Improving IV&V Techniques Through the Analysis of Anomalies

Tim Menzies (WVU), tim@menzies.us
Jason Gookins
8/12/08

Problem

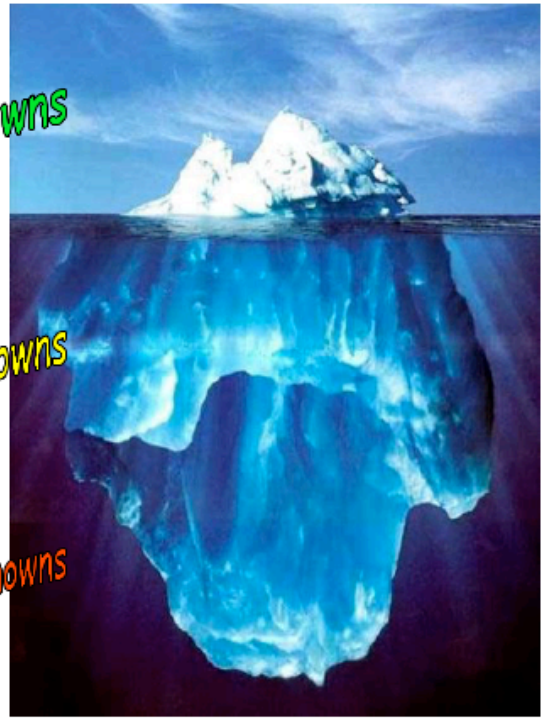
*Flying Safely to 2020 and beyond means attacking **relentlessly** all three levels of the risk iceberg!*

- Brian O'Connor
March 20, 2003

- **Mishap recommendations**
- **Problem solutions**
- **IFA fixes**

- **FMEA/Hazard controls**
- **Close call recommendations**

- **Ignored close calls?**
- **Old cert, new environment?**
- **Inadvertent excursions out of cert/family?**
- **Hardware talking…nobody listening?**

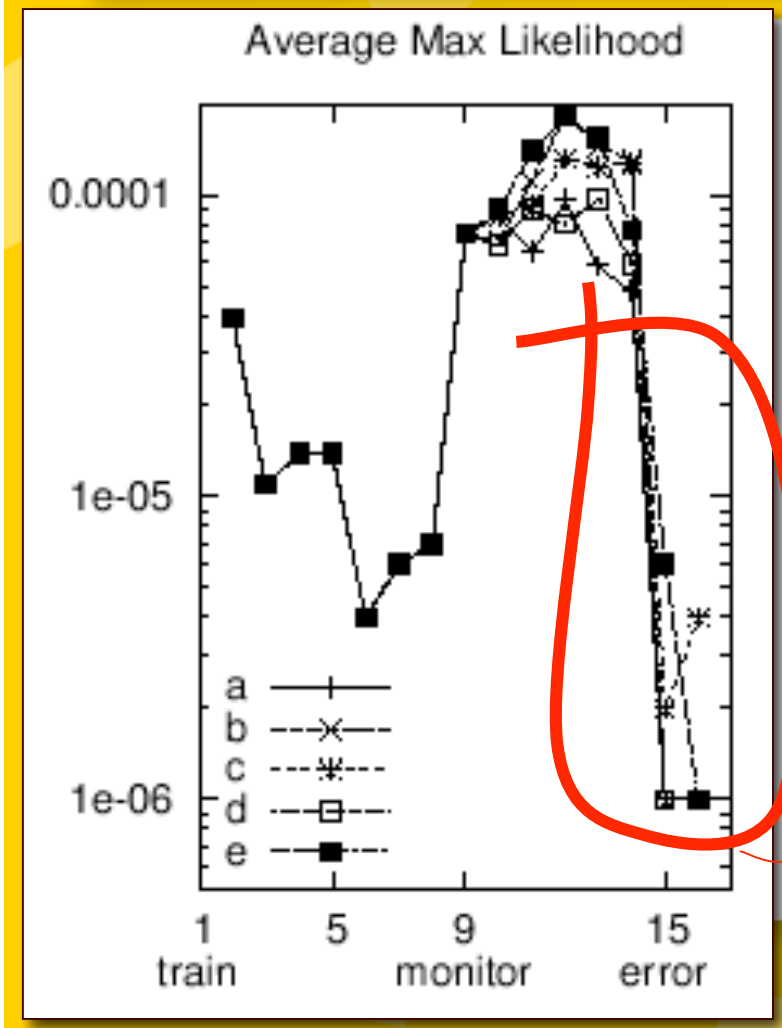


But how to know what we don't know?

How we thought we'd do it (queue creep in music)

- Find data sources used routinely at IV&V
  - PITS issue tracking system
    - Can build serverity predictors
      - [Menzies&Marcus ICSM'08]
      - http://menzies.us/pdf/08severis.pdf
  - MDP metrics data repository
    - Can build defect predictors
      - [Menzies et.al. TSE'07]
      - http://menzies.us/pdf/06learnPredict.pdf
  - SILAP data set
    - Can build severity* frequency predictor
      - [Menzies, Benson Costello, Moats, Northey, Richardson, NASA Innovations SysSoft Eng'08]
      - http://menzies.us/pdf/07ivv.pdf
  - LINKER meta-database (one ring to rule them all)
- Instrument them : report if they go "odd"

How we thought we'd do it count, alert, fix



Count: stuff seen in past
Alert: if new counts different
Fix: find delta new to old
- Very, very fast

An incremental discretizer + a Bayes classifier where all inputs are all mono-classified

Track average max likelihood for data processing in "era"'s of X instances

Contrast set learning

Linear time inference, Tiny memory footprint

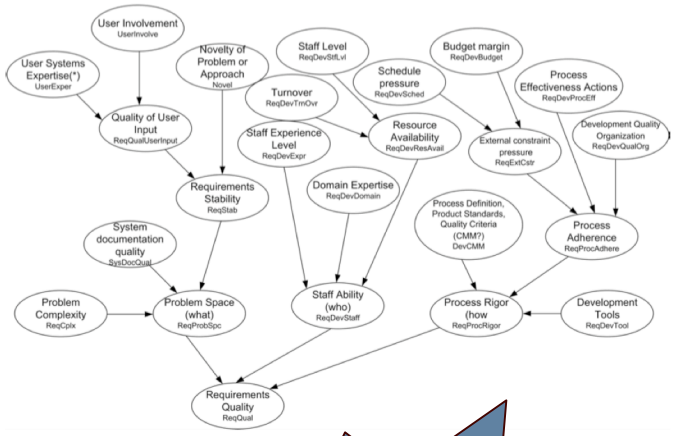
- And, it works [Orrego, 2004]
  - F15 simulator data [courtesy B. Cukic]
  - Five flights: a,b,c,d,e
  - each with different off-nominal condition imposed at "time" 15
  - Off-nominal condition not present in prior data
  - In all cases, **massive change detected**

Problem

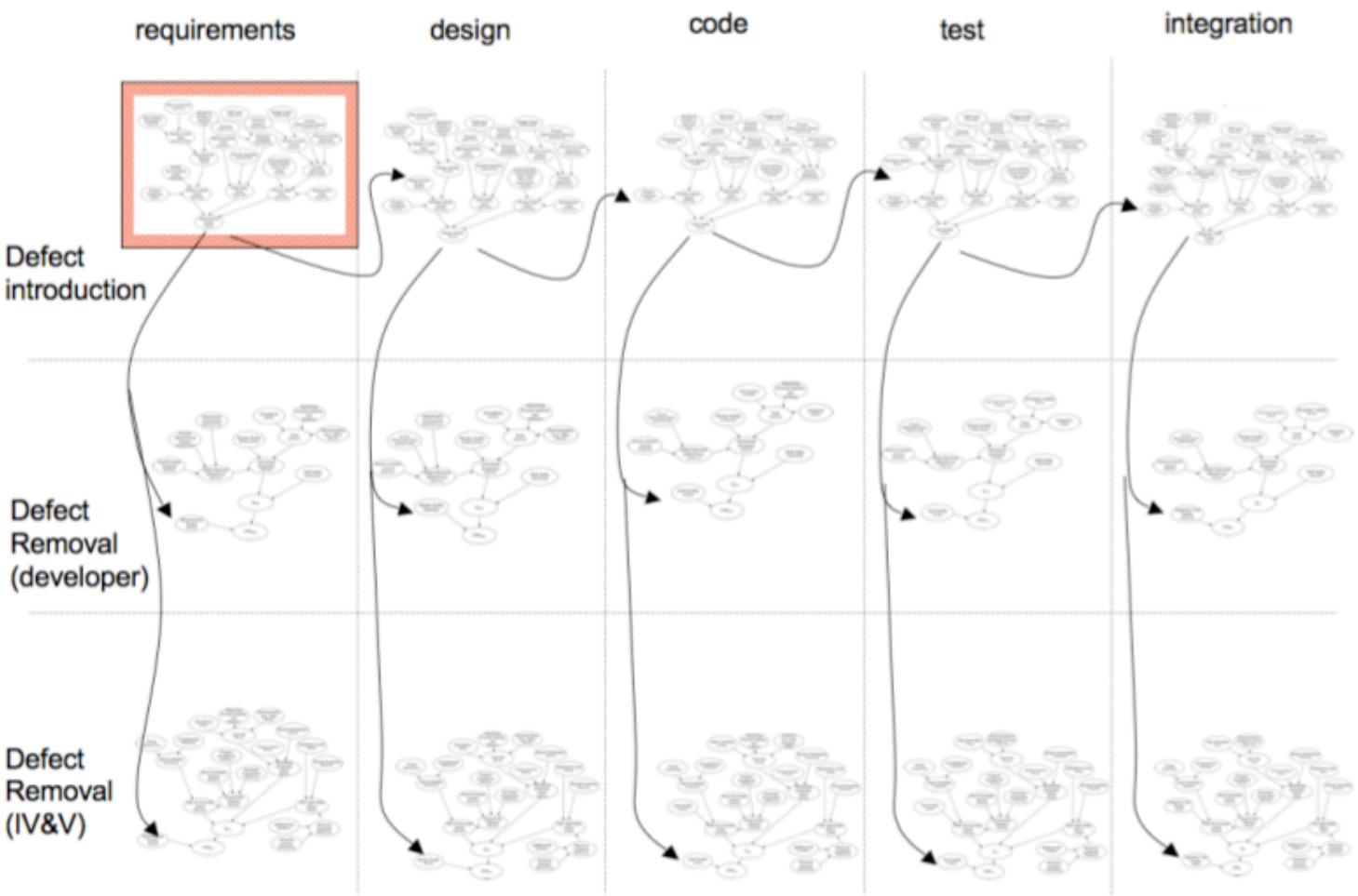
- SILAP, PITS, MDP, LINKER are
  - in a state of flux
  - being phased
  - are now historical
  - are being reorganized
- Hard to instrument X
  - When X keeps moving away
- Q: what is constant now?
  - and what will remain constant in the future?
- The data always seems to change
  - But the fact that people read the data, seems constant
  - So lets model the people reading the data, and not the data

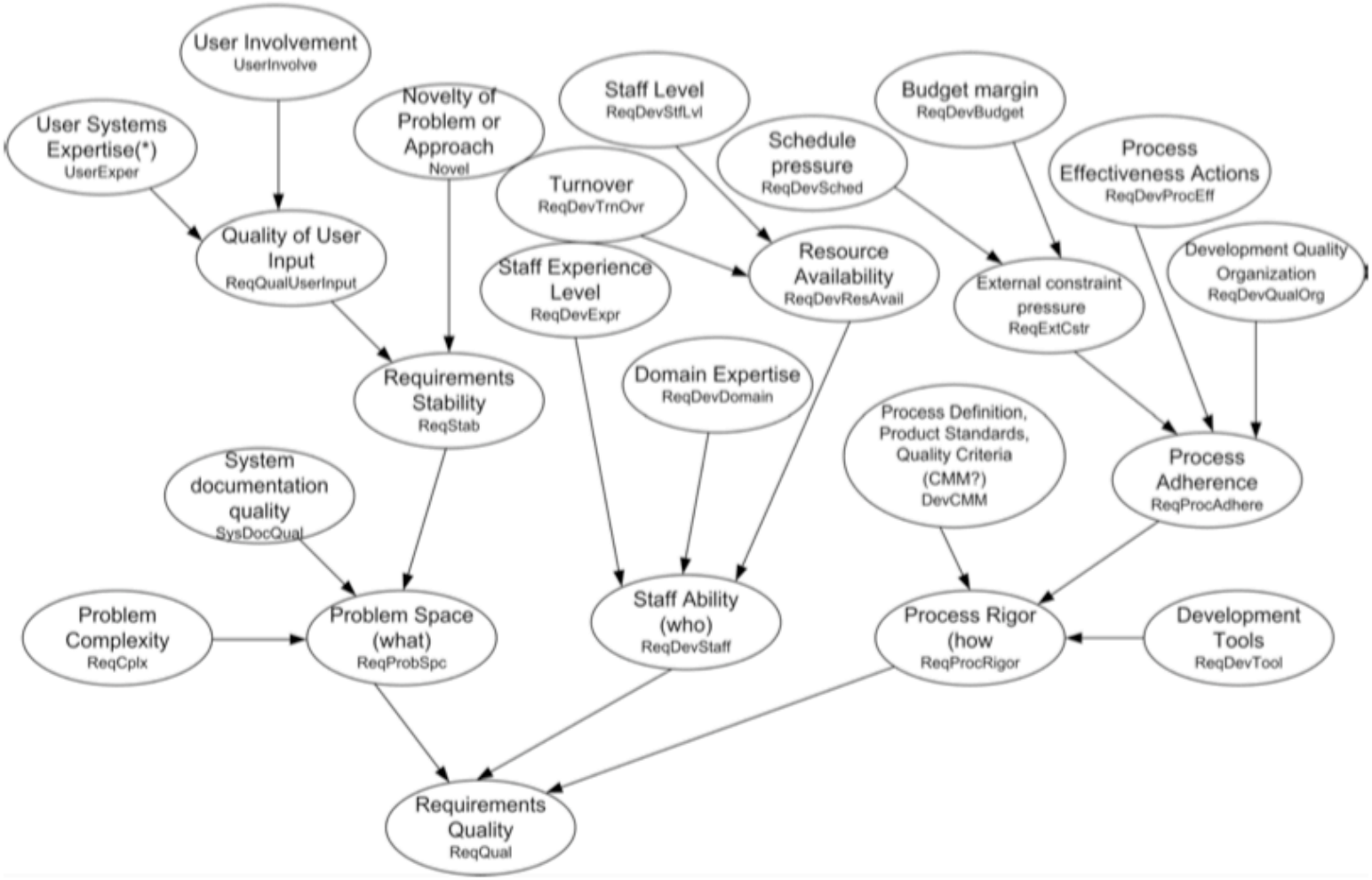
Approach (details)

- 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 inputted knowledge
- A frontend GUI component allows the user to input data and quantify the results
  - (see next page)

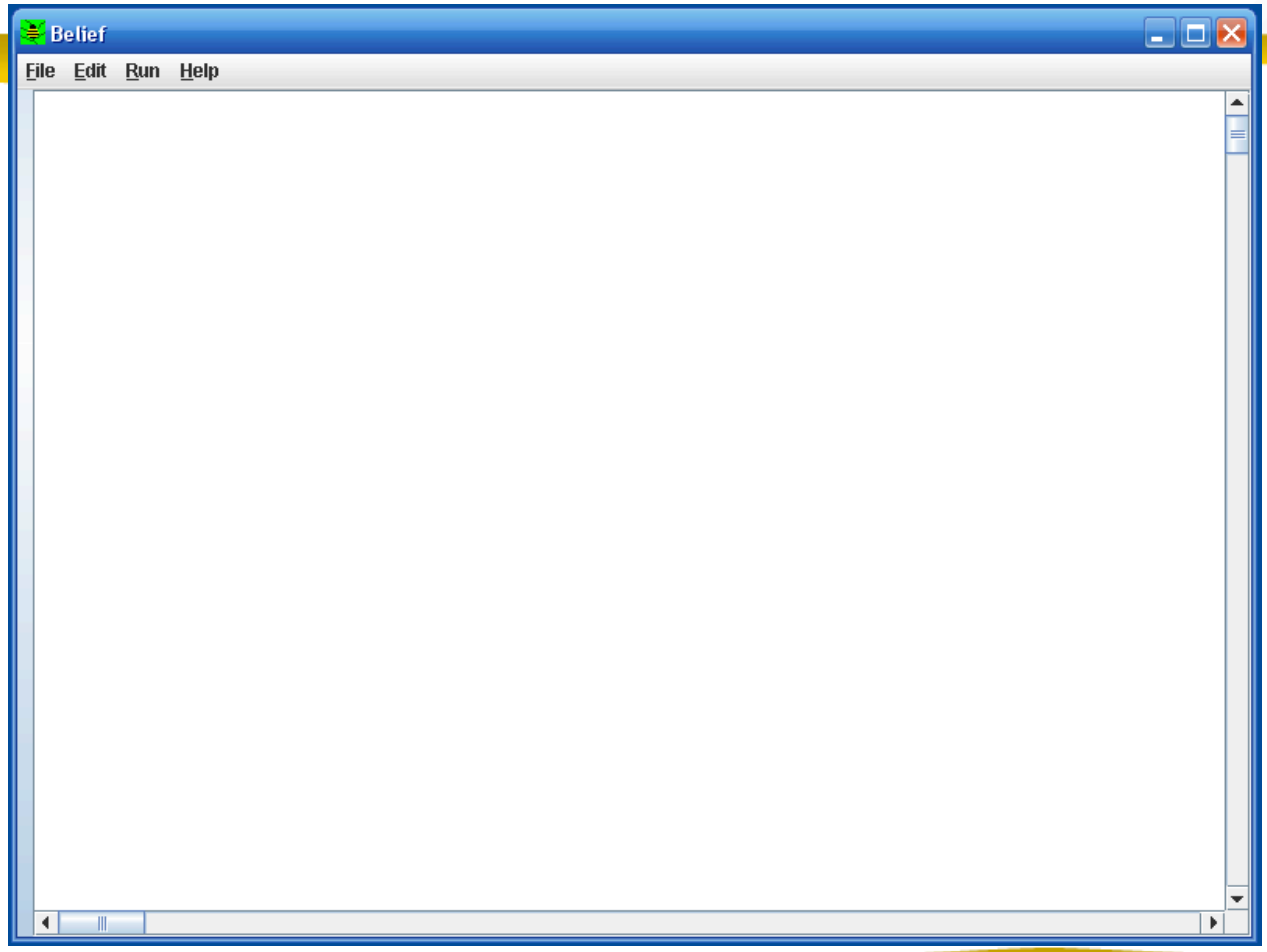


Initialize graph with Dabney's Bayes net descriptions of IV&V

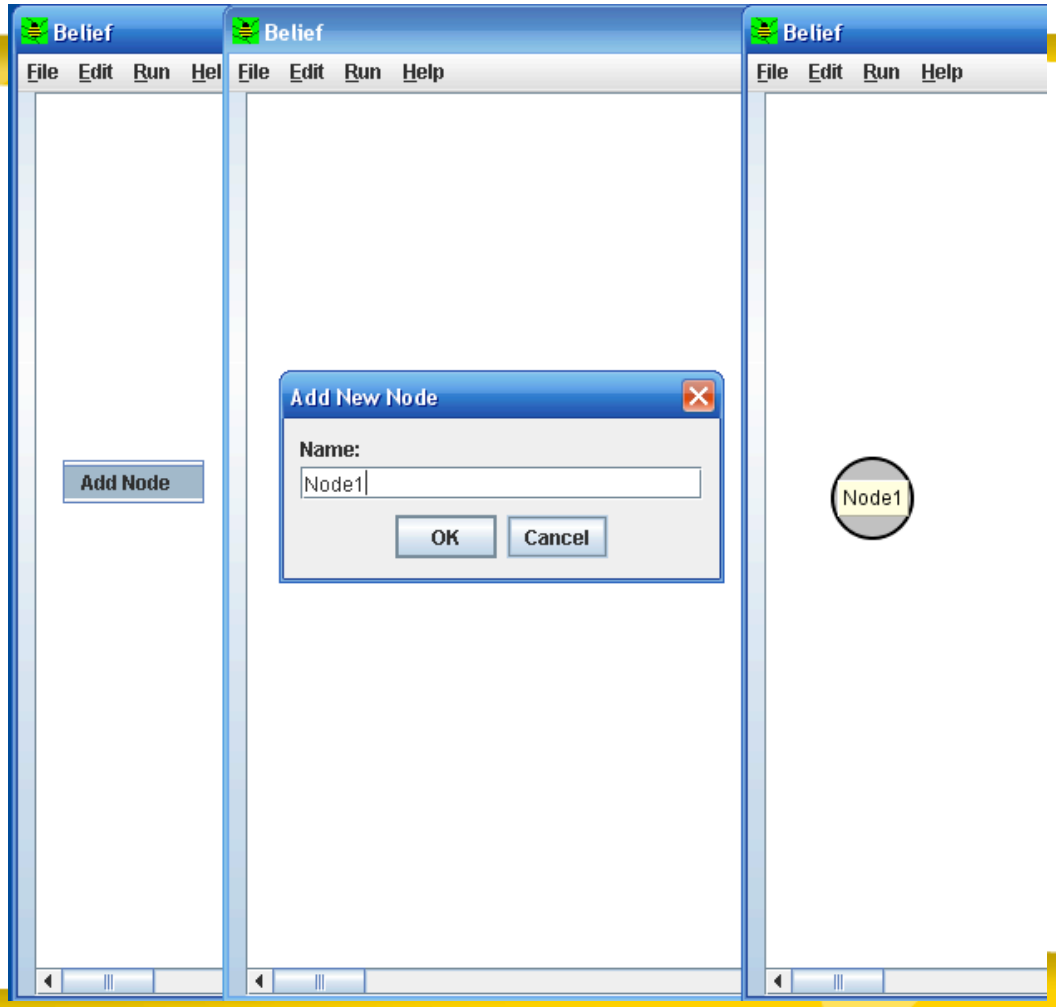

requirements
design
code
test
integration
Defect introduction
Defect Removal (developer)
Defect Removal (IV&V)


User Involvement
UserInvolve
User Systems Expertise(*)
UserExper
Novelty of Problem or Approach
Novel
Staff Level
ReqDevStfLvl
Budget margin
ReqDevBudget
Schedule pressure
ReqDevSched
Process Effectiveness Actions
ReqDevProcEff
Turnover
ReqDevTrnOvr
Quality of User Input
ReqQualUserInput
Staff Experience Level
ReqDevExpr
Resource Availability
ReqDevResAvail
External constraint pressure
ReqExtCstr
Development Quality Organization
ReqDevQualOrg
Requirements Stability
ReqStab
Domain Expertise
ReqDevDomain
Process Definition, Product Standards, Quality Criteria (CMM?)
DevCMM
Process Adherence
ReqProcAdhere
System documentation quality
SysDocQual
Problem Complexity
ReqCplx
Problem Space (what)
ReqProbSpc
Staff Ability (who)
ReqDevStaff
Process Rigor (how
ReqProcRigor
Development Tools
ReqDevTool
Requirements Quality
ReqQual

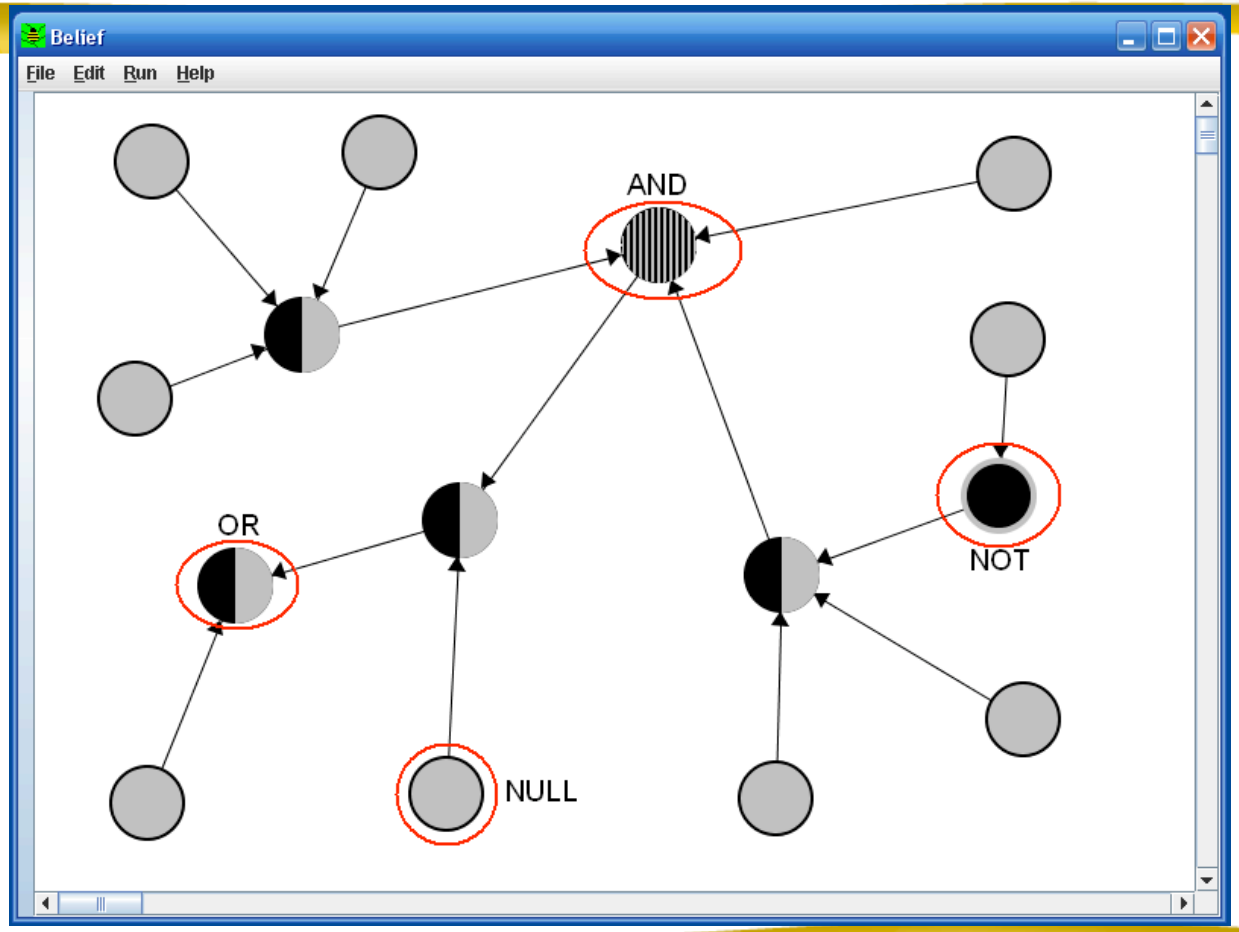
GUI component – Starting screen



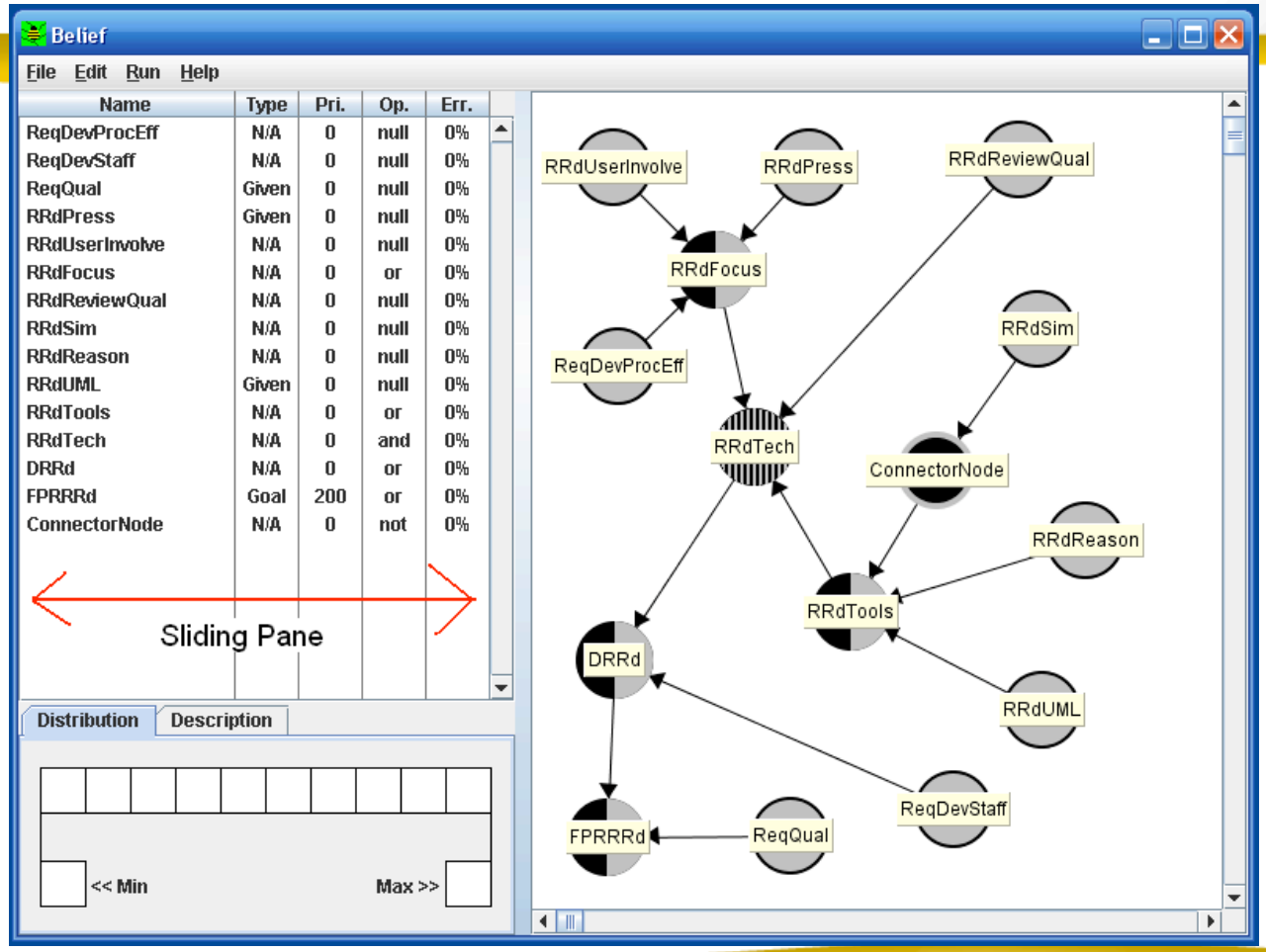
GUI component – Adding a node



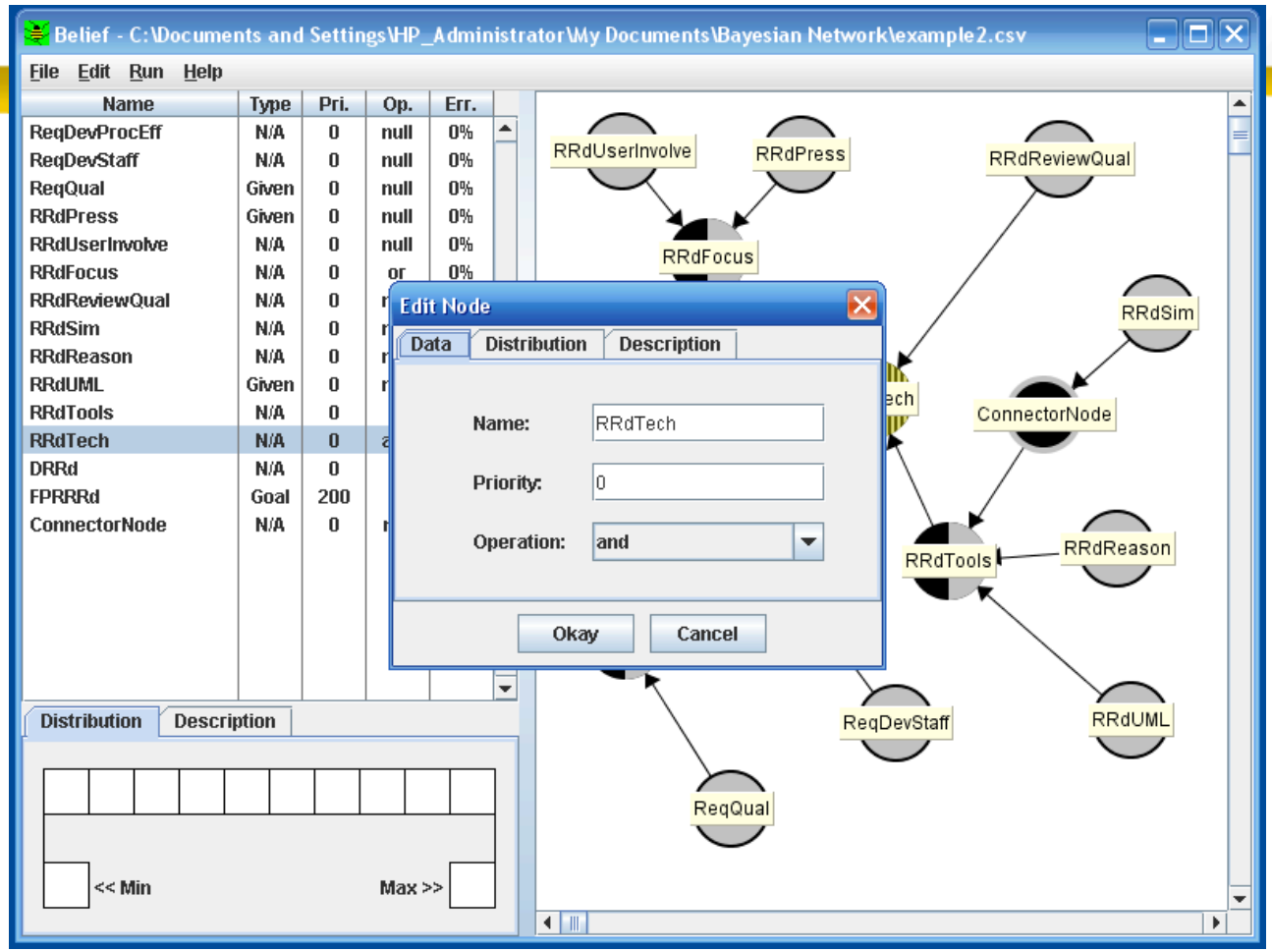
GUI component – Node types



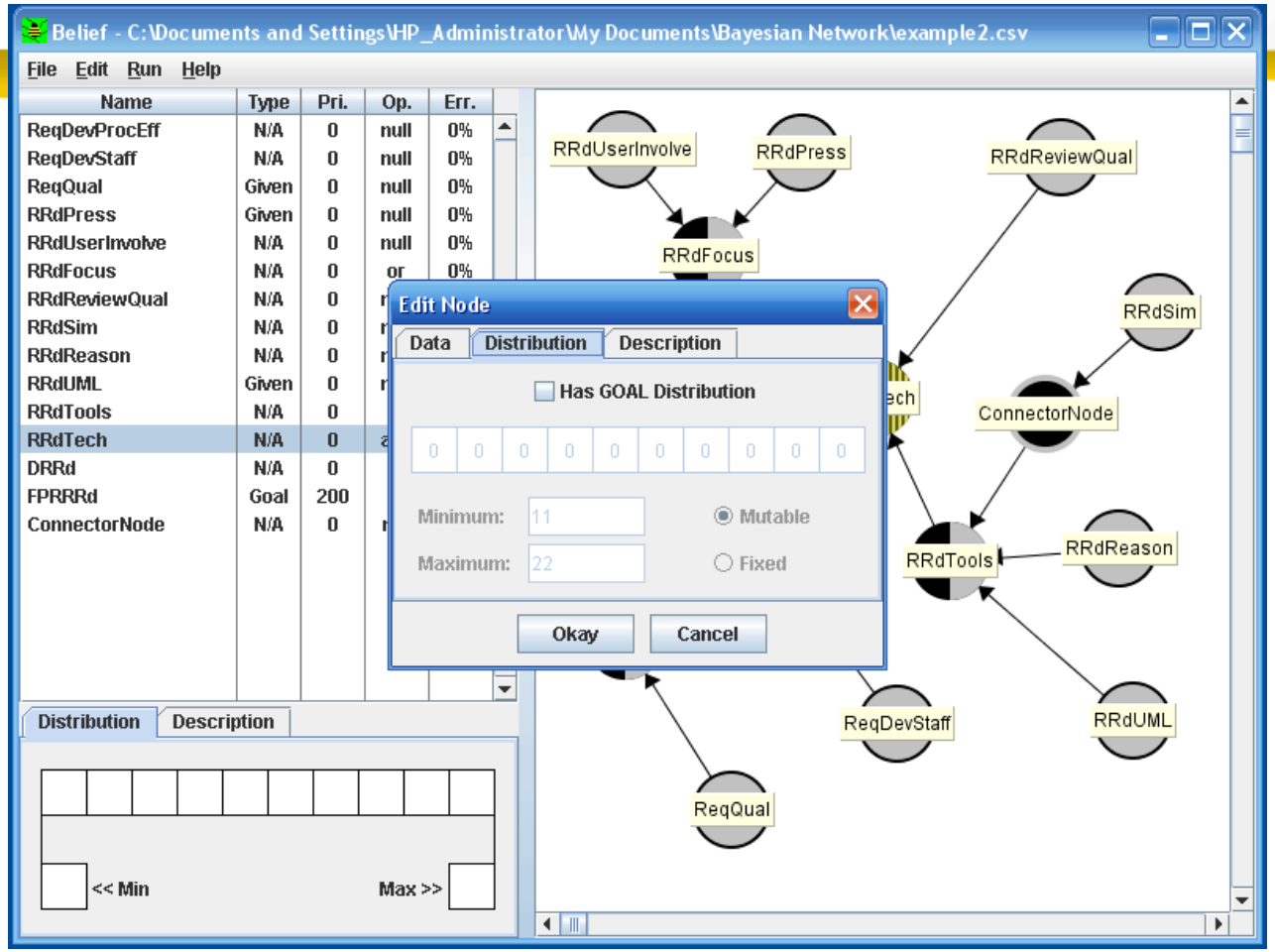
GUI component – Data display



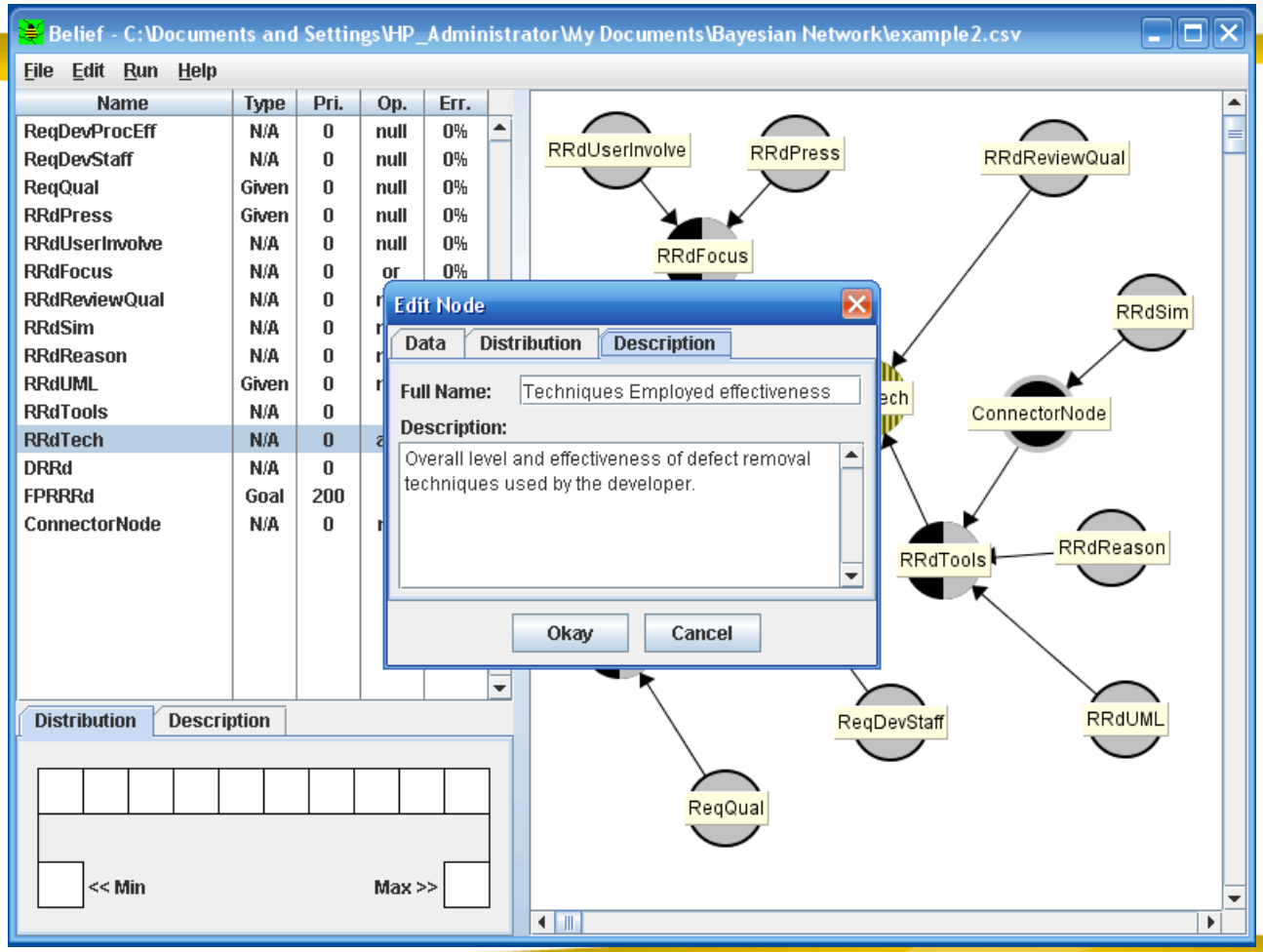
GUI component – Edit node



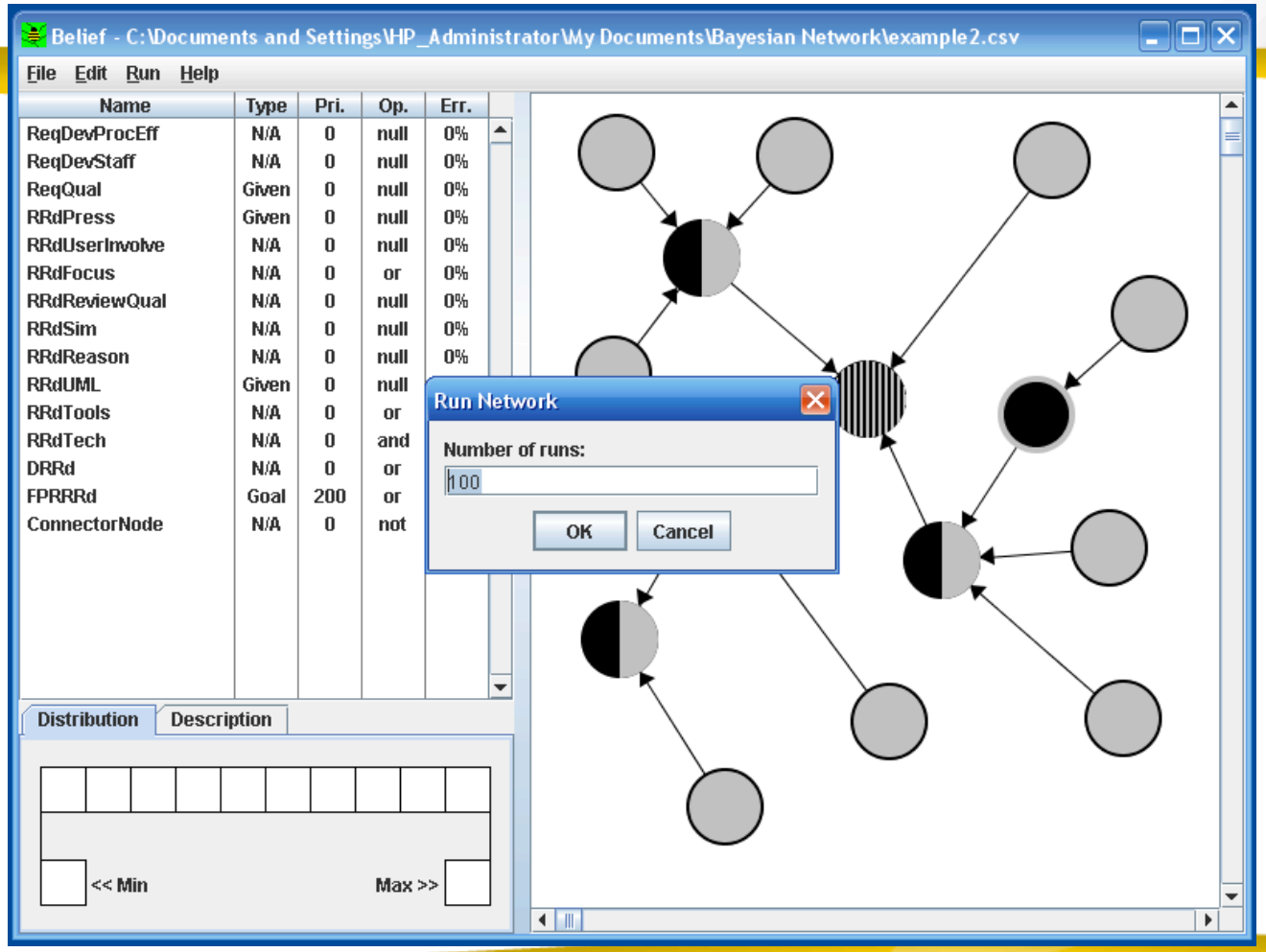
GUI component – Edit node (contd)



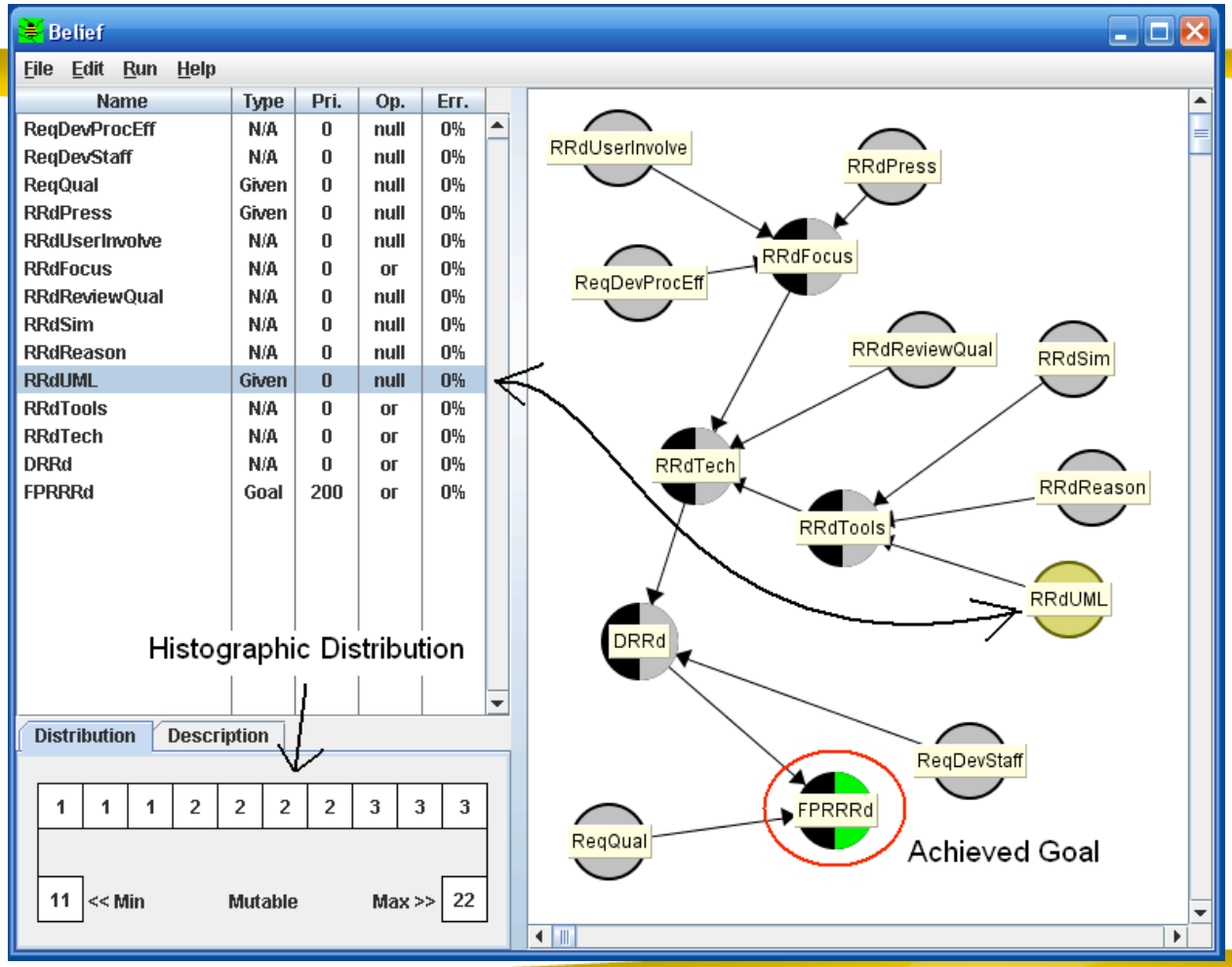
GUI component – Edit node (contd)



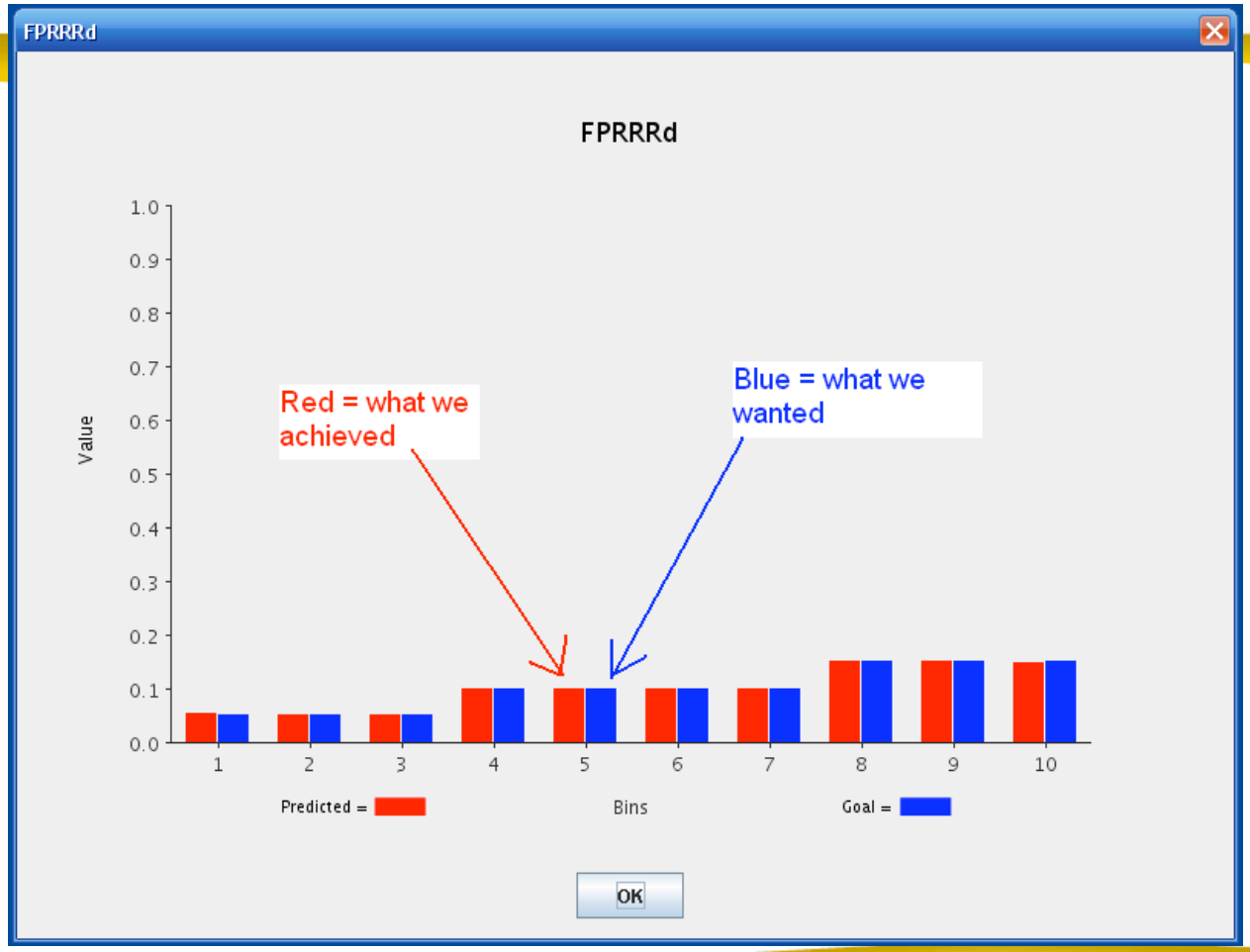
GUI component – Run network



GUI component – Network post-run

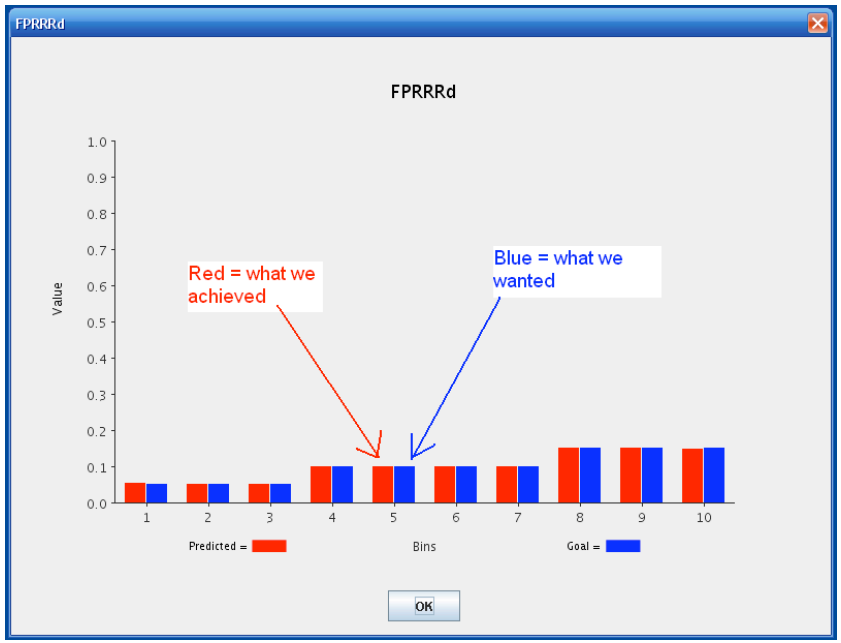


GUI component – Goal node's dists.

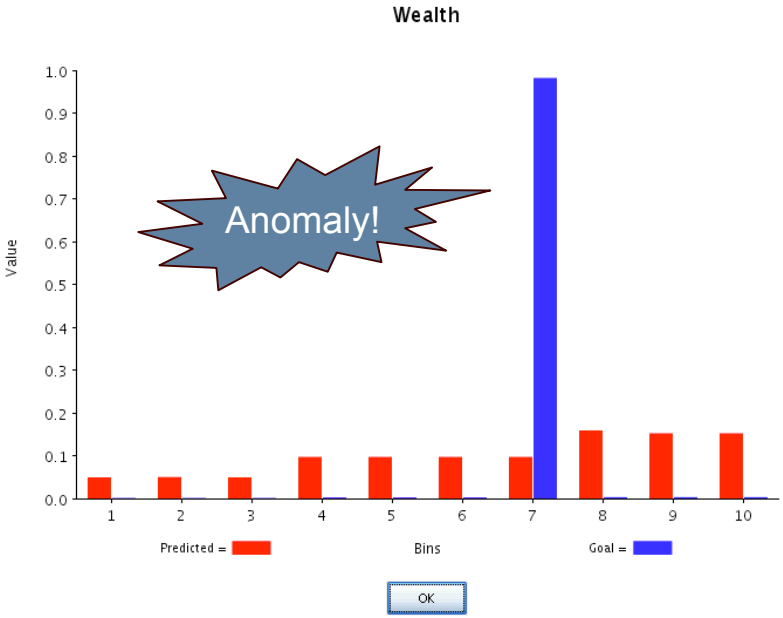


But what has this to do with anomaly detection?

Example1: you got what you want



Example2: you can't always get what you want

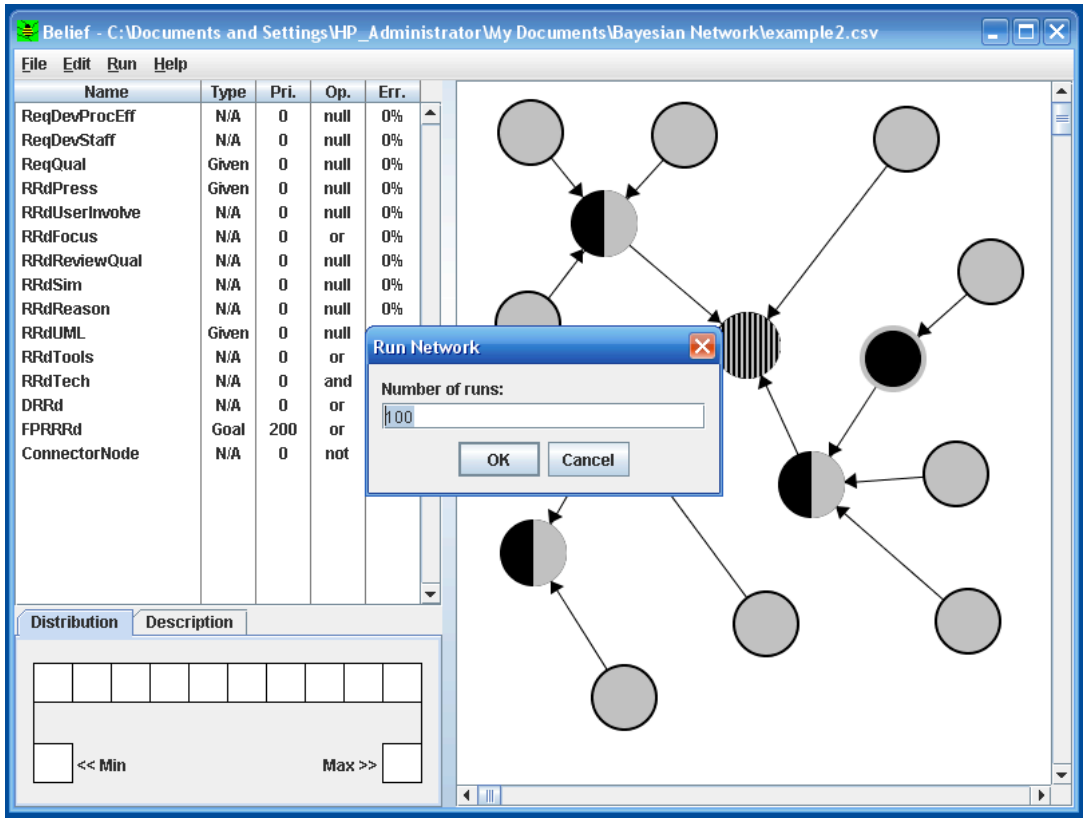


From anomaly to repair

- Anomaly detection
  - Add expectations / knowns (desires)
  - Add project data
  - Pediction: forward propergate onwards from data
  - Anomaly when not(predicted == desired)
    - I.e. this is unexpected
- Repair
  - Allow alerations to project data
  - Add in goals
  - Propagate backwards to achieve the goals
  - Repair: backwards properatin backwards from goals
  - Repair wh en not(final == initial)
    - I.e. this is what we need to change

Repair = least change

- Sort by
  - A) size of change
  - B) priority
- Try top N=1,2,3… things
  - Graph the output error as N increases
  - Return the smallest N with most effect
- An automated search.
  - Forward seelct throguht variables
  - Ordered via an intiail Bayes inference
  - Now under development



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
  - 0.6 seconds, not 58 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

Possible Future Features

- Ability to go back and redo specific runs through the network
  - Network is broken into subgraphs during processing
  - Nodes can be in multiple subgraphs, but are locked to change after the first subgraph they are seen in is processed
  - It could be possible to obtain better outcomes if the ability to go back, unlock, and re-run specific nodes exists

Possible Future Features (contd)

- The backend processing in Belief is currently done using a homebrew algorithm
- Future research could be done using different algorithms such as…
  - Simulated annealing
  - Stochastic hill climbing
  - Etc.
- …And could be translated into a feature to allow users to choose from a selection of algorithms before processing a network