

# Evaluating Centralized, Hierarchical, and Networked Architectures for Rule Systems

**Benjamin Craig**

University of New Brunswick  
Faculty of Computer Science  
Fredericton, NB, Canada

Senior Technical Report Presentation  
November 20, 2008

# Outline

- Defining Terminology
  - Rules, Distributed Systems, Topologies, OO jDREW, Rule Responder
- Topologies for distributed Architectures
  - Star Topology
    - Advantages and Disadvantages
  - P2P Topology
    - Advantages
- Knowledge Maintenance for Rule Systems
  - Knowledge Organization
  - Knowledge Maintenance
- Conclusion

# What is a rule?

- Fact (POSL format):

spending(Peter Miller, min 5000 euro, last year).

- Rule (POSL format):

premium(?Customer) :-

spending(?Customer, min 5000 euro, last year).

- A deductive rule engine would determine that Peter Miller is a premium customer

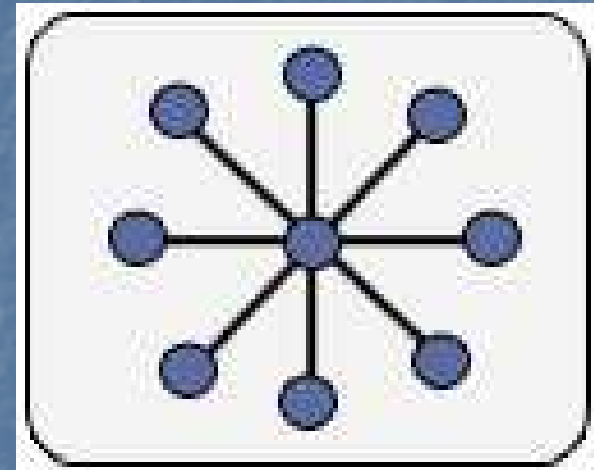


# Distributed Systems

- A distributed system is a set of computer processes that appear to the user as a single system
- The distributed system must coordinate all of these processes
- Distributed systems are implemented using middleware that creates a topology

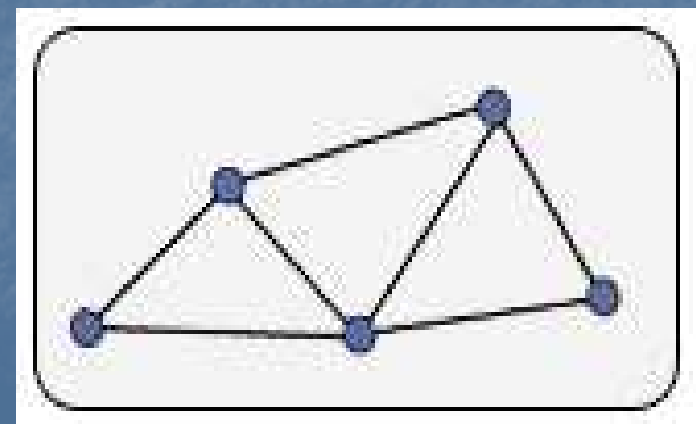
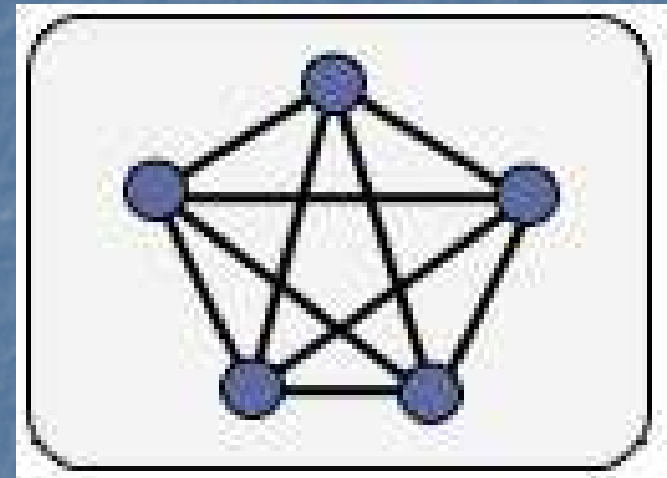
# Hierarchical - Star Topology

- Single level hierarchy
- Connects all spokes with a centralized hub
- All information must be sent through the hub to the spokes



# Networked (P2P-Like Architecture)

- Fully connected network
  - Connects all nodes together with a direct connection
  - Full mesh topology
- Partially connected network
  - Only a subset of nodes are connected together
  - Partial mesh topology



# OO jDREW

- Centralized Rule System
- Object Oriented Java Deductive Reasoning Engine for the Web – extensions of jDREW
- Supports rules in two formats:
  - POSL: Positional Slotted presentation syntax
  - RuleML: XML interchange syntax  
(can be generated from POSL)



# Rule Responder

- Distributed Rule System
- Is currently implemented as a hierarchical rule system
- Rule Responder is a prototypical multi-agent system for virtual communities
- Supports rule-based collaboration between the distributed members of community
- Members are assisted by semi-automated **rule-based** agents, which use rules to describe the decision and behavioral logic



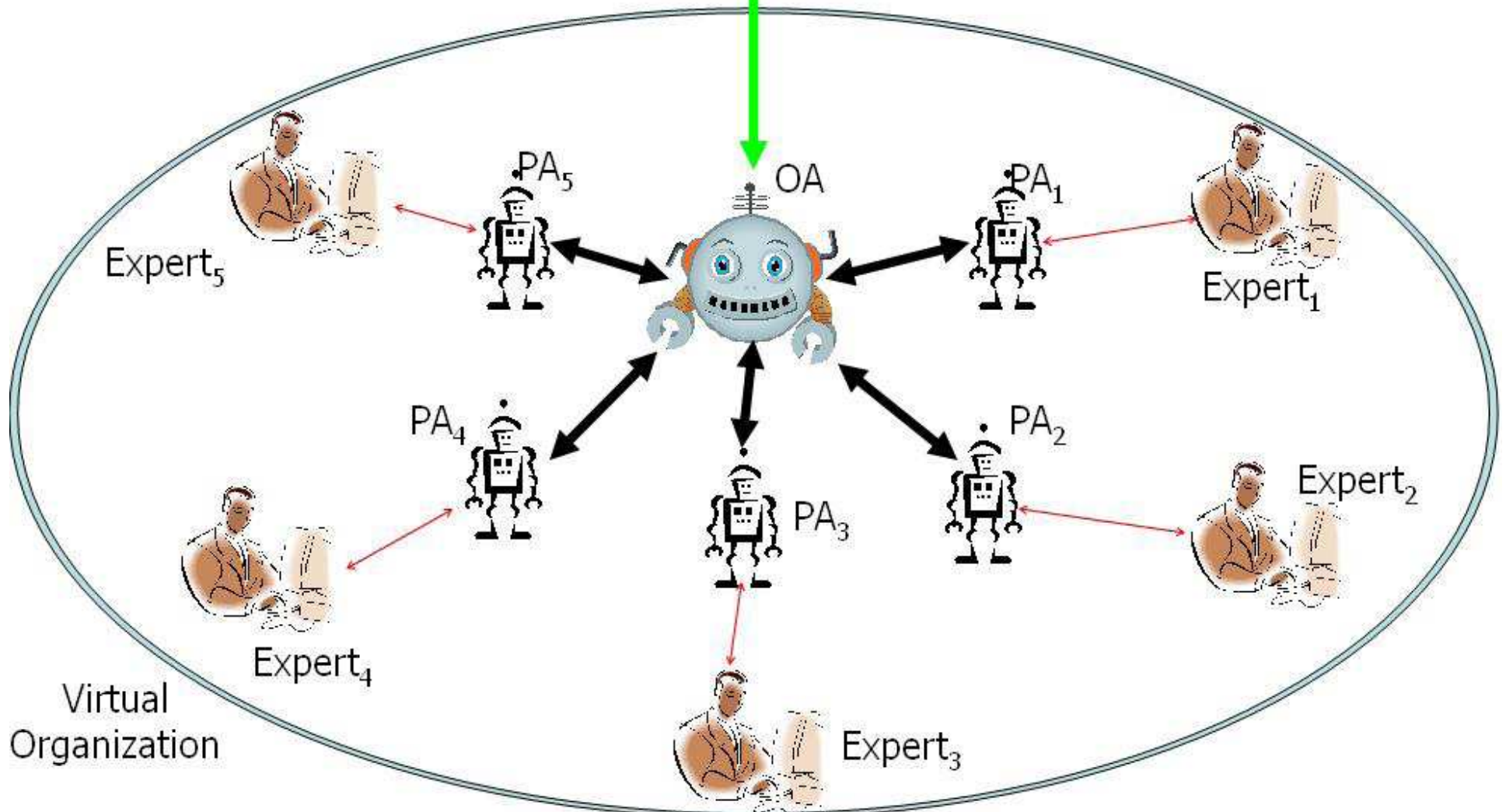
# Star-Like Rule Responder Architecture



EA: External Agent

Hub — OA: Organizational Agent

Spoke — PA: Personal Agent



# Topology Performance

- When building a distributed system a topology is required
- Distributed topologies all have communication overhead that centralized systems do not have
- A key design goal for distributed systems is to minimize this communication overhead

# Star Advantages

- Isolation of spokes from other spokes
  - If one spoke fails then it does not affect others
- Adding and remove spokes to the hub is trivial
- Inspection of all traffic through the topology
  - Improved Security
- Trouble shooting is easy
- Easy to understand and implement



# Star Disadvantages

- Scalability, reliability and performance of the star topology rely on the hub
- If the hub fails then the entire system fails
- The hub can become over loaded and the system will encounter slow downs
- To prevent the bottle neck of the star topology a P2P topology can be used

# P2P Advantages

- Does encounter bottleneck performance issues like the star topology
- Whenever a node is added the total bandwidth capacity is increased
- When a node fails the system will be able to recover
  - A peer can act in place of another peer

# Knowledge Maintenance for Rule Systems

- A distributed system has many different knowledge bases distributed across the system
  - Many files and databases
  - Each knowledge base is known as a module
- A centralized system has all of the knowledge stored in a single location
  - Either a file or a database



# Knowledge Organization - I

- When deciding how to group modules one of two ways can be used
  - Predicate Centric
    - All predicates are stored in one module
  - Person Centric
    - All knowledge about one person or thing is stored in a module
    - Rule Responder uses person centric organization
- *example on next slide*

# Knowledge Organization - II

## ■ Person Centric:

- phoneOf(ben, 1-506-270-3403)
- emailOf(ben, ben.craig@unb.ca)
- phoneOf(jim, 1-506-275-9712)
- emailOf(jim, jim.lorde@unb.ca)

## ■ Predicate Centric:

- phoneOf(ben, 1-506-270-3403)
- phoneOf(jim, 1-506-275-9712)
- emailOf(ben, ben.craig@unb.ca)
- emailOf(jim, jim.lorde@unb.ca)

# Module Boundaries

- When querying modules sometimes information from multiple modules is required
- Example Query
  - “What are the phone numbers of everyone in the organization?”
  - This query must backtrack across multiple modules when using person centric storage



# Centralize Maintenance

- All knowledge is stored in a single format
  - No translation steps when using a rule engine to execute the rules and facts
- All knowledge is stored in a single location
  - Updating knowledge is simple
  - Do not need to worry about inconsistencies in the knowledge

# Distributed Maintenance

- Knowledge is stored in many formats
  - Translation steps are required when sending a query from one rule engine to another
  - An interchange language is required
- Knowledge is stored in a many locations
  - Each agent can update their own knowledge
  - Knowledge bases could be incomplete or inconsistent
  - Consistency rules can be used to test if the knowledge is complete and consistent

# Benchmarking Use Case



- RuleML-20xy Symposia
  - An organizational agent acts as the single point of entry to **assist** with symposium planning:
    - Currently, query answering about the symposium
    - Ultimately, preparing and running the symposium
  - Personal agents have supported symposium chairs since 2007 (deployed as Q&A in 2008)
    - General Chair, Program Chair, Panel Chair, Publicity Chair, etc.



# OO jDREW (centralized) Benchmarking

Query:	Computation Time (ms):
1)	141
2)	31
3)	22
4)	18
5)	16

- Results show that a centralized system does not take much computation time

# Rule Responder (Hierarchical) Benchmarking

- Same 5 queries used as in the OO jDREW benchmarking

Query:	Computation Time (ms):
1)	3430
2)	4861
3)	4057
4)	9048
5)	2780

- The variation in the computation time comes from the number of answers to the queries
- Computation time is much higher from the communication overhead of a distributed system

# Theoretical Networked Performance

- Speed ups can be obtained using a P2P topology
- Instead of having all communication going through the hub - direct communication from one spoke to another could be used
  - Will reduce the amount of communication steps in the distributed system
- The bottleneck issue of a hierarchical system does not show exists in a networked system



# Conclusion

- A rule system can be either distributed or centralized
- When using a distributed system a topology must be used
- The advantages and disadvantages of the topology must be evaluated when deciding which topology to use in a distributed rule system
- Distributed Knowledge maintenance must be addressed when building a distributed rule system
- The Benchmarking done shows the difference in computation time between a Centralized and a distributed Hierarchal system

