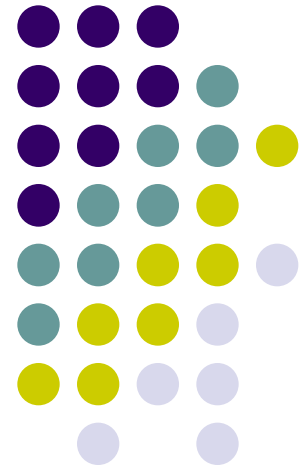
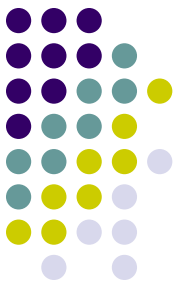


Using Belief Functions in Software Agents to Test the Strength of Application Controls: A Conceptual Framework

Robert Nehmer, Associate Professor, Oakland University

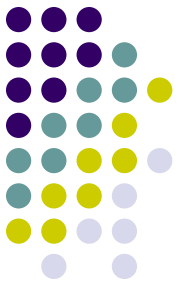
Rajendra P. Srivastava, Ernst and Young
Distinguished Professor of Accounting, and Director
Ernst & Young Center for Auditing Research and
Advanced technology ,University of Kansas





Prior Literature

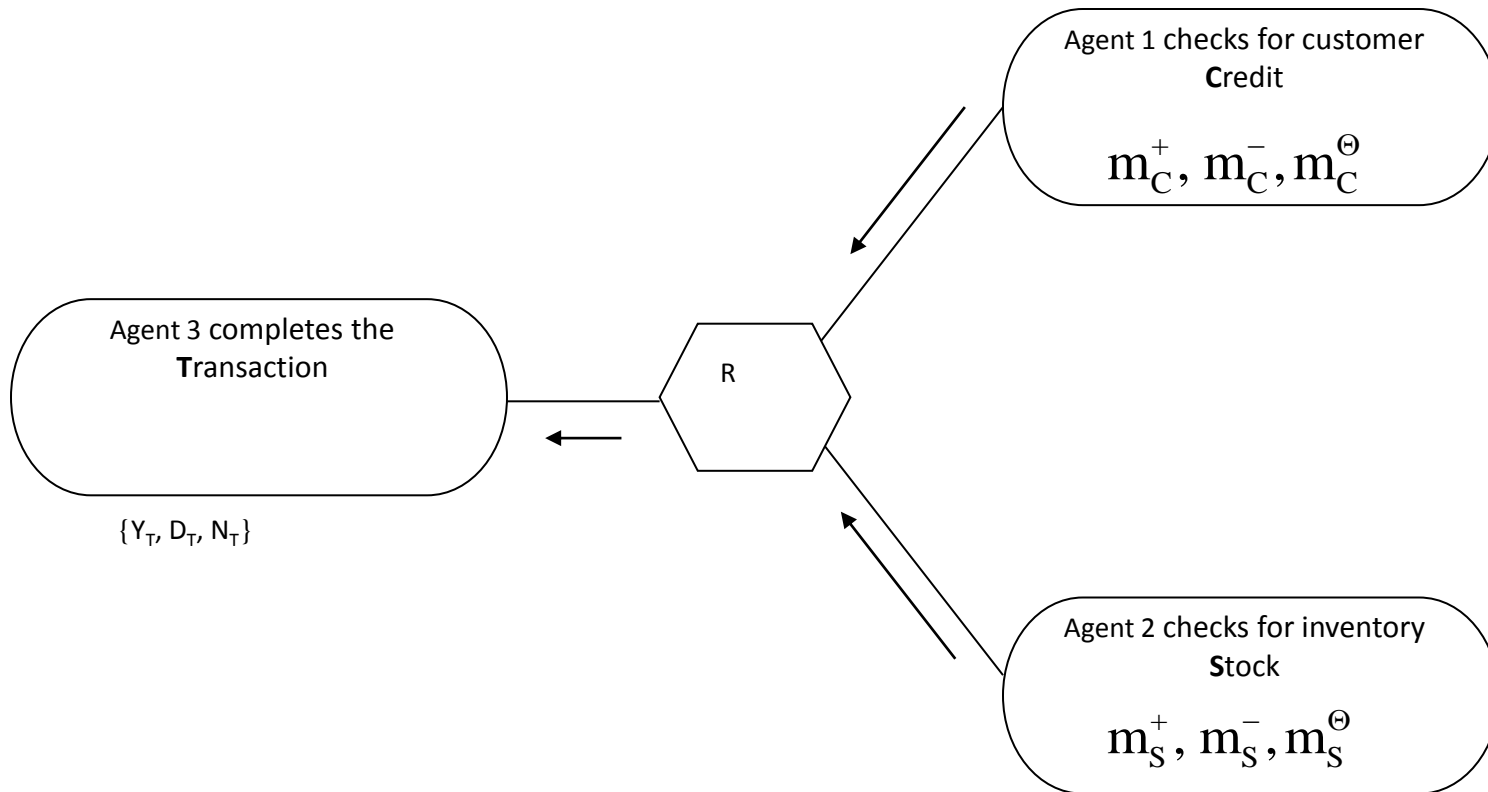
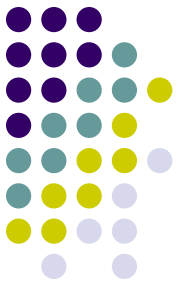
- Nelson *et al.*, 2000 - software agents in audits
- Bovee *et al.*, 2007- a framework for assessing information quality using belief functions
- Srivastava and Shafer, 1992 - uses belief functions to determine audit risk
- Srivastava, 1997 - reviews the literature on audit decisions using belief functions
- Nehmer, 2009 - a simulation of CPA firm services using software agents
- Nehmer, 2003 - provides a framework for using software agents in internal control settings

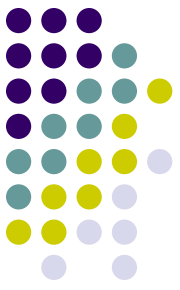


Objectives of the Research

- Extend research on automated control systems
- Investigate combining partial audit evidence in automated control environments using belief functions
- Model an real world example
- Simulate and manipulate the model

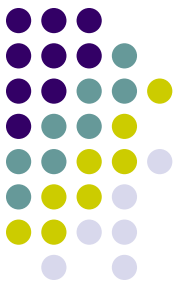
A Belief Function Model





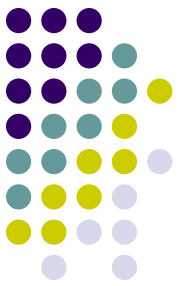
Step 1

- Agent 1 checks for customer credit: There are two possible states $\{Y_C, N_C\}$:
 - $Y_C = \text{Yes}$, the credit is approved
 - $N_C = \text{No}$, the credit is not approved
 - The belief masses pertaining to the above states are represented by the following symbols:
 - $m(\{Y_C\}) = m_C^+$, $m(\{N_C\}) = m_C^-$,
 - $m(\{Y_C, N_C\}) = m_C^\ominus$



Step 2

- Agent 2 checks whether inventory stock is available: There are two possible states $\{Y_S, N_S\}$:
- $Y_S = \text{Yes}$, the item is available
- $N_S = \text{No}$, the item is not available
- The belief masses pertaining to the above states are represented by the following symbols:
- $m(\{Y_S\}) = m_S^+$, $m(\{N_S\}) = m_S^-$, $m(\{Y_S, N_S\}) = m_S^\ominus$



Step 3

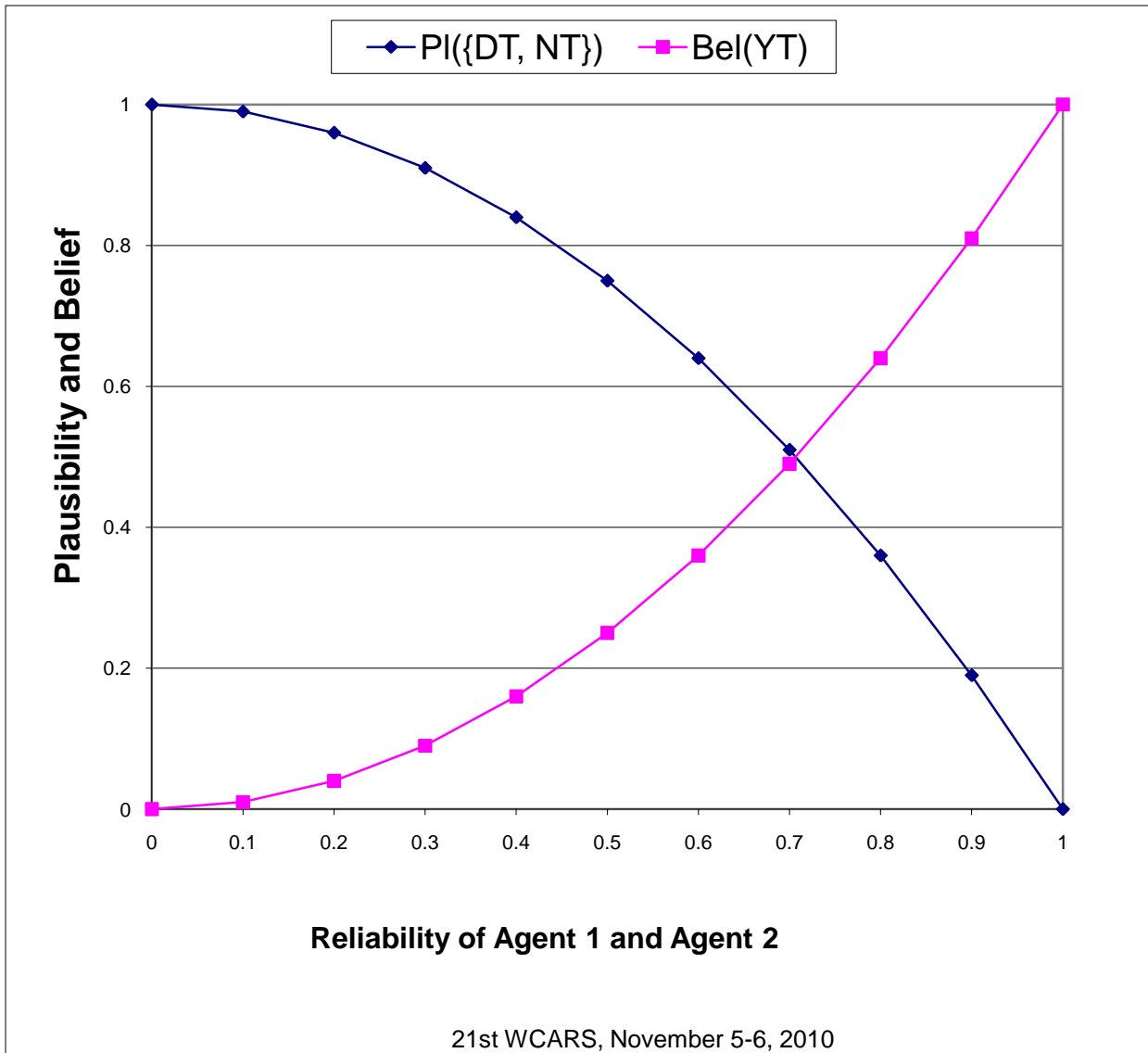
- Agent 3 completes the transaction, i.e., the agent sends the signal to shipping department for shipment of the item. There are three possible states $\{Y_T, D_T, N_T\}$:
- $Y_T = \text{Yes}$, Complete the transaction, i.e., ship the item
- $D_T = \text{Delay}$ the shipment, i.e., inform the customer that item will be shipped at a later date (item is on the back order)
- $N_T = \text{No}$, do not ship the item

Agent 3's Actions

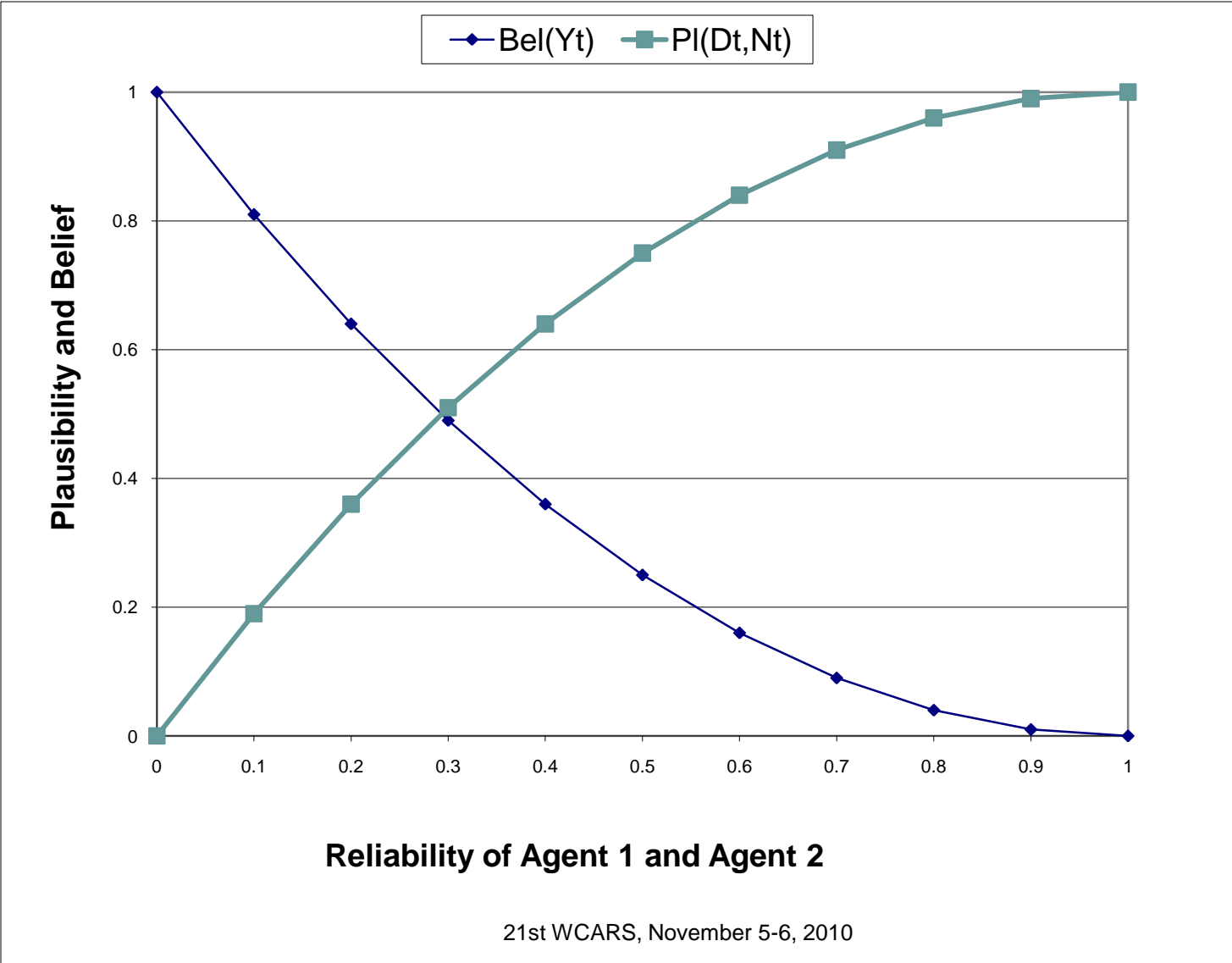
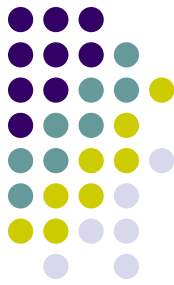


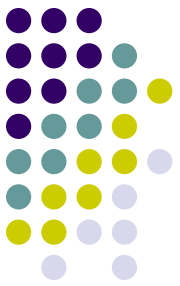
- Agent 3 acts upon the information received from Agent 1 and Agent 2. The following actions of Agent 3 are defined based on the various possible states of Agent 1 and Agent 2.
- $Y_T = Y_C Y_S$ (Yes complete the transaction, i.e., ship the item since credit is approved and Yes, the item is available)
- $D_T = Y_C N_S$ (Delay the shipment of item, credit is approved but the item is not in the stock)
- $N_T = \{N_C Y_S, N_C N_S\}$ (Do not complete the transaction, i.e., do not ship the item because the credit is not approved whether the item is available or not)

Agents Communicate without Noise



Agents Communicate Bad Credit as Good Credit

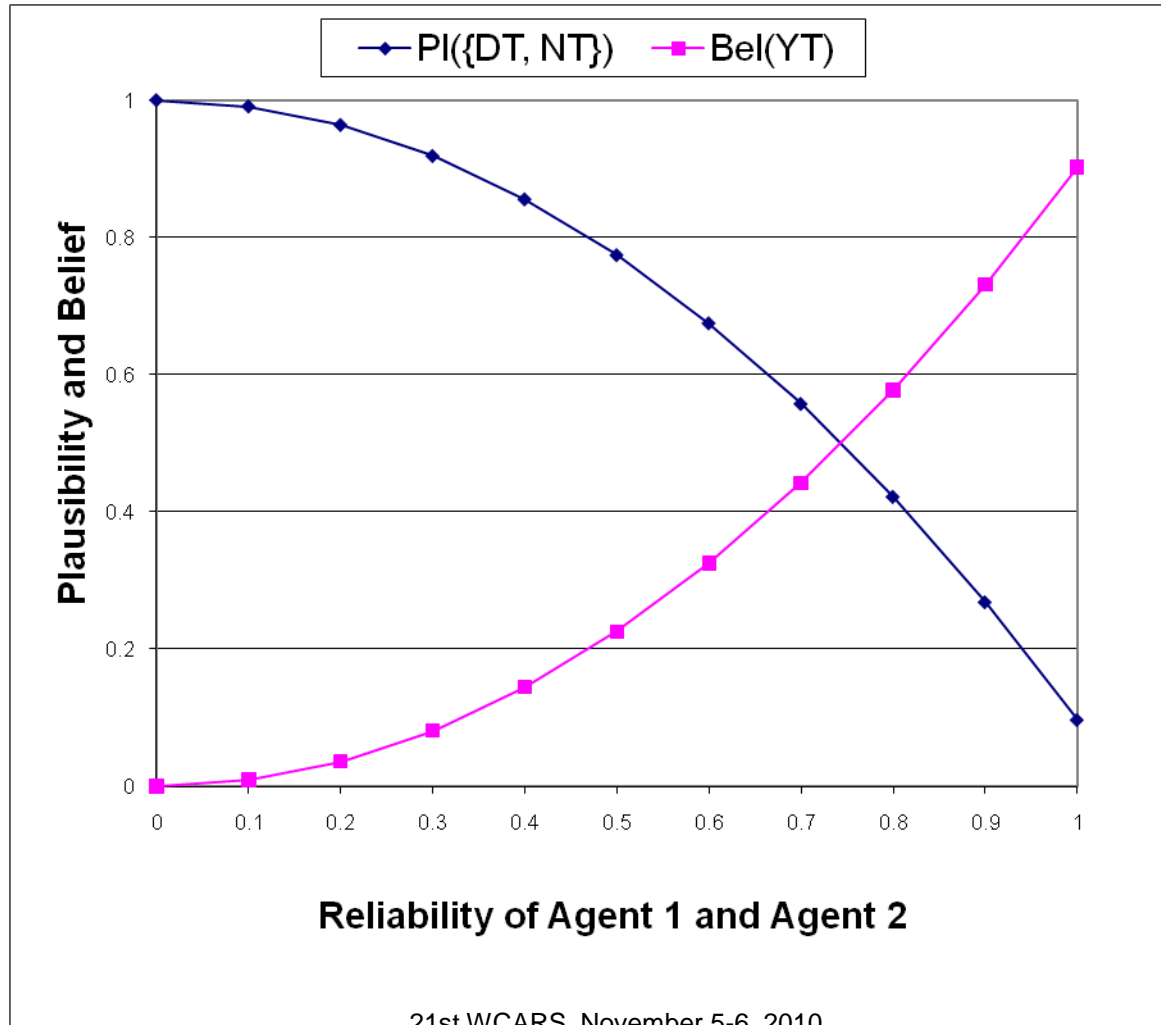




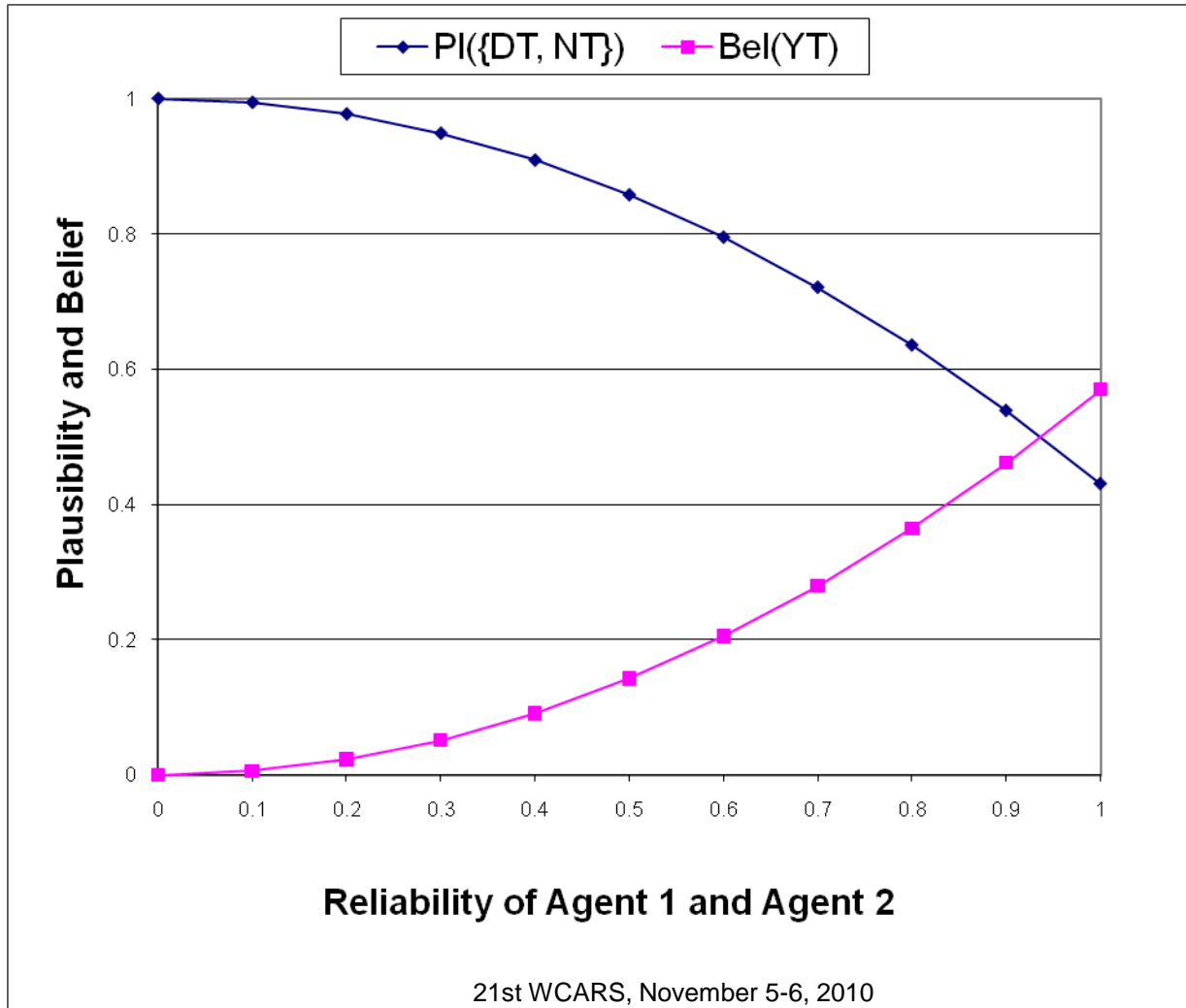
Other Issues

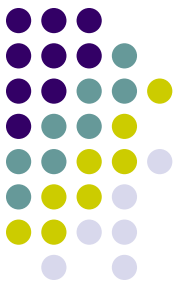
- Management control unites with internal controls through automation
- Agents fed by clusters of control agents
- Compensating controls – affect on Agent 3's decisions
- Agent 3 checking on the veracity of Agents 1 and 2 signals (next slides)

Agent 1 and 2 at 95% Confidence



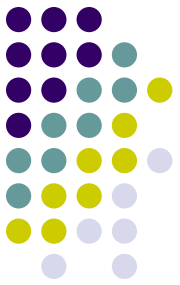
Agent 1 begins to fail and falls to 60% Confidence





Conclusions

- Software agents can provide a powerful way of implementing automated systems of internal control
- Belief functions allow the combination of disparate audit evidence without the “diminishing to zero” problems of classical probability theory
- Adding intelligent processing of agent inputs allows the system to detect failures more quickly.



Questions?

Raj Srivastava

rsrivastava@ku.edu

Rob Nehmer

nehmer@oakland.edu