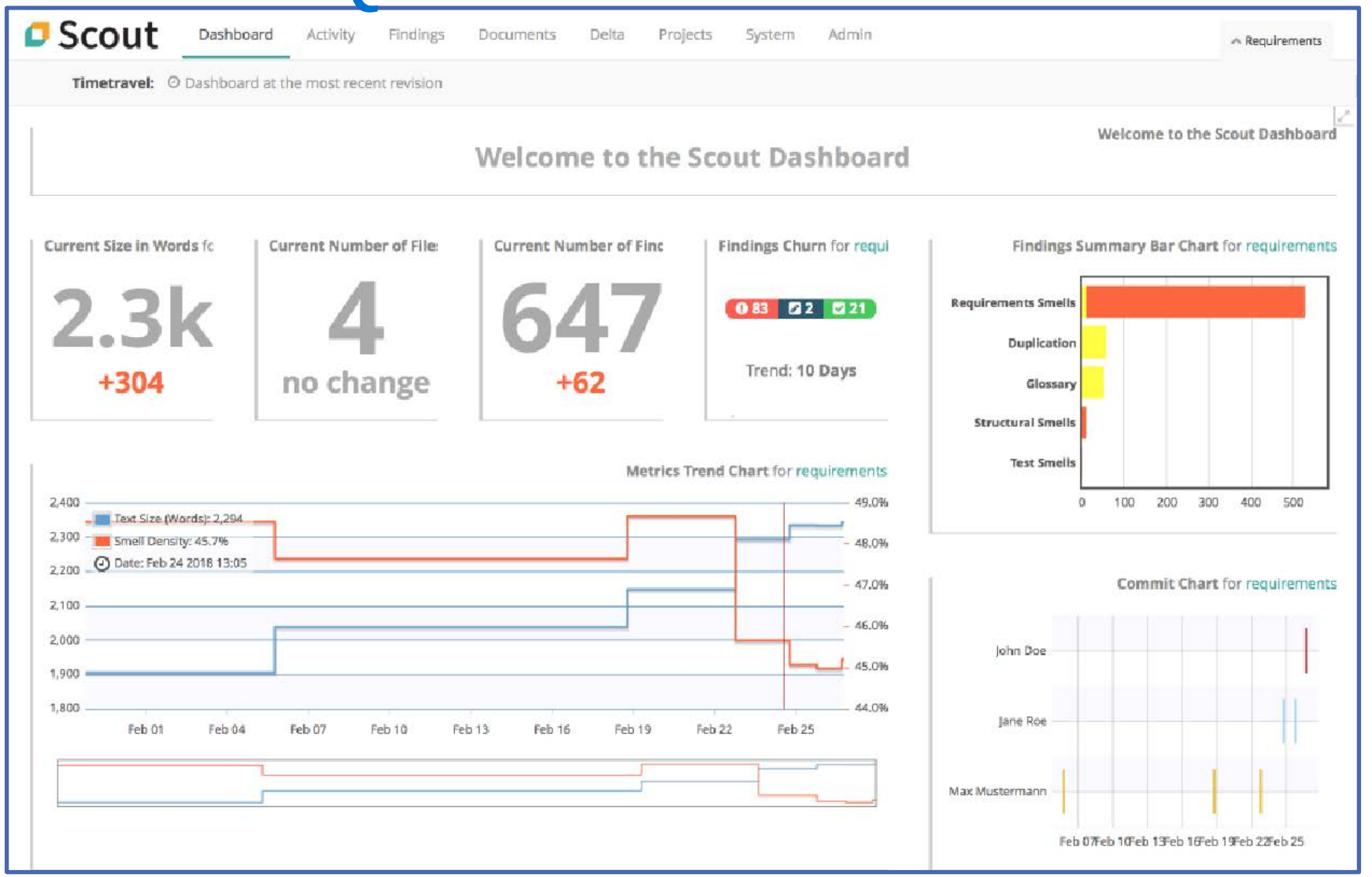
Key quality indicators

- Long, complicated sentences
- Passive voice
- Multiple negations
- Vague phrases and pronouns
- Comparatives and superlatives
- Dangerous slash
- UI details
- Cloning (same data not in sync
- ToDos

Scout from Qualicen



Scout from Qualicen



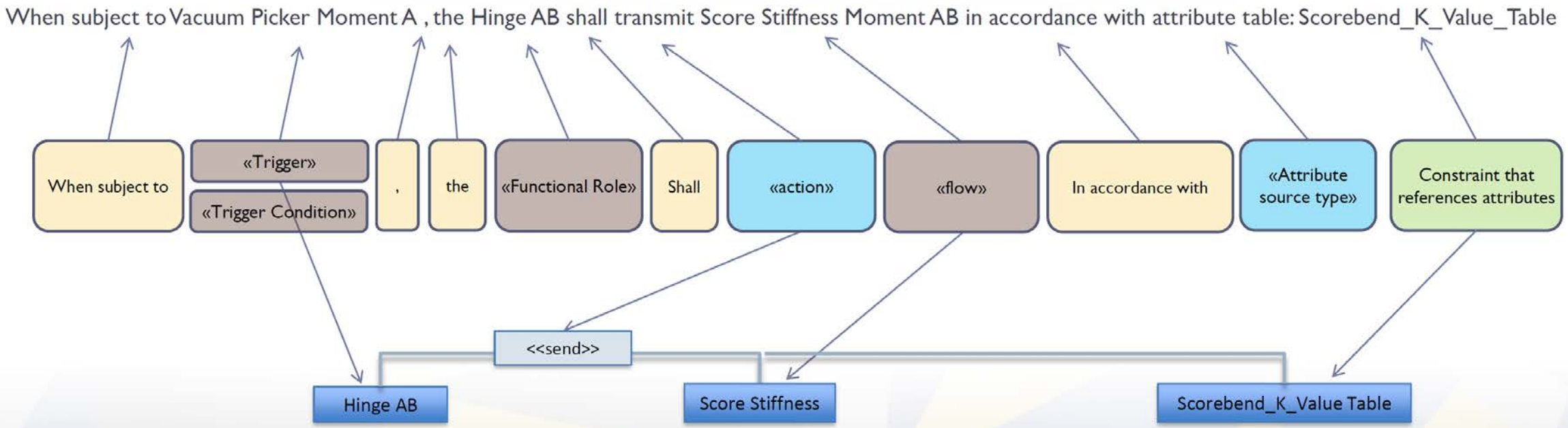
The REUSE Company

- Offerings
  - Requirements Authoring Tools (RAT)
  - V&V Studio
  - Traceability Studio
  - Knowledge Manager (KM)
- Unified model
  - Based on a single, integral, underlying graph
  - NLU is used to unify textual assets with modeling and simulation assets

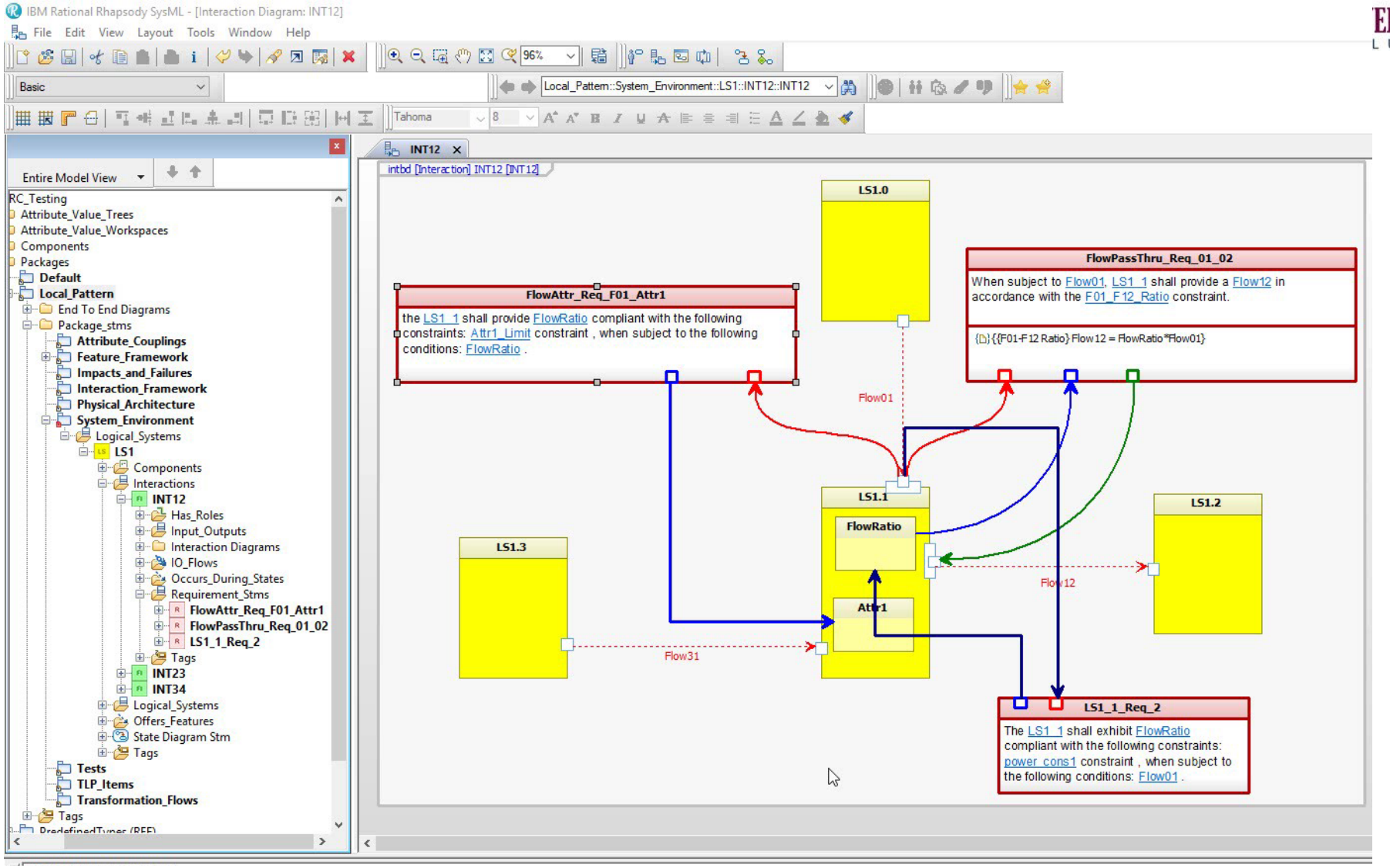
The REUSE Company

Structuring Requirements
A formal representation

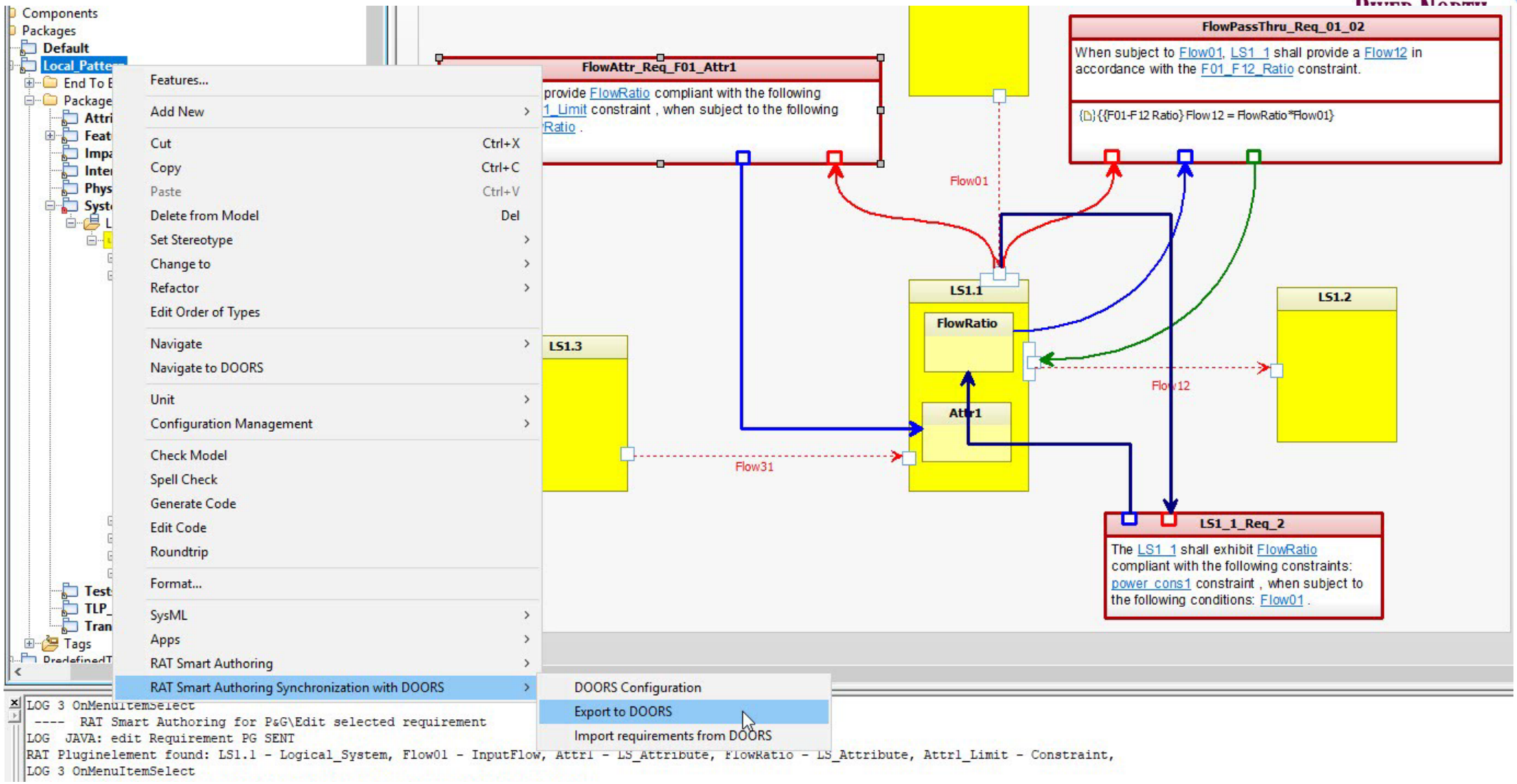
When subject to **Vacuum Picker Moment A**, **Hinge AB** shall **transmit Score Stiffness Moment AB** in accordance with **attribute table: Scorebend_K_Value_Table.**



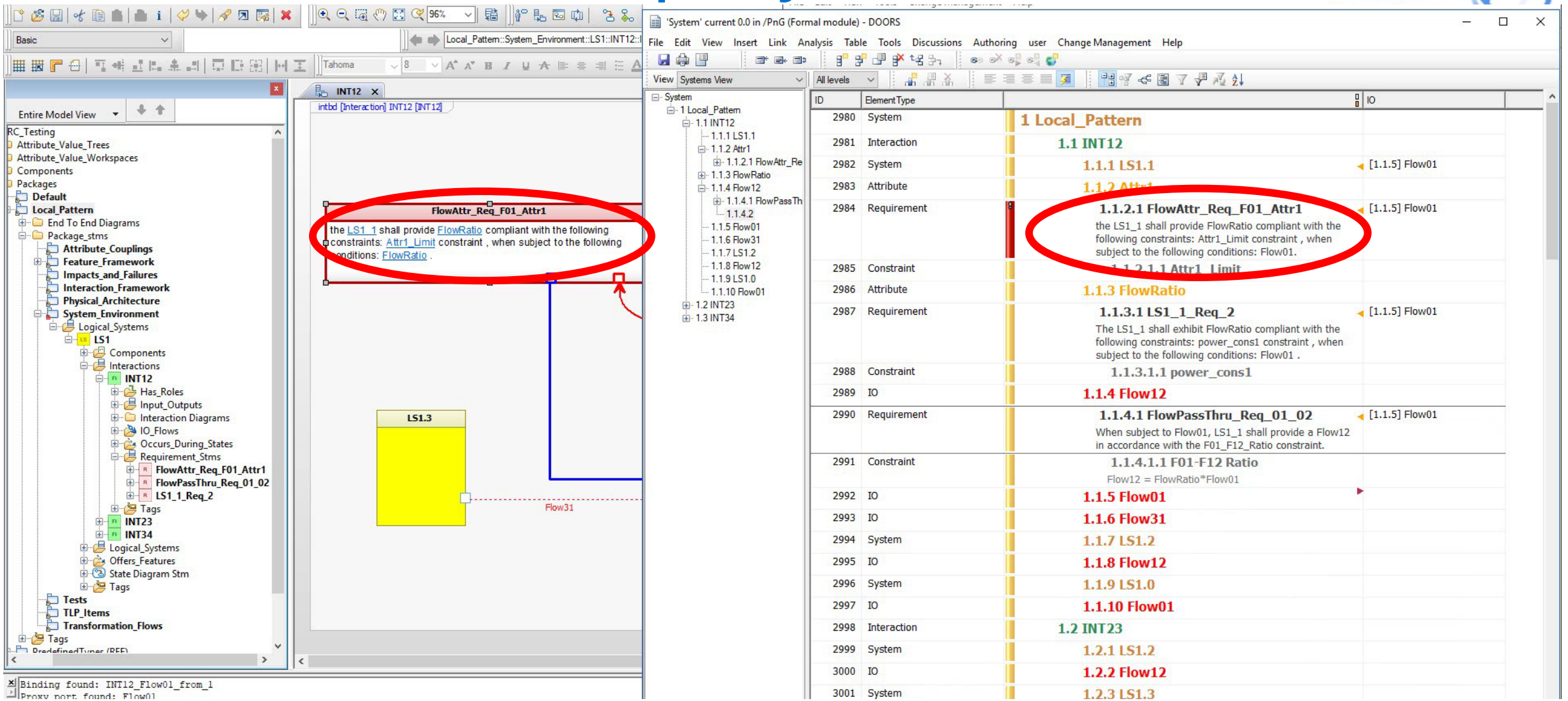
The REUSE Company



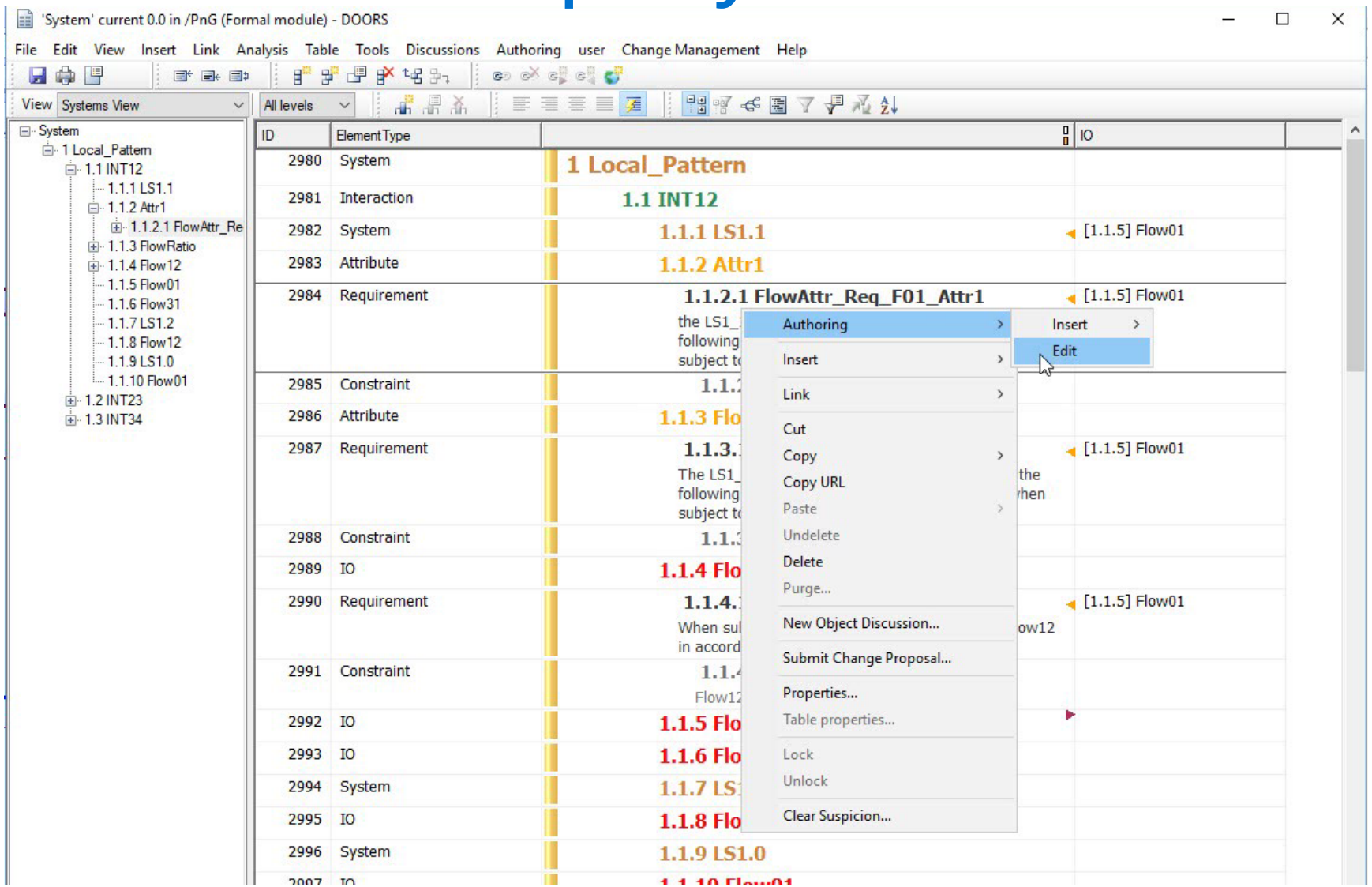
The REUSE Company



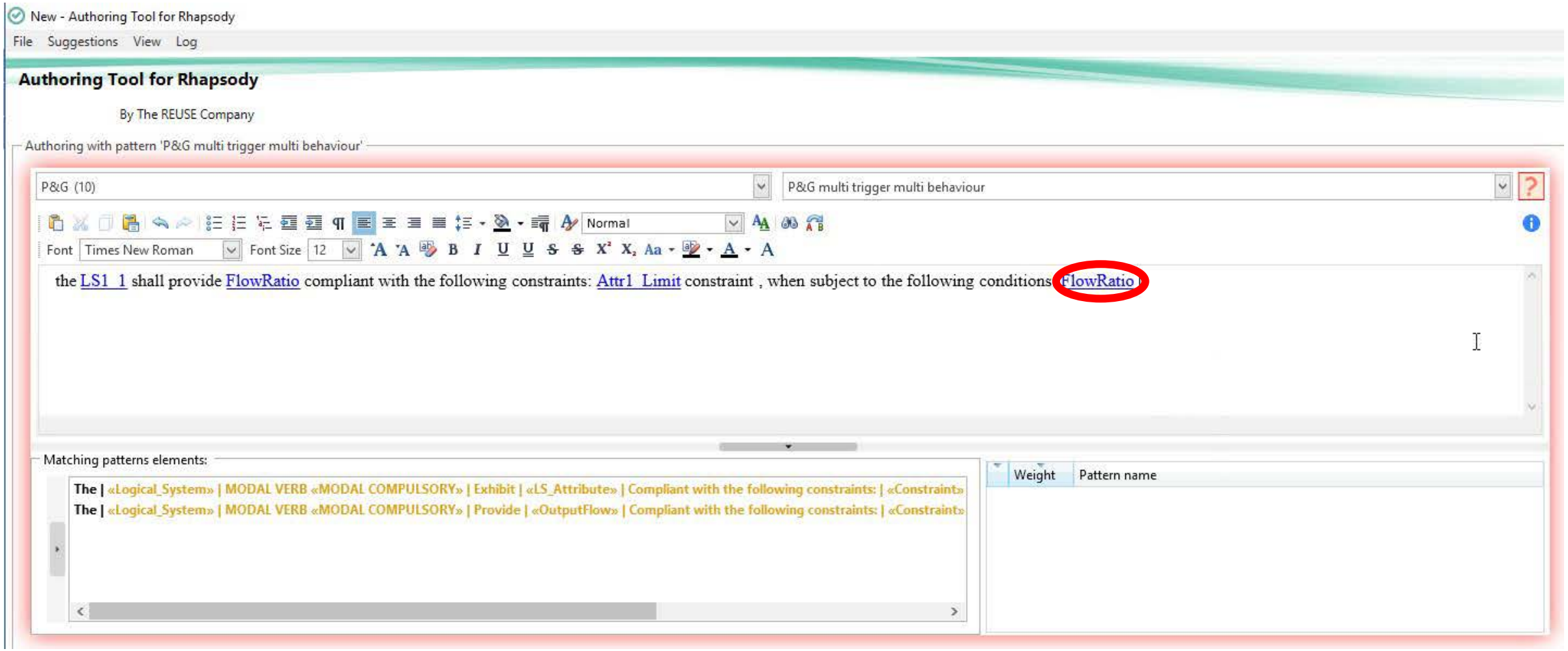
The REUSE Company



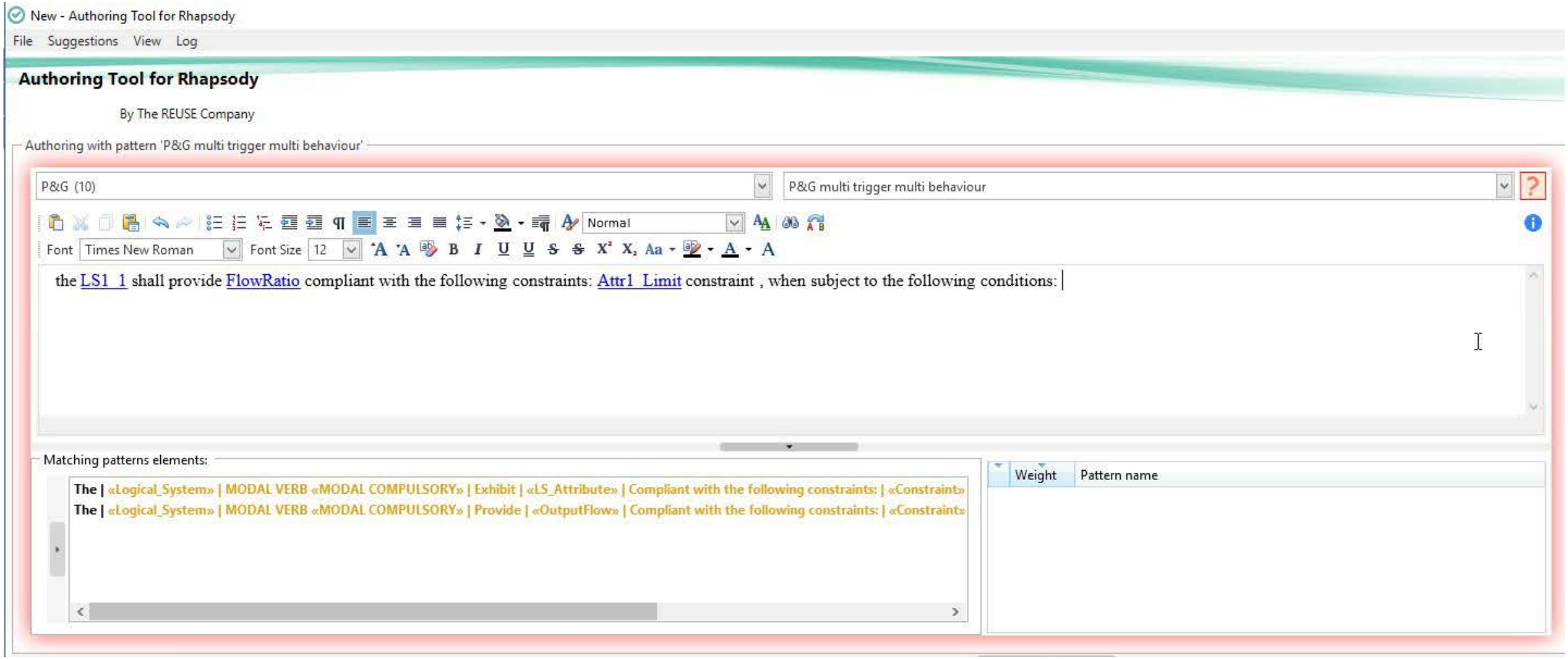
The REUSE Company



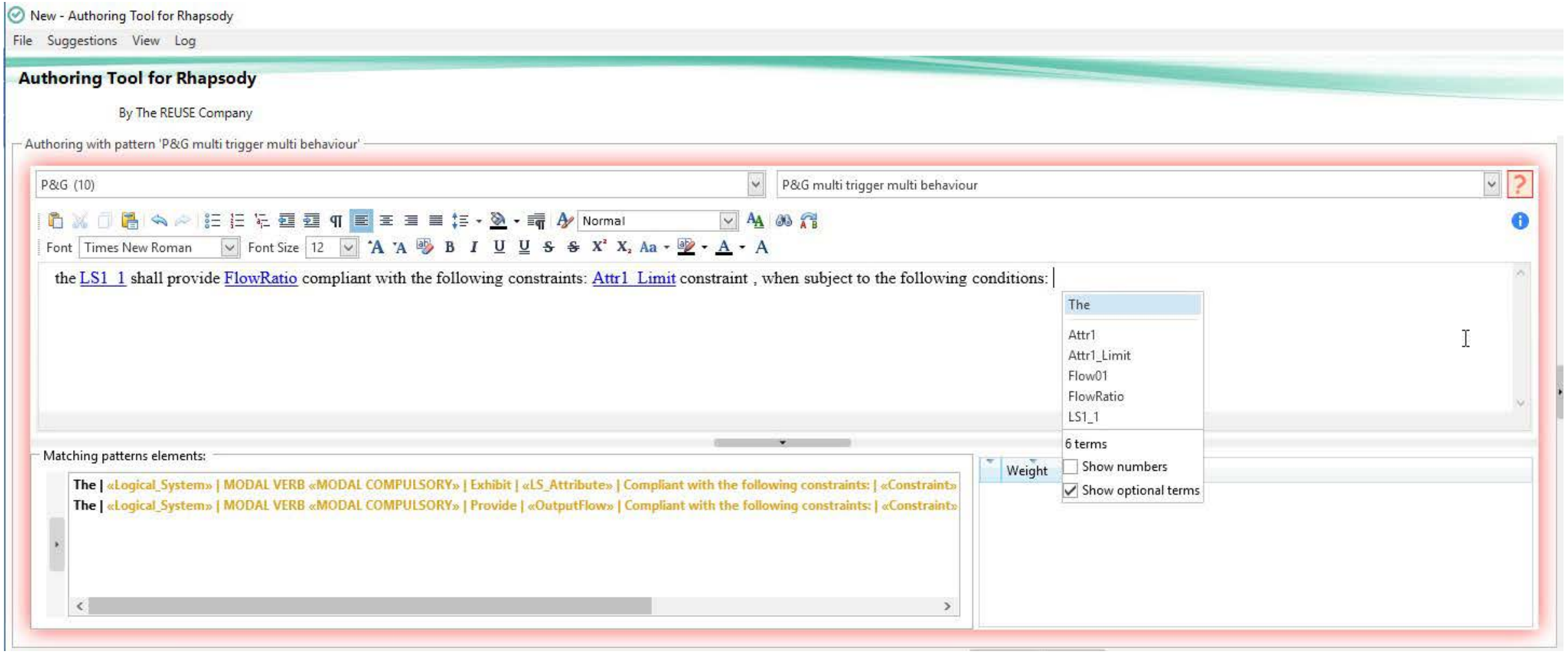
The REUSE Company



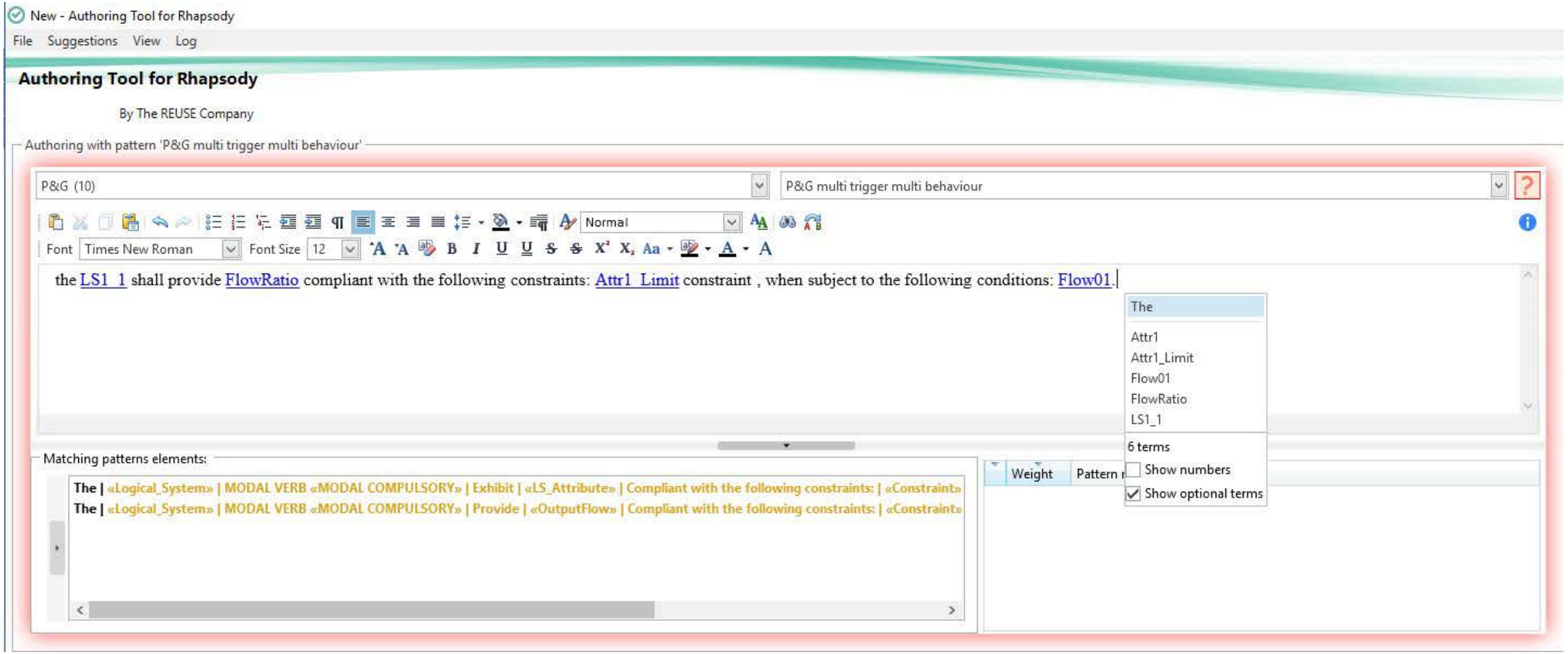
The REUSE Company



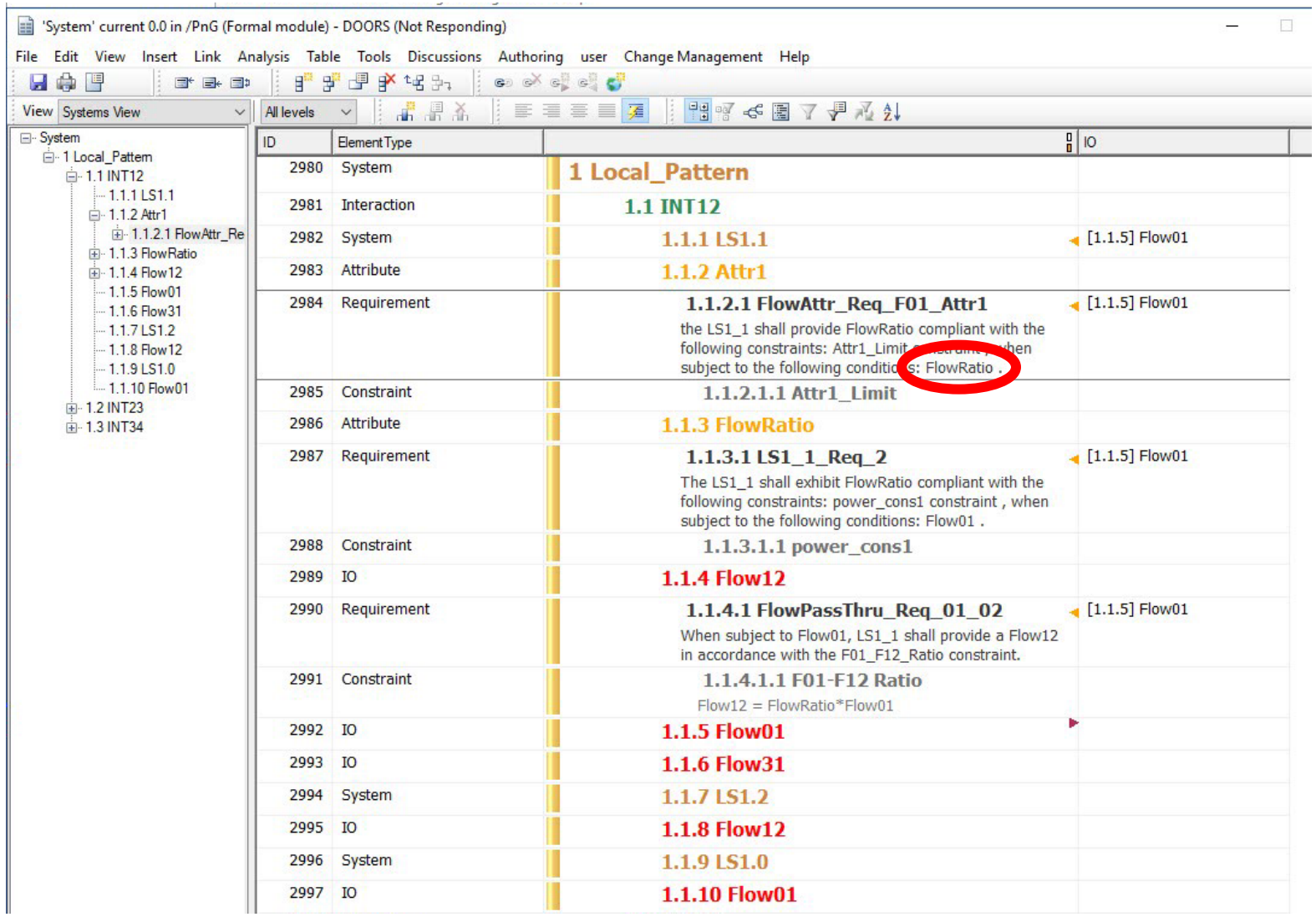
The REUSE Company



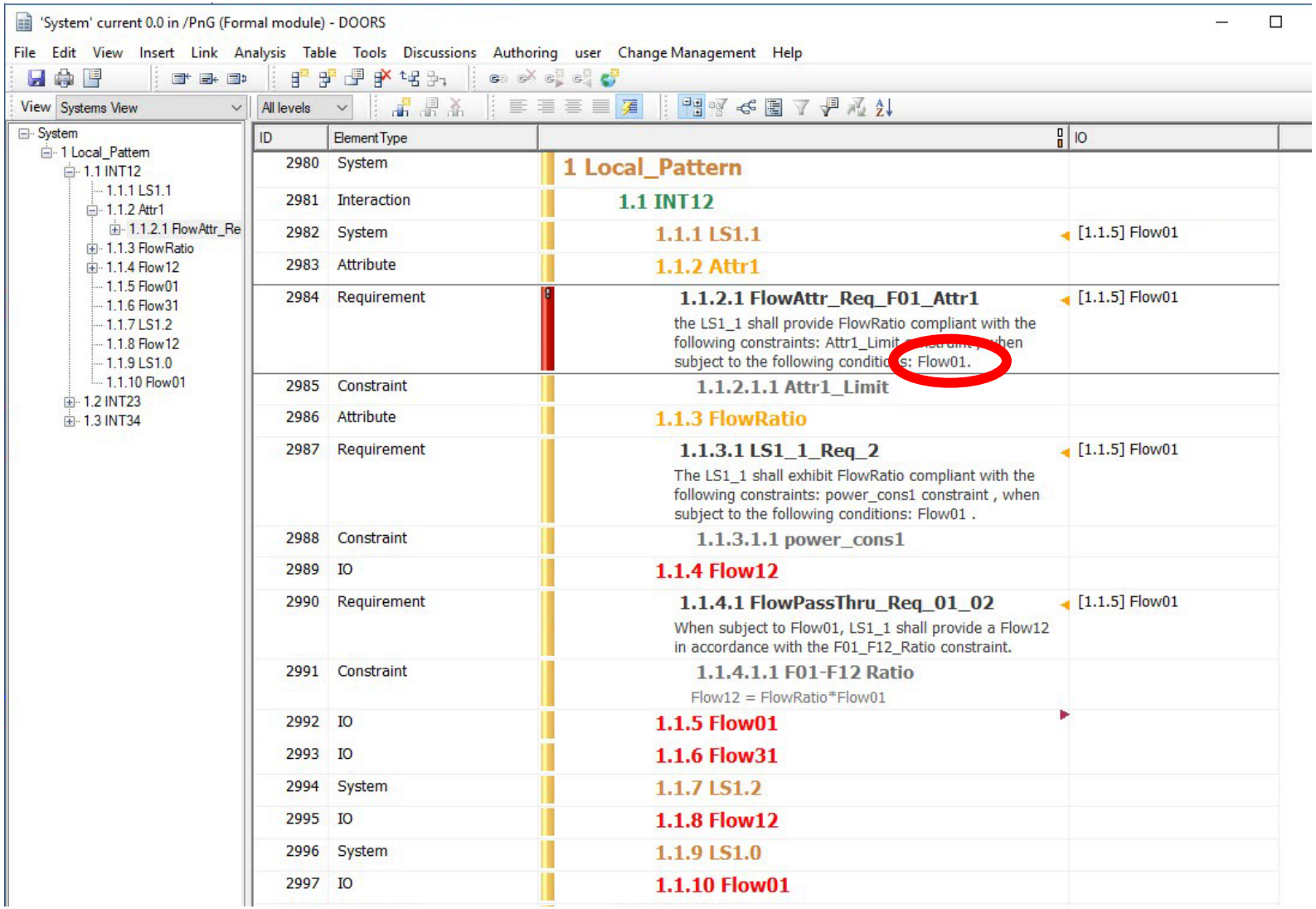
The REUSE Company



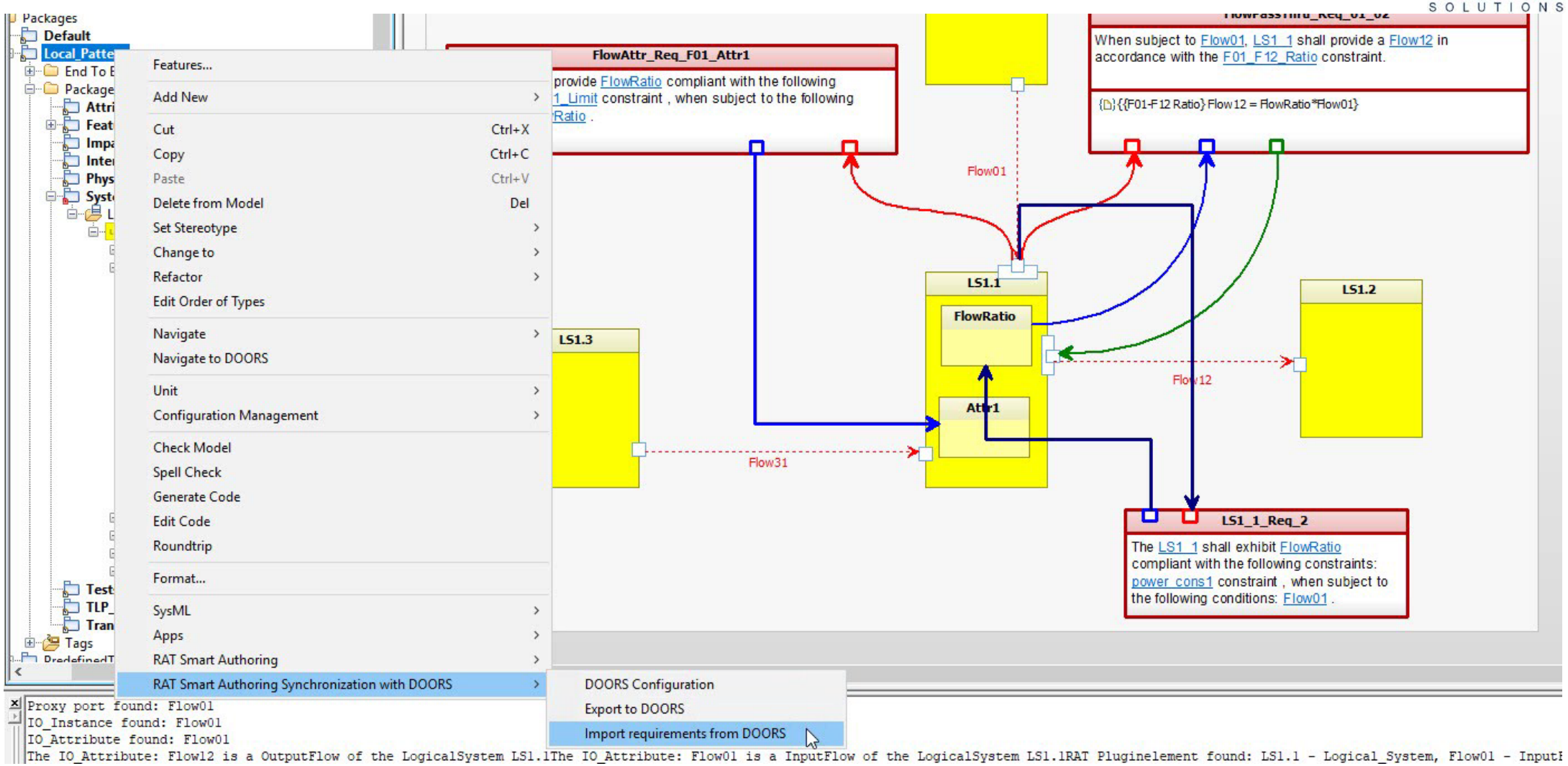
The REUSE Company



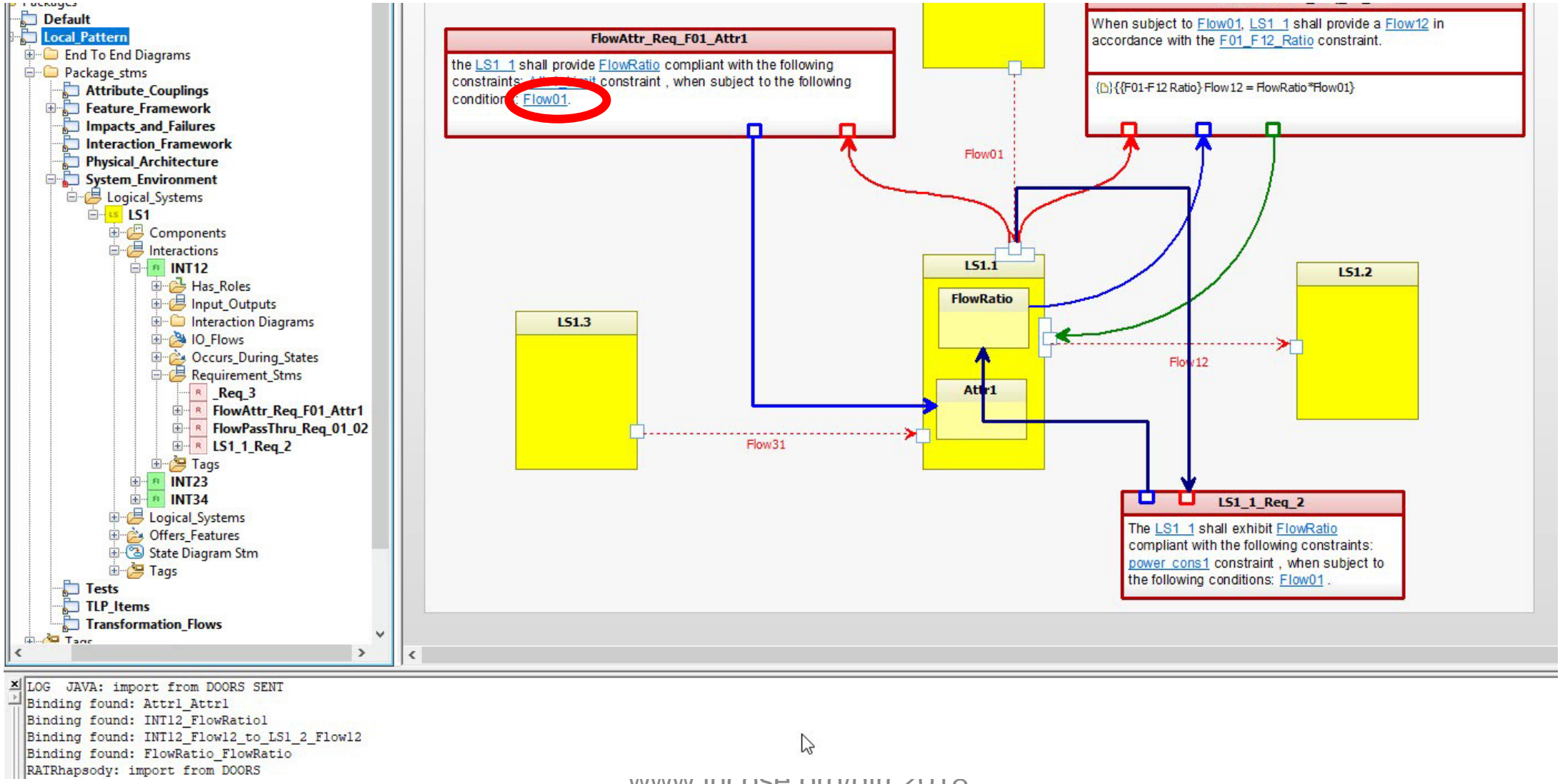
The REUSE Company



The REUSE Company



The REUSE Company

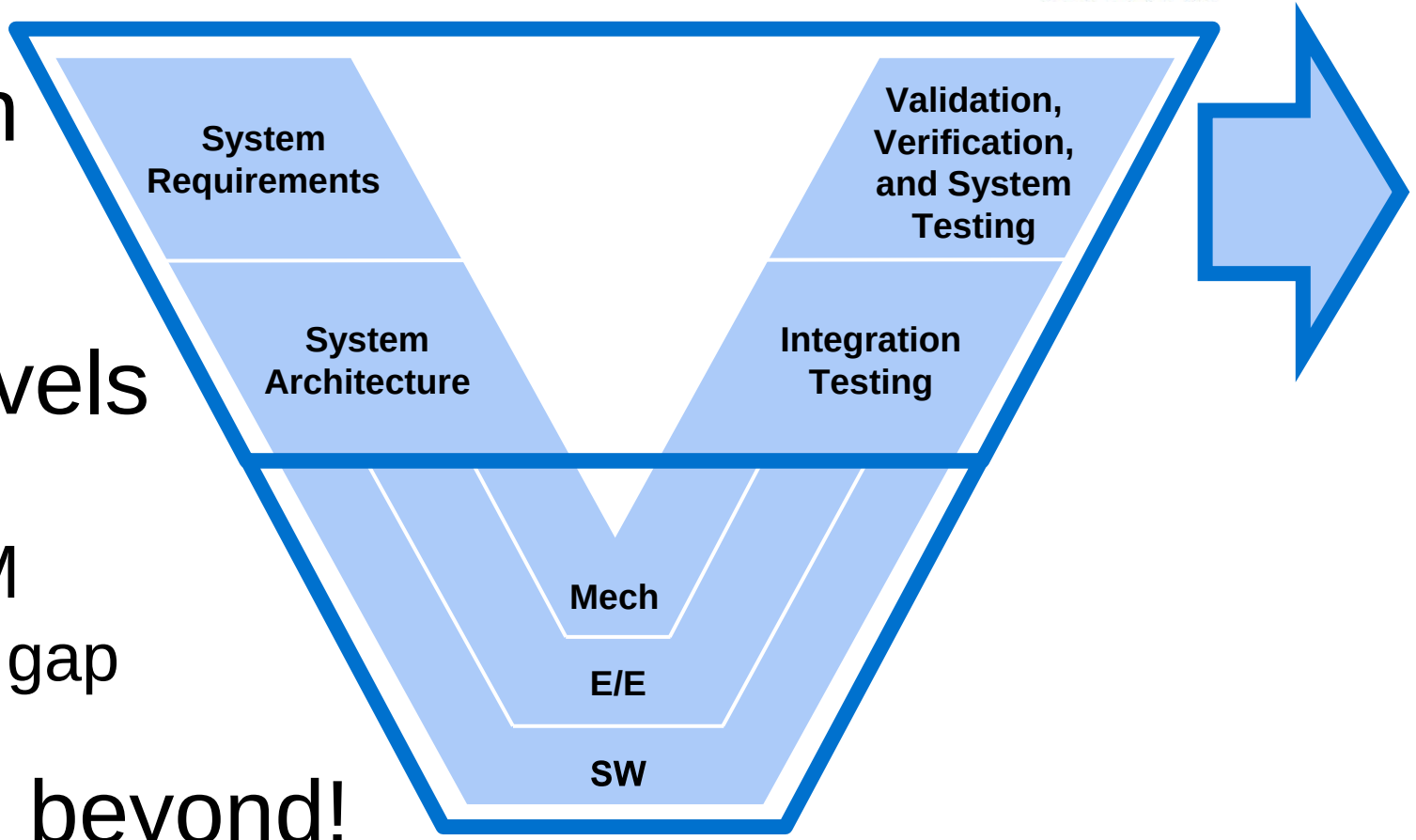


General benefits of cognitive requirements and test offerings

- Automation of time-intensive, error-prone tasks
- Pinpoints where focus needs to be delivered
- Prioritizes work
- Configurable for domain-specific applications
- Customizable to company policies and best practices
- Accelerates good requirements training for new users
- Accelerates high-quality requirements authoring even for experienced requirements engineers
- Accelerates review of customer requirements specs

Future of cognitive engineering

- Current focus is on top left/right of V
- Moving to lower levels
  - All product data
  - Cognitive ALM-PLM
    - To help bridge the gap
      - To the V and beyond!
        - Including manufacturing, service, and operations



Summary

- Cognitive engineering is maturing and growing rapidly
- Cognitive requirements and test tools can enable:
  - Higher productivity and efficiency
  - Immediate feedback to author for faster learning
  - Moving defect discovery to early development phases
  - Less downstream churn for requirements, design, test
  - Reducing product development cost and schedule
- Technology is moving fast in this space
- Expect rapid developments in the near term

Challenge

Pilot these capabilities

Report back next year

- at IW, IS, and/or GLRC

Questions?

2018 Annual INCOSE
Great Lakes Regional Conference
SYSTEMS AT THE CROSSROADS
17 - 20 October 2018 | Indianapolis, Indiana

www.incose.org/glrc2018

Bob Parro, River North Solutions - bparro@rivernorthsolutions.com
Steve Denman, Stephen D. Denman Consulting, LLC - stephen@sddenmanconsulting.com
Kevin McHugh, IBM Watson IoT Lab Services - mchughm@us.ibm.com