



Improving IV&V Techniques Through the Analysis of Anomalies



West Virginia University.

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

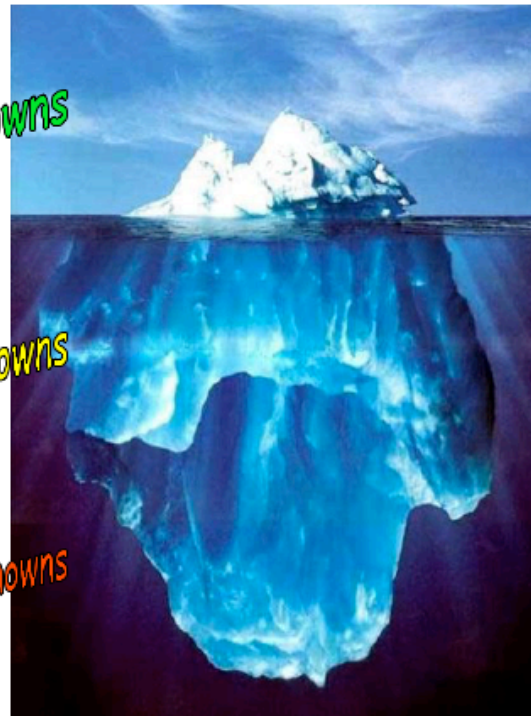
- FMEA/Hazard controls
- Close call recommendations

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

Known Knowns

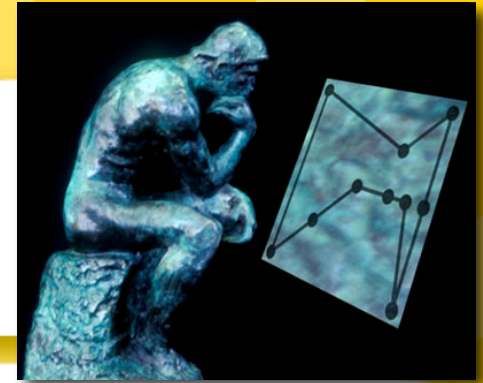
Known Unknowns

Unknown Unknowns



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 severity 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”



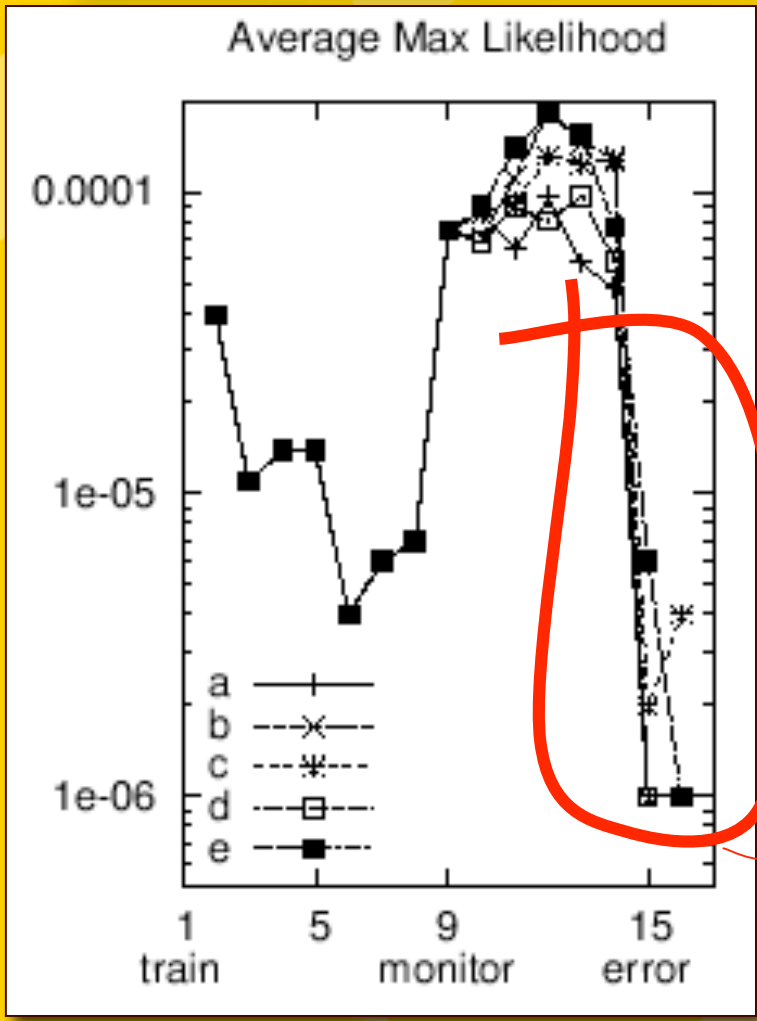
How we thought we'd do it count, alert, fix

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



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

- 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

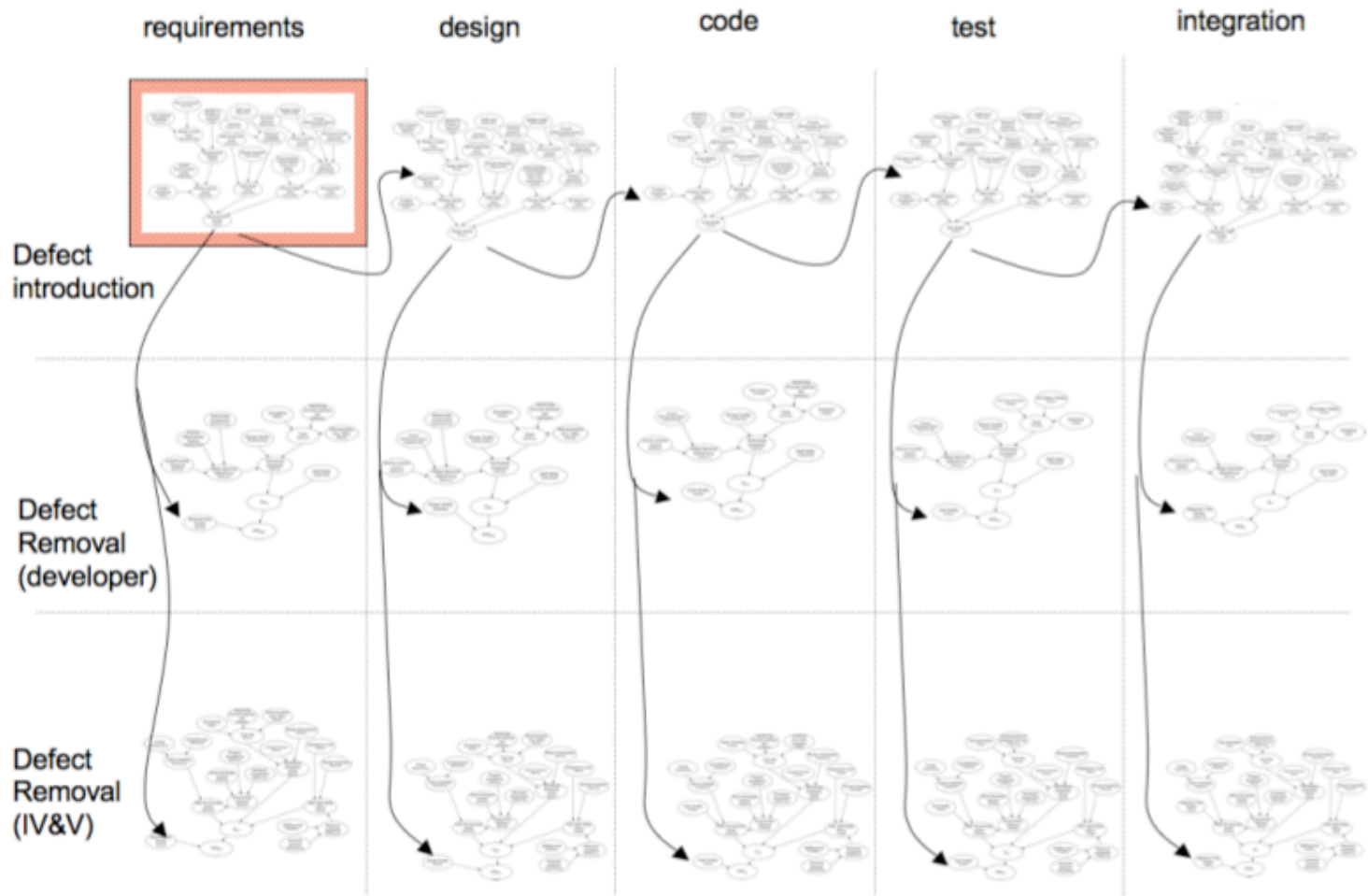


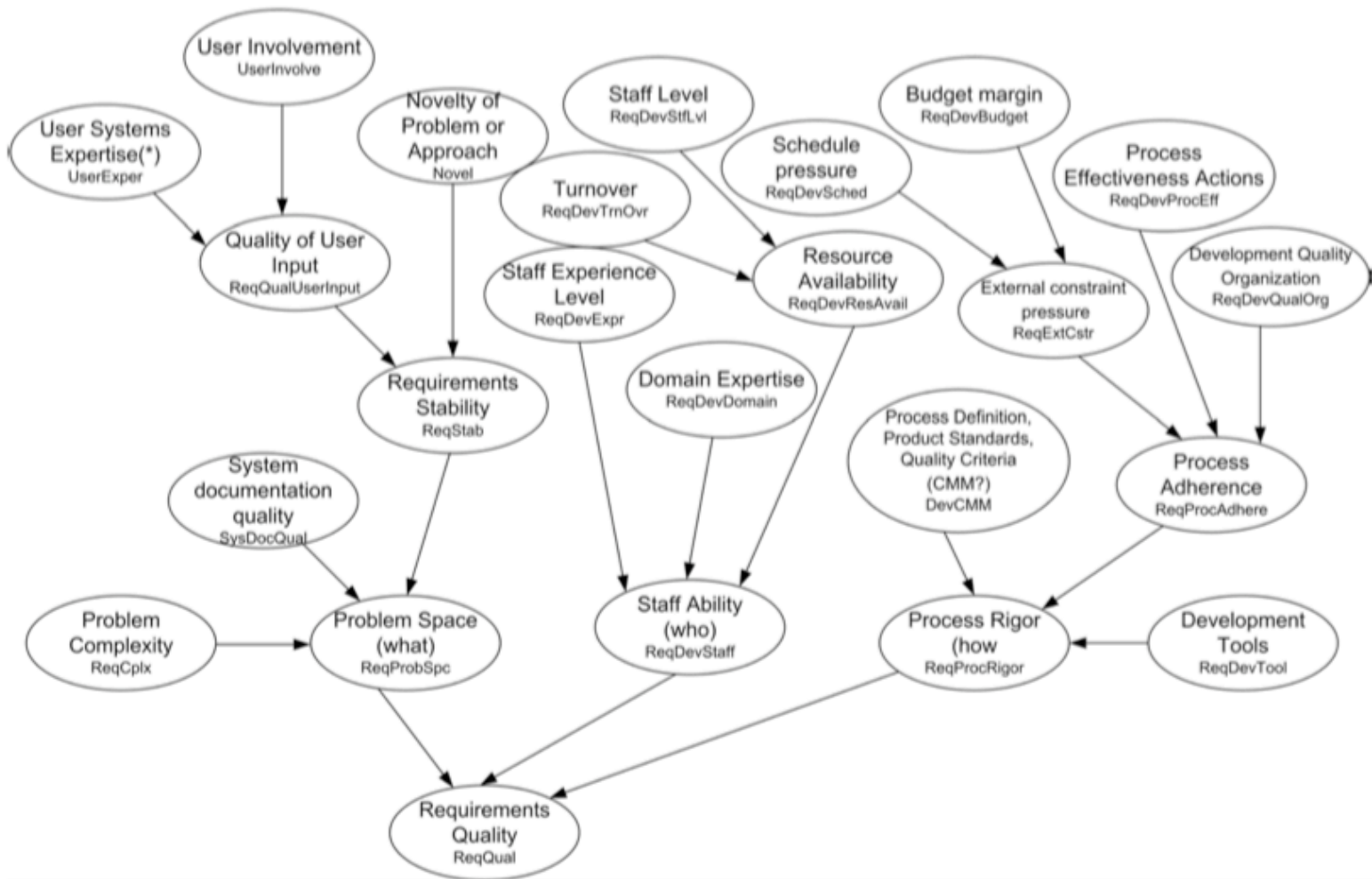
Approach (details)

- ➔ 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)

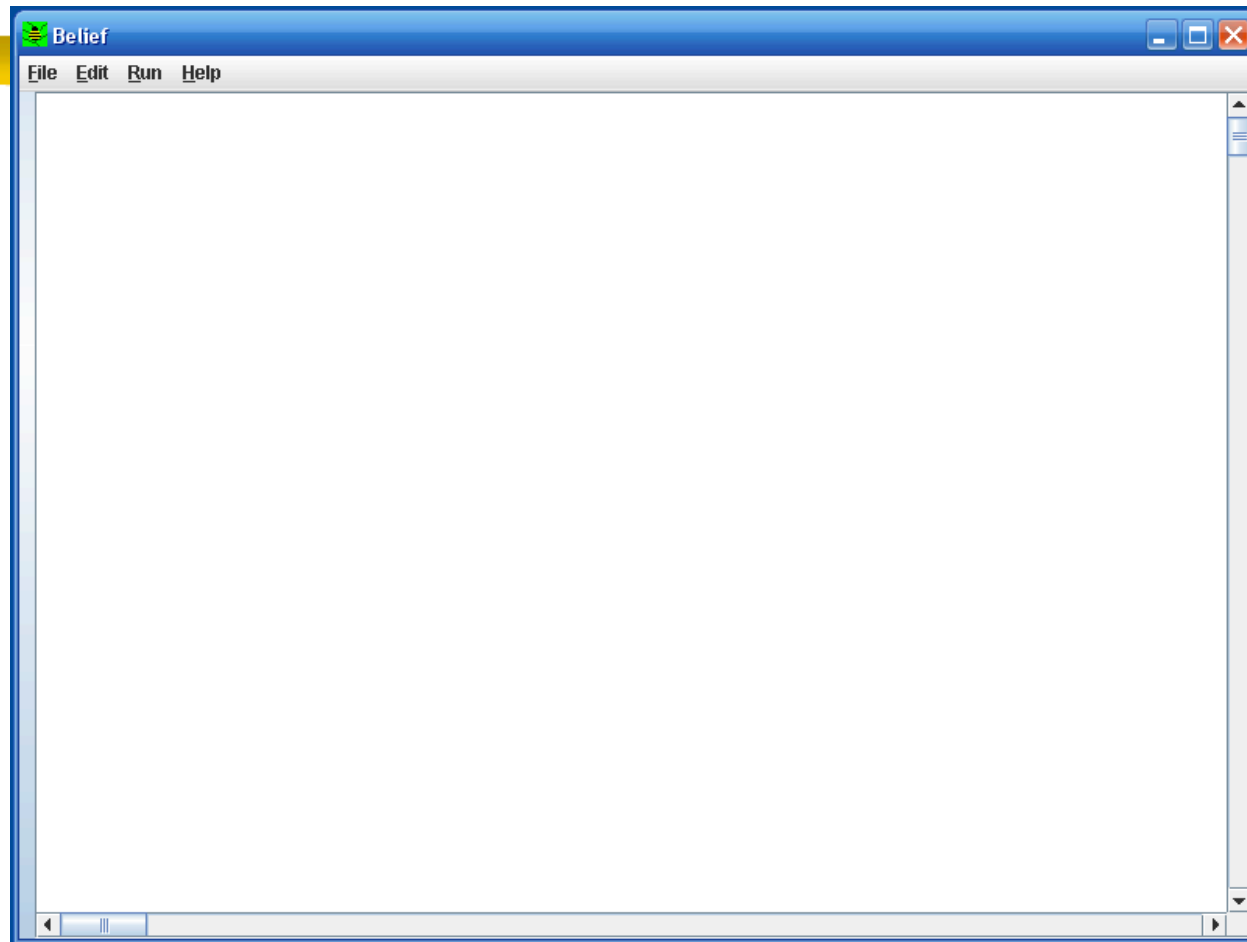


Initialize graph
with Dabney's Bayes
net descriptions
of IV&V

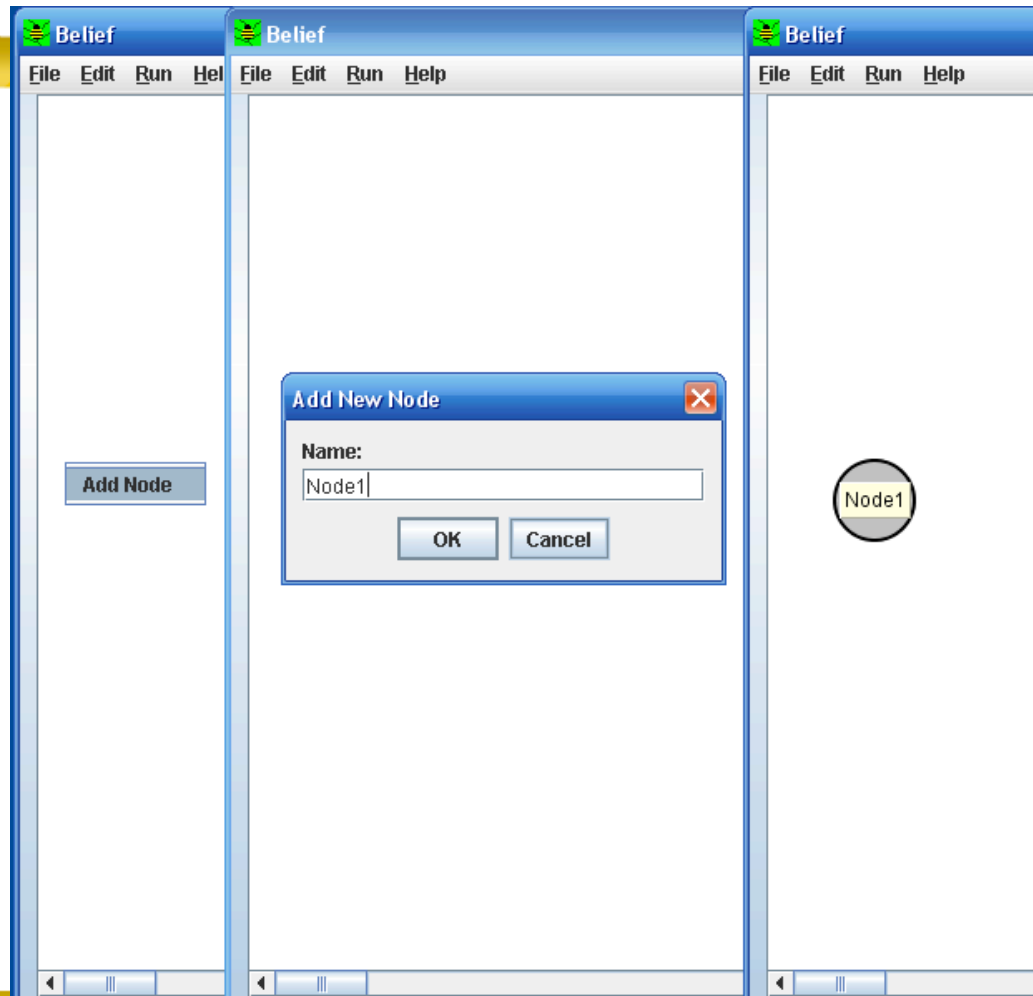




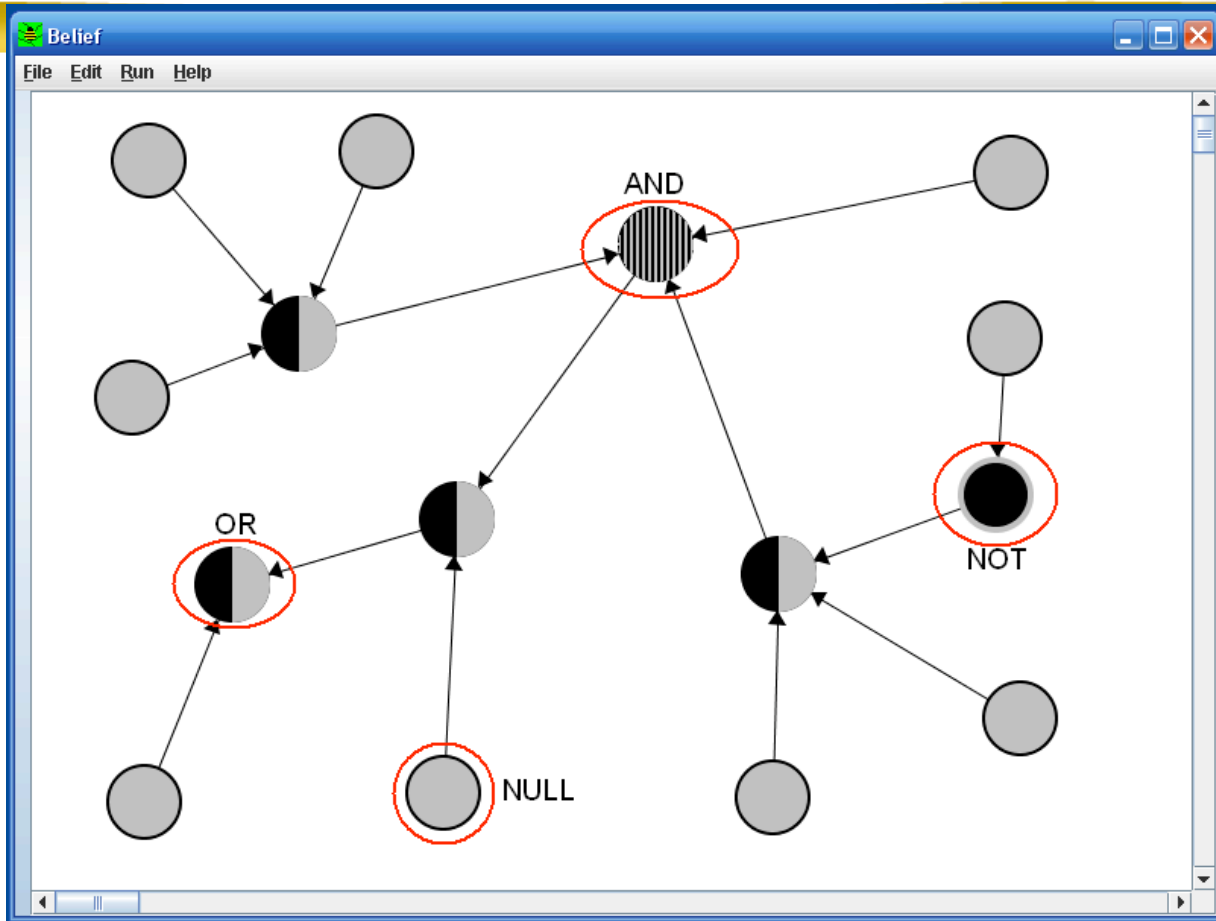
GUI component – Starting screen



GUI component – Adding a node



GUI component – Node types



GUI component – Data display

The screenshot shows the Belief GUI with a table of data on the left and a network diagram on the right. A red double-headed arrow labeled "Sliding Pane" indicates the interaction between the two views.

Name	Type	Pri.	Op.	Err.
ReqDevProcEff	N/A	0	null	0%
ReqDevStaff	N/A	0	null	0%
ReqQual	Given	0	null	0%
RRdPress	Given	0	null	0%
RRdUserInvolve	N/A	0	null	0%
RRdFocus	N/A	0	or	0%
RRdReviewQual	N/A	0	null	0%
RRdSim	N/A	0	null	0%
RRdReason	N/A	0	null	0%
RRdUML	Given	0	null	0%
RRdTools	N/A	0	or	0%
RRdTech	N/A	0	and	0%
DRRd	N/A	0	or	0%
FPRRRd	Goal	200	or	0%
ConnectorNode	N/A	0	not	0%

The network diagram on the right shows nodes representing the data items in the table, connected by arrows indicating relationships. The nodes include: RRdUserInvolve, RRdPress, RRdReviewQual, RRdFocus, ReqDevProcEff, RRdSim, RRdTech, ConnectorNode, RRdReason, RRdTools, RRdUML, DRRd, ReqQual, ReqDevStaff, and FPRRRd. The RRdTech node is highlighted with a striped pattern.

GUI component – Edit node

The screenshot displays the Belief software interface. On the left, a table lists nodes with their properties. The 'RRdTech' node is highlighted in the table. In the center, a diagram shows a Bayesian network with nodes like RRdUserInvolve, RRdPress, RRdReviewQual, RRdFocus, RRdSim, ConnectorNode, RRdReason, RRdTools, RRdUML, ReqDevStaff, and ReqQual. An 'Edit Node' dialog box is open over the 'RRdTech' node, showing its configuration.

Name	Type	Pri.	Op.	Err.
ReqDevProcEff	N/A	0	null	0%
ReqDevStaff	N/A	0	null	0%
ReqQual	Given	0	null	0%
RRdPress	Given	0	null	0%
RRdUserInvolve	N/A	0	null	0%
RRdFocus	N/A	0	or	0%
RRdReviewQual	N/A	0		
RRdSim	N/A	0		
RRdReason	N/A	0		
RRdUML	Given	0		
RRdTools	N/A	0		
RRdTech	N/A	0	and	
DRRd	N/A	0		
FPRRRd	Goal	200		
ConnectorNode	N/A	0		

Edit Node dialog box details:

- Name: RRdTech
- Priority: 0
- Operation: and

GUI component – Edit node (contd)

The screenshot displays a software interface for editing a node in a Bayesian network. The main window, titled "Belief - C:\Documents and Settings\HP_Administrator\My Documents\Bayesian Network\example2.csv", contains a table of nodes and a network diagram. The "Edit Node" dialog box is open for the "RRdTech" node, showing the "Distribution" tab with the following settings:

- Has GOAL Distribution
- Minimum: 11
- Maximum: 22
- Mutable
- Fixed

The network diagram shows nodes including RRdUserInvolve, RRdPress, RRdReviewQual, RRdFocus, RRdSim, RRdReason, RRdTools, RRdUML, ReqDevStaff, and ReqQual. The "RRdTech" node is highlighted in the diagram and in the table below.

Name	Type	Pri.	Op.	Err.
ReqDevProcEff	N/A	0	null	0%
ReqDevStaff	N/A	0	null	0%
ReqQual	Given	0	null	0%
RRdPress	Given	0	null	0%
RRdUserInvolve	N/A	0	null	0%
RRdFocus	N/A	0	or	0%
RRdReviewQual	N/A	0		
RRdSim	N/A	0		
RRdReason	N/A	0		
RRdUML	Given	0		
RRdTools	N/A	0		
RRdTech	N/A	0		
DRRd	N/A	0		
FPRRRd	Goal	200		
ConnectorNode	N/A	0		

GUI component – Edit node (contd)

The screenshot displays a Bayesian Network GUI with a table of nodes and an 'Edit Node' dialog box. The table lists various nodes with their types, priorities, and error rates. The 'Edit Node' dialog box is open for the 'RRdTech' node, showing its full name and description.

Name	Type	Pri.	Op.	Err.
ReqDevProcEff	N/A	0	null	0%
ReqDevStaff	N/A	0	null	0%
ReqQual	Given	0	null	0%
RRdPress	Given	0	null	0%
RRdUserInvolve	N/A	0	null	0%
RRdFocus	N/A	0	or	0%
RRdReviewQual	N/A	0		
RRdSim	N/A	0		
RRdReason	N/A	0		
RRdUML	Given	0		
RRdTools	N/A	0		
RRdTech	N/A	0		
DRRd	N/A	0		
FPRRRd	Goal	200		
ConnectorNode	N/A	0		

Edit Node Dialog Box:

- Full Name: Techniques Employed effectiveness
- Description: Overall level and effectiveness of defect removal techniques used by the developer.

GUI component – Run network

The screenshot displays a SAS application window titled "Belief - C:\Documents and Settings\HP_Administrator\My Documents\Bayesian Network\example2.csv". The window contains a table of variables and a graphical representation of a Bayesian network. A "Run Network" dialog box is overlaid on the network diagram, prompting for the "Number of runs" (set to 100).

Name	Type	Pri.	Op.	Err.
ReqDevProcEff	N/A	0	null	0%
ReqDevStaff	N/A	0	null	0%
ReqQual	Given	0	null	0%
RRdPress	Given	0	null	0%
RRdUserInvolve	N/A	0	null	0%
RRdFocus	N/A	0	or	0%
RRdReviewQual	N/A	0	null	0%
RRdSim	N/A	0	null	0%
RRdReason	N/A	0	null	0%
RRdUML	Given	0	null	0%
RRdTools	N/A	0	or	0%
RRdTech	N/A	0	and	0%
DRRd	N/A	0	or	0%
FPRRRd	Goal	200	or	0%
ConnectorNode	N/A	0	not	0%

The Bayesian network diagram shows a central node (hatched) with several parents (black and grey nodes) and children (grey nodes). The "Run Network" dialog box is positioned over the central node, with the "Number of runs" field set to 100. The dialog has "OK" and "Cancel" buttons.

At the bottom of the main window, there are tabs for "Distribution" and "Description". The "Description" tab is active, showing a grid of empty cells and a range selector with "Min" and "Max" labels.

GUI component – Network post-run

Belief

File Edit Run Help

Name	Type	Pri.	Op.	Err.
ReqDevProcEff	N/A	0	null	0%
ReqDevStaff	N/A	0	null	0%
ReqQual	Given	0	null	0%
RRdPress	Given	0	null	0%
RRdUserInvolve	N/A	0	null	0%
RRdFocus	N/A	0	or	0%
RRdReviewQual	N/A	0	null	0%
RRdSim	N/A	0	null	0%
RRdReason	N/A	0	null	0%
RRdUML	Given	0	null	0%
RRdTools	N/A	0	or	0%
RRdTech	N/A	0	or	0%
DRRd	N/A	0	or	0%
FPRRRd	Goal	200	or	0%

Histogrammic Distribution

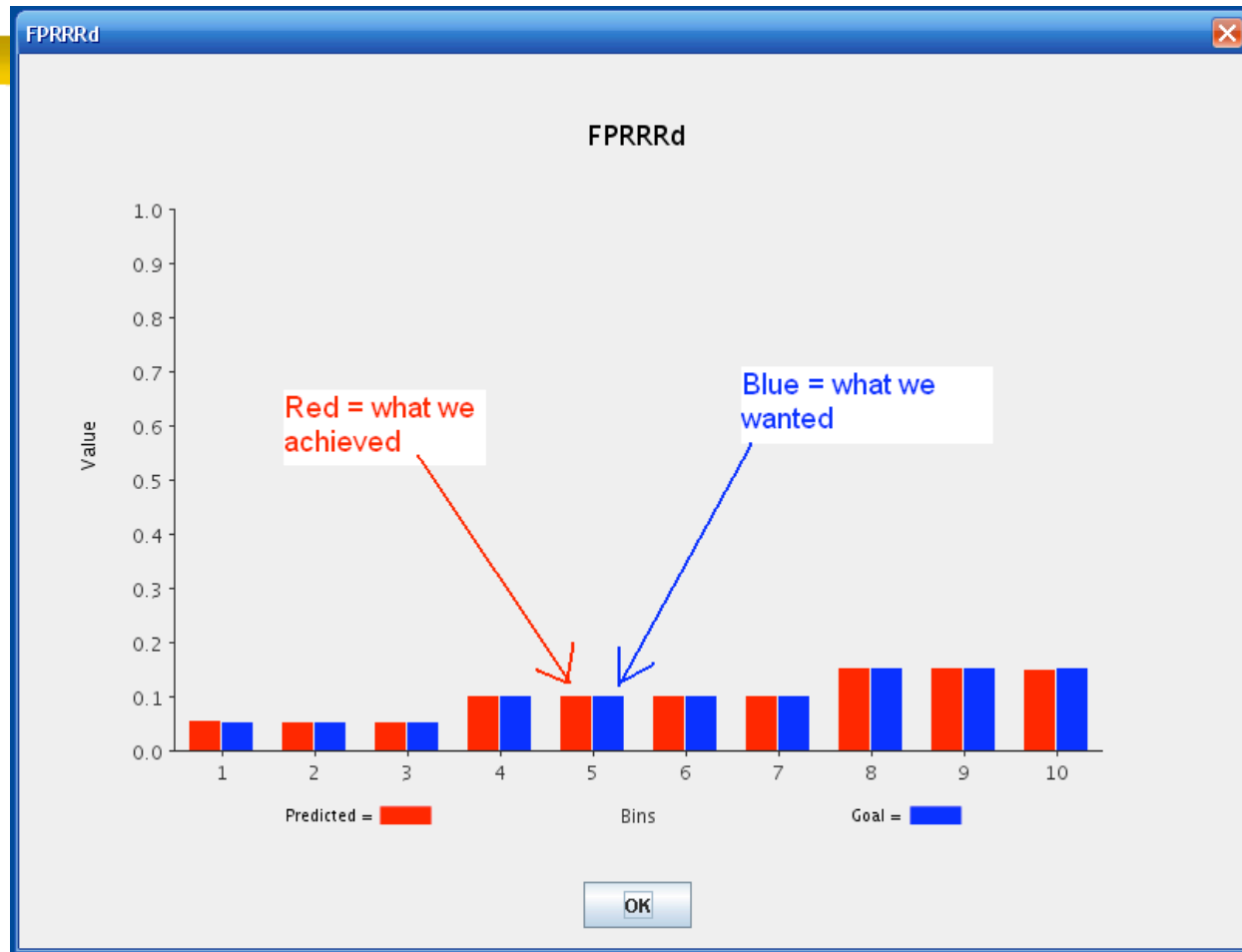
Distribution Description

1	1	1	2	2	2	2	3	3	3
---	---	---	---	---	---	---	---	---	---

11 << Min Mutable Max >> 22

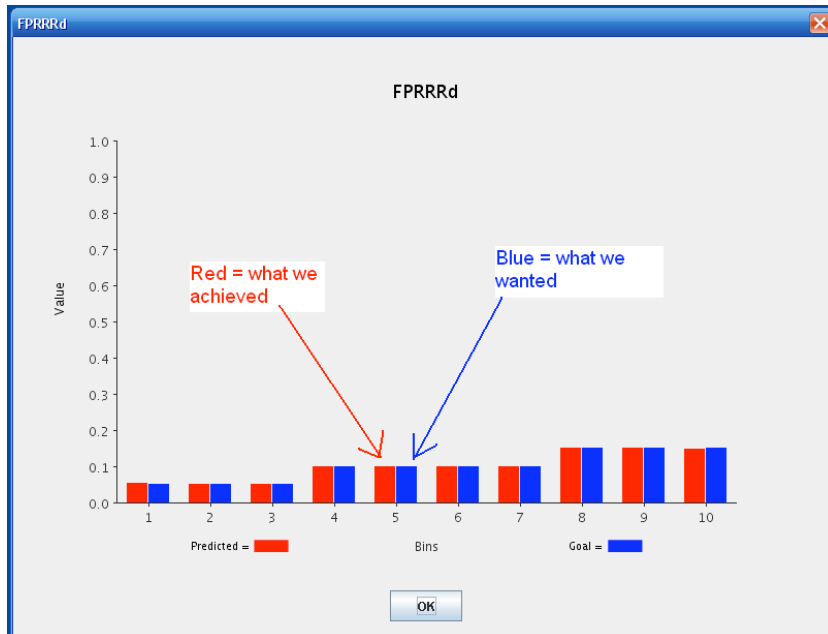
Achieved Goal

GUI component – Goal node's dists.

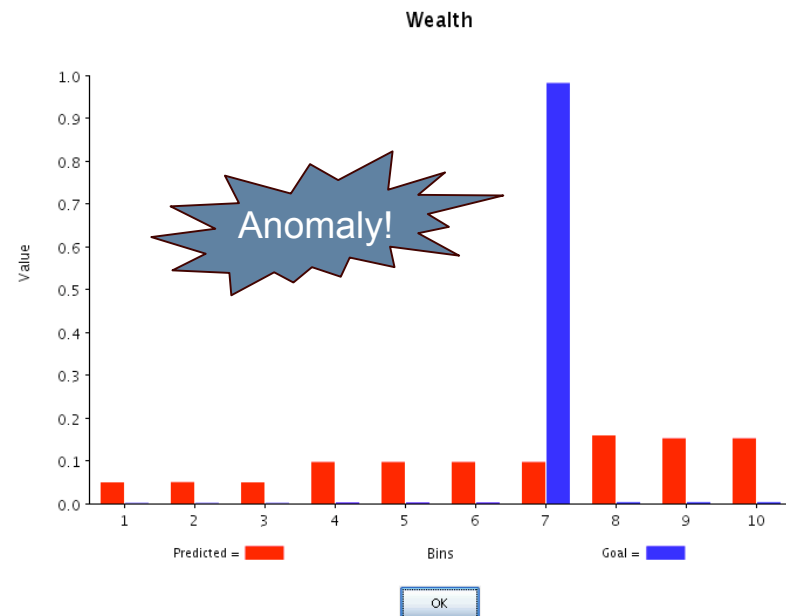


But what has this to do with anomaly detection?

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
- Prediction: forward propagate onwards from data
- Anomaly when $\text{not}(\text{predicted} == \text{desired})$
 - I.e. this is unexpected

⇒ Repair

- Allow alterations to project data
- Add in goals
- Propagate backwards to achieve the goals
- Repair: backwards propagation backwards from goals
- Repair when $\text{not}(\text{final} == \text{initial})$
 - 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 select through variables
 - Ordered via an initial Bayes inference
 - Now under development

The screenshot shows a software window titled "Belief - C:\Documents and Settings\HP_Administrator\My Documents\Bayesian Network\example2.csv". The window contains a table with columns: Name, Type, Pri., Op., and Err. Below the table is a "Run Network" dialog box with a "Number of runs:" field set to "100" and "OK" and "Cancel" buttons. The background of the window displays a Bayesian network diagram with nodes and directed edges. Some nodes are shaded or filled with patterns, indicating their current state or selection.

Name	Type	Pri.	Op.	Err.
ReqDevProcEff	N/A	0	null	0%
ReqDevStaff	N/A	0	null	0%
ReqQual	Given	0	null	0%
RRdPress	Given	0	null	0%
RRdUserInvolve	N/A	0	null	0%
RRdFocus	N/A	0	or	0%
RRdReviewQual	N/A	0	null	0%
RRdSim	N/A	0	null	0%
RRdReason	N/A	0	null	0%
RRdUML	Given	0	null	0%
RRdTools	N/A	0	or	0%
RRdTech	N/A	0	and	0%
DRRd	N/A	0	or	0%
FPRRRd	Goal	200	or	0%
ConnectorNode	N/A	0	not	0%



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