Improving IV&V Techniques Through the Analysis of Anomalies

Tim Menzies (WVU)
8/12/08

Problem

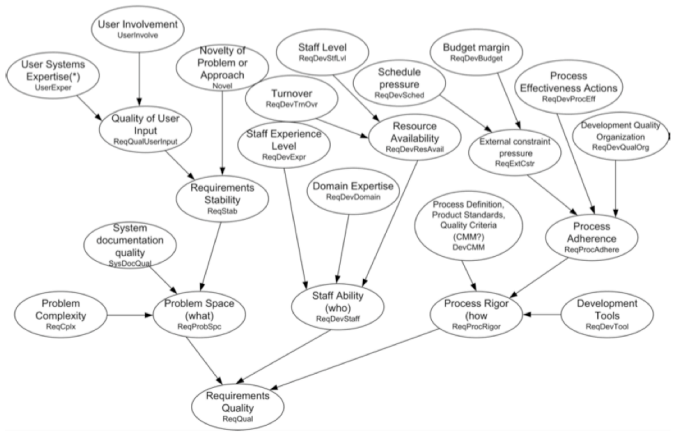
- How do we recognize an anomalous project?
  - By using automatic analysis of NASA IV&V databases. SILAP, etc
- But database structures at NASA IV&V are continually changing.
  - How do we deal with this constant evolution?

Approach

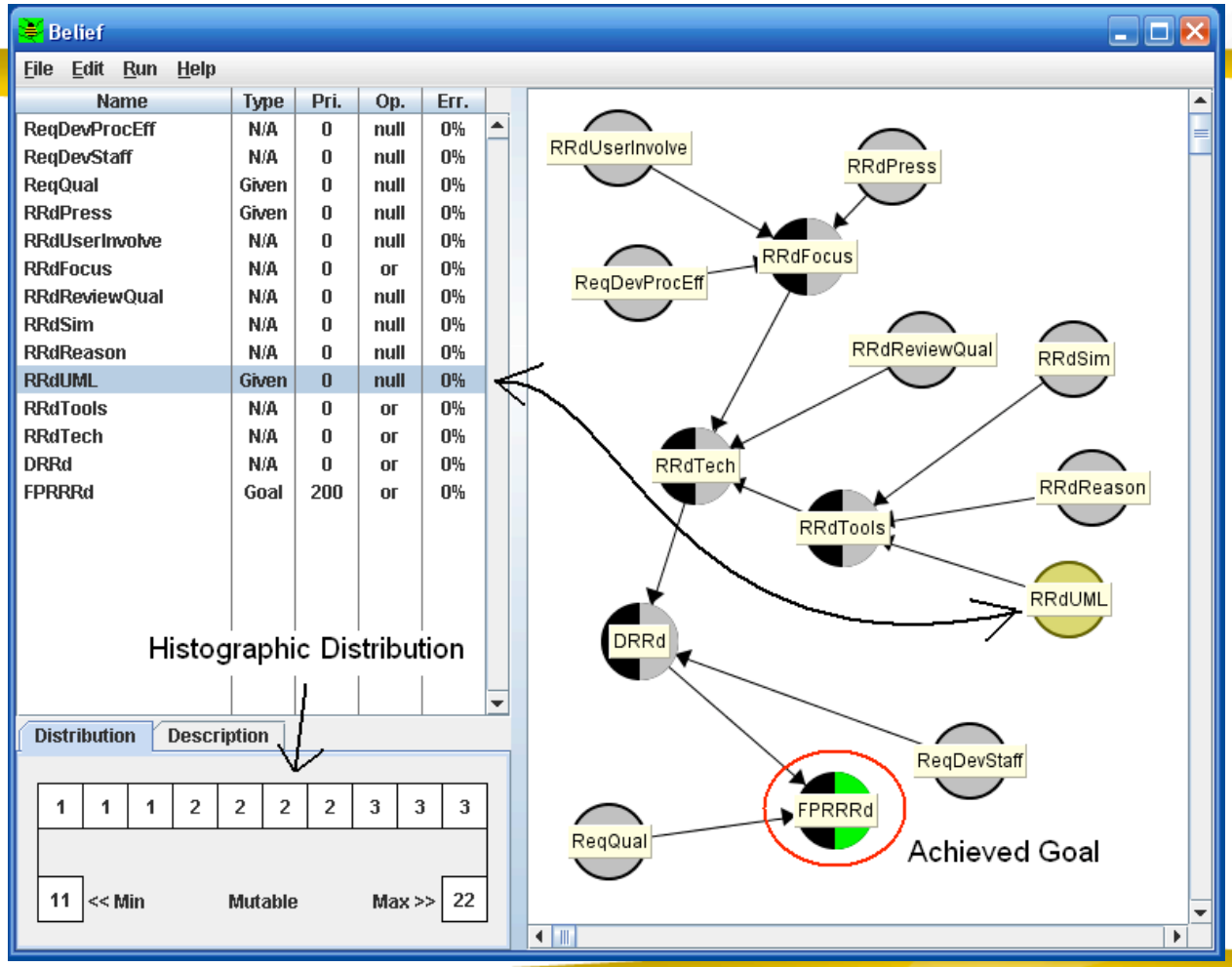
- The problems with data
  - The sheer amount of data available may make direct analysis impractical or even impossible
  - Most project data is ephemeral and/or subjective, making it hard to construct tools for direct analysis
- New method
  - Build a fast-change format for intuitions about a project
  - Use manager expectations for those intuitions
  - Initialize the intuitions using Dabney's Bayes nets
  - Build a GUI to make the change easy

Approach (details)

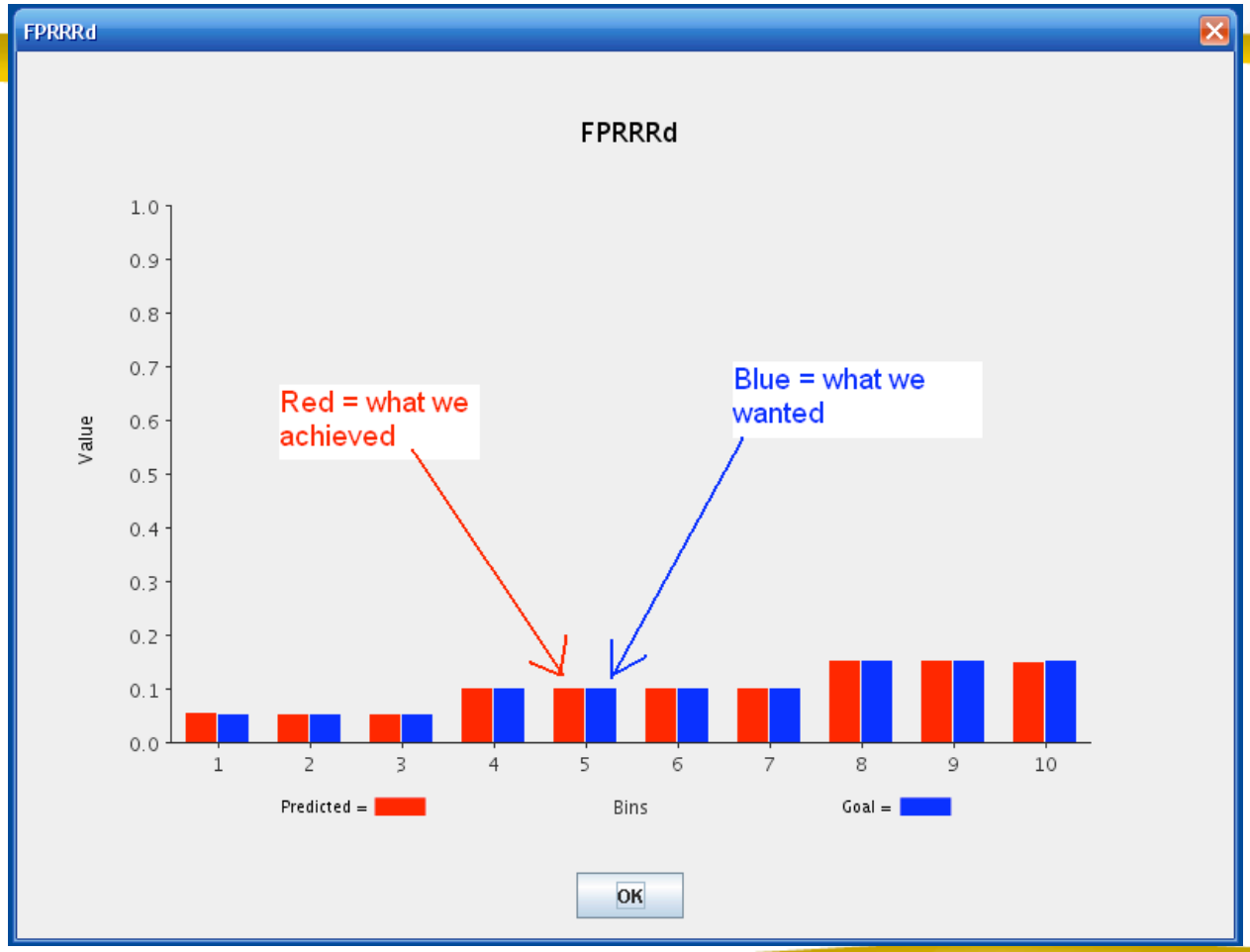
- Create a network that represents a project
  - Nodes are factors within the project
  - Edges indicate the flow of influence
- Insert manager expectations into the network as specific goals
- Insert domain knowledge as specific data about certain factors
  - Use the tool to estimate the rest
- The backend tool evaluates the network and attempts to achieve expectations based on input knowledge
- A frontend GUI component allows the user to input data and quantify the results
  - (see next page)



GUI component



GUI component (contd.)



Relevance to NASA

- Management of software projects = difficult
  - How to control them?
  - New methods introduce new opportunities for anomalies
  - Anomaly detection for new methods is not always easy
  - Anyone can see the fire. Can you see the smoke?
- Old world: collect measures.
  - But what to collect
- Here: use background knowledge (from Dabney) to inform data collection and interpretation
  - Generalized strategies for detection and minimization are needed
  - From anomaly detection to repair

Accomplishments

- After a first round of user trials with…
  - Experienced NASA developers/managers
- …refinements to usability and addition of features

- Migration of the backend to Java
  - 0.016 seconds, not 43 seconds
  - Allows more direct interfacing with the actual network evaluation aspects
  - First version of an ability to generate recommendations for refinement of expectations

Next Steps

- Required: More user trials
- Update usability requirements based on feedback
- Implement further usability features