Lecture 26: RBR, MAS and Summary

Victor R. Lesser

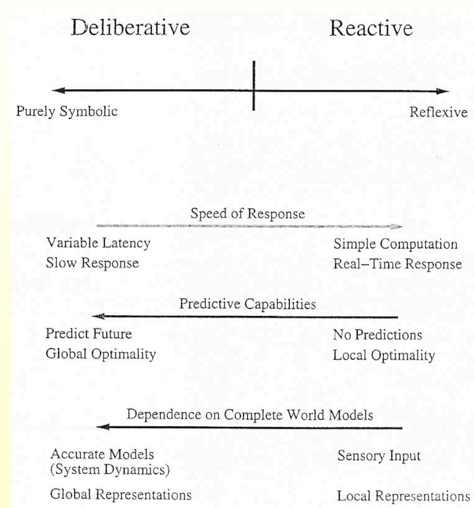
CMPSCI 683
Fall 2004

Today's Lecture

- More on Resoure Bounded Reasoning
- Something on Multi-Agent Systems
- Review of Course

Building a Resource Bounded Agent Architecture

Systems Spectrum

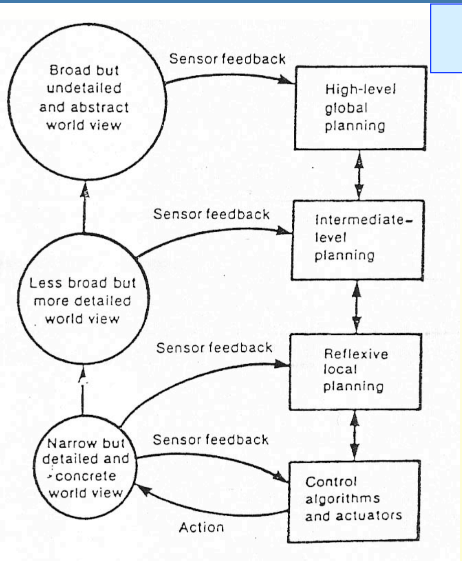


Hierarchical / Deliberative Systems

- Top-down process
- Communication and control in a predictable and predetermined manner
- Production orientation for structured environments
- Higher levels establish subgoals for lower levels
- Clear division of planning functionality typically including:
  - Mission planner
    - Establishes objectives
  - Navigator
    - Develops global path subject to mission planner's constraints
  - Pilot
    - Implements global plan piecewise
    - Handles avoidance of observed obstacles

Example Hierarchical System

Armored Personnel Carrier
Parodi and Nitao (FMC Corp)



(from Parodi 1985)

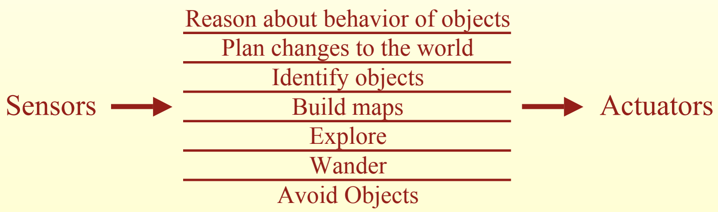
General Characteristics of Reactive Control

- It is typically manifested by a decomposition into primitive behaviors
- Global representations are generally avoided.
- Sensor decoupling is preferred over sensor fusion.
- It is well situated for dynamic changes in the world.

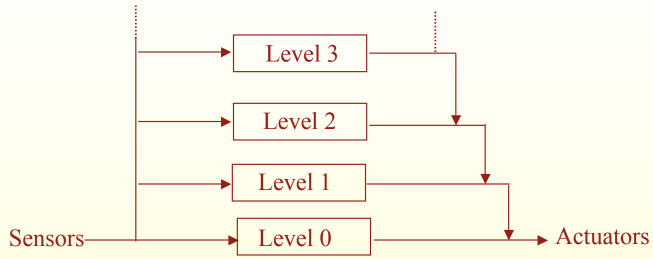
Decomposition of Mobile Robot Control

Traditional Decomposition

Behavioral Decomposition



Subsumption Architecture (Brooks 1986)



- **Design Criteria**
  - Incremental design from lower to higher levels of competence
  - Each level represents a complete control system
  - No global memory
  - Lower levels remain unchanged and unaware of higher levels
  - Higher levels can subsume lower ones

Integrating Reflective/Deliberative and Reactive Problem Solving

- **Reactive requires immediate response**
  - Little integration of data
  - no search
- **Reflective/Deliberative (cognitive) can spend time deliberating**
  - A lot of data can be integrated
  - Extensive search
  - But still needs to be controlled in terms of resources used, deadlines,…

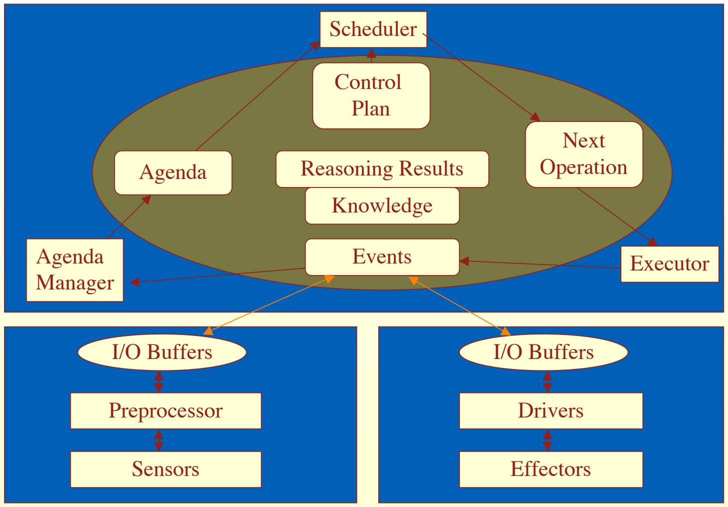
An Example Domain: The ICU

- **Designed to monitor post-op patients on life-support equipment**
- **Time-critical problems at many scales**
- **Many types of reasoning are applicable.**

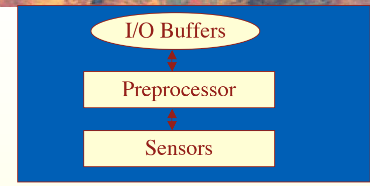
Common Features

- **Real-time constraints**
- **Data glut**
- **Diverse events**
- **Impossible to enumerate state space**
- **Environment somewhat predictable**
- **Interactions between actions**
- **Diverse demands**
- **Variable stress**

BB1 Architecture



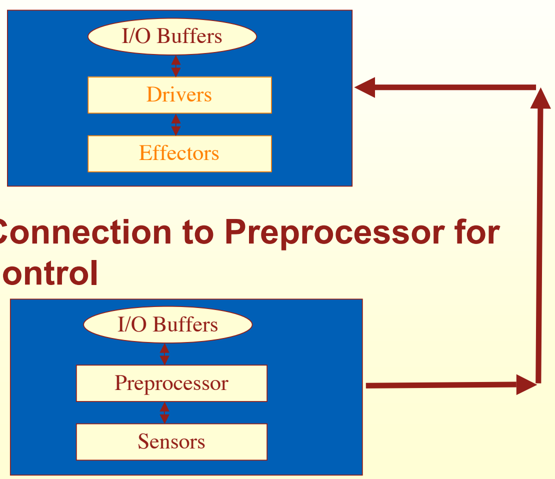
Architectural Features of BB1 Perceptual Subsystem



- **Limited I/O buffer**
  - Best-first retrieval, worst-first overflow
  - Item's relevance to current reasoning
  - Recency
  - Urgency of processing items
- **Controllable preprocessor**
  - Abstraction
    - Compresses data, running averages, patterns across mulitiple values
  - Filtering
    - Changes in critical data (p%)
    - Send new values at least seconds
  - Annotation
    - Define relevance; importance and urgency
- **Tunable by Cognition**
  - Relevance, Sensitivity, Data rate

Reactive Action Subsystem

- **Executes and monitors commands**
- **Direction Connection to Preprocessor for Reactive Control**



Architecture Features of BB1 Cognitive Subsystem

- Cognition Subsystem
  - Control plans responsive to
    - Buffer overload
    - change in task priorities/critical events appearance
  - Change characteristics of agenda manager for evaluating possible reasoning actions (KSs)
    - Multiple methods (KS) with different trade-off in time vs. quality
  - Change criterion for perceptual subsystem
  - Change amount of effort to find best action

Multi-Agent Systems

Definition of a Multi-Agent System

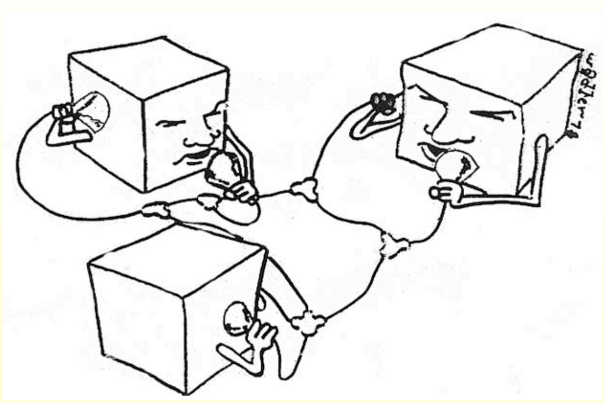
- Multi-Agent Systems are computational systems in which several semi-autonomous agents interact or work together to perform some set of tasks or satisfy some set of goals.
  - agents that are homogeneous or heterogeneous
  - agents having common goals or goals that are distinct
  - agents that are humans or intelligent computational systems

Not concerned with low-level parallelization or synchronization issues that are more the focus of distributed computing.

Features of MAS

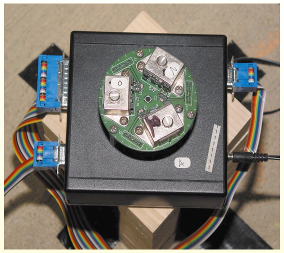
- **Multiple agents connected through low bandwidth communication network**
- **Cooperation- no agent has sufficient information resources to solve entire problem**
- **Decentralized control- no “master” agent**
- **Decentralized data- no global data storage**
- **Loosely coupled- more time spent computing than in communication?**
- **Asynchronous- multiple activities operating in parallel**

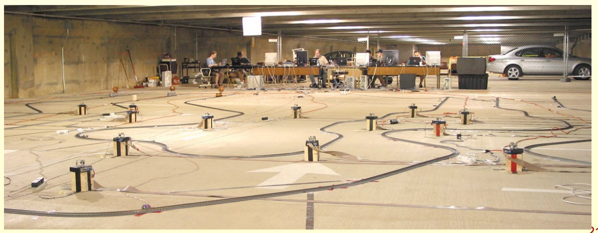
Complexes of sophisticated AI systems that *organize themselves* to work together effectively.



Example MAS Application: Distributed Sensor Network

- Small 2D Doppler radar units (30's)
  - Scan one of three 120° sectors at a time
- Commodity Processor associated with each radar
- Communicate short messages using one of 8 radio channels
- Triangulate radars to do tracking





Representative MAS Issues

- Need for Dynamic Coordination/Distributed Resource Allocation
  - Multiple sensors need to collaborate on tasks
    - View objects of interest from multiple angles with different types of sensors
    - Sensing time windows need to be closely aligned
  - Environmental Dynamics
    - Sensor configuration changes as target moves
  - Potential for Resource Overloads
    - Multiple target in overlapping sensor regions
    - Limited Communication Channels

Representative MAS Issues, cont.

- Soft Real-time **Coordination**
  - Limited time window for sensing
  - Must anticipate where target is moving in order to effectively allocate sensor resources
  - Time for coordination affects time for sensing
- Distribution**: communication latency/limited bandwidth precludes global knowledge/control**
  - distributed data fusion
- Scalability**: need to be able to handle large numbers of sensor nodes**
- Robustness**: local failures should not induce global collapse**
  - Handle uncertain information, sensor/processor/communication failures

Why Build Multi-Agent Systems?

- **Natural decentralization**
  - Model a distributed real-world system
    - distribution of expertise/knowledge,data and resources
  - Agents with individual interests (self-interested)
  - Privacy needs
  - Authority or chain of command issues
- **Ease of development and maintenance**
  - modularity coming from the agent decomposition
  - provide new paradigm for design of complex, highly reliable systems

Nearly-Decomposable MAS

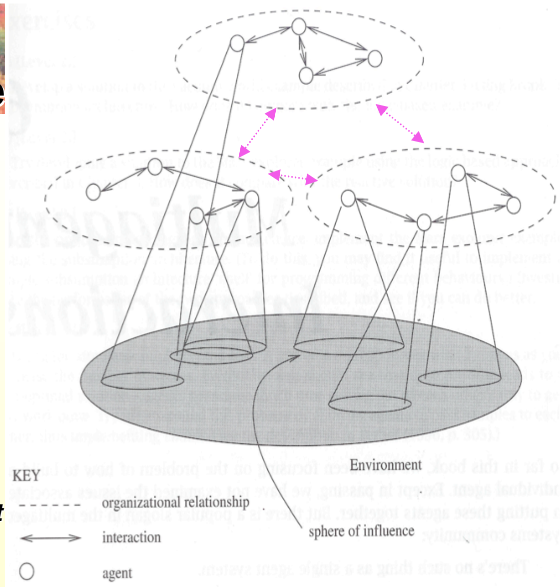


Figure 6.1 Typical structure of a multiagent system.

from
M. Wooldridge's *An Introduction to MultiAgent Systems*. Copyright 2002. John Wiley & Sons, Ltd.

Computational Advantages

- **Easier scale-up (avoids centralized bottlenecks)**
  - Speed-up due to concurrent processing
  - Less communication bandwidth requirements because processing is located nearer the source of information
  - Real-time responsiveness due to processing, sensing and effecting being co-located
- **Robustness, reliability**
  - lack of a single point failure

Framework for systems operating in Open Environments that will Persist Over Time

Some More Advantages

- **extend the range of applications possible for distributed hardware and AI**
- **provide the technology for intelligent agents to cooperate**
- **improvement of control in knowledge-based systems**

- ***Understand cooperation and organizational design***

Example Issues in MAS Design

- **What are the different roles/goals that agents handle**
  - Are these static or dynamically assigned
- **How are agents organized**
  - Hierarchical, peer-to-peer
  - Who makes the decision about role assignment
  - Does the organization change over time
- **What type of protocol is used for information fusion**
  - How is it decided to what, when and to whom to communicate
  - Does all information need to be transferred

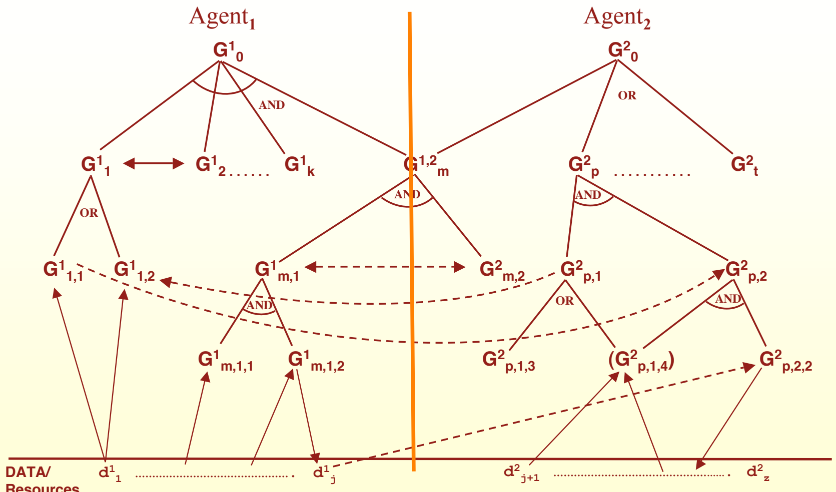
Example Issues in MAS Design, cont

- **What type of protocol is used for agents to coordinate their activity**
  - What type of coordination over resources and relationships among activities is needed
  - How many agents are involved in coordinating (2 or n)
  - How centralized should these protocols be
- **How to handle incomplete,missing, inaccurate information for control and domain problem-solving**

Example Issues in MAS Design, cont

- **Can the system learn how to be more efficient over time**
- **What is the relationship between local agent control and coordination among agents**
- **How should agents relate to each other**
  - Self-interested vs. cooperative
- **How can the system as whole continue functioning if sensors, processors or communication channels fail ……**

MAS as Distributed Search

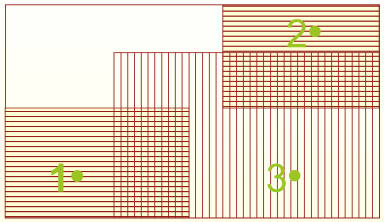


A distributed goal search tree involving $Agent_1$ and $Agent_2$. The dotted arrows indicate interdependencies between goals and data/resources in different agents, solid arrows dependencies within an agent. The superscripts associated with goals and data indicate the agent which contains them (Jennings, 1993 based on Lesser, 1990).

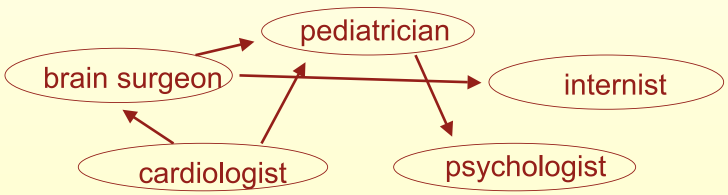
Where do Agents in MAS Come From

- **Spatial, Functional or Temporal distribution of**
  - information, expertise, resources, sensing and effecting
- **Separate authority (lines of control) over resources**
  - organizational imperatives
- **Layered systems' architectures**

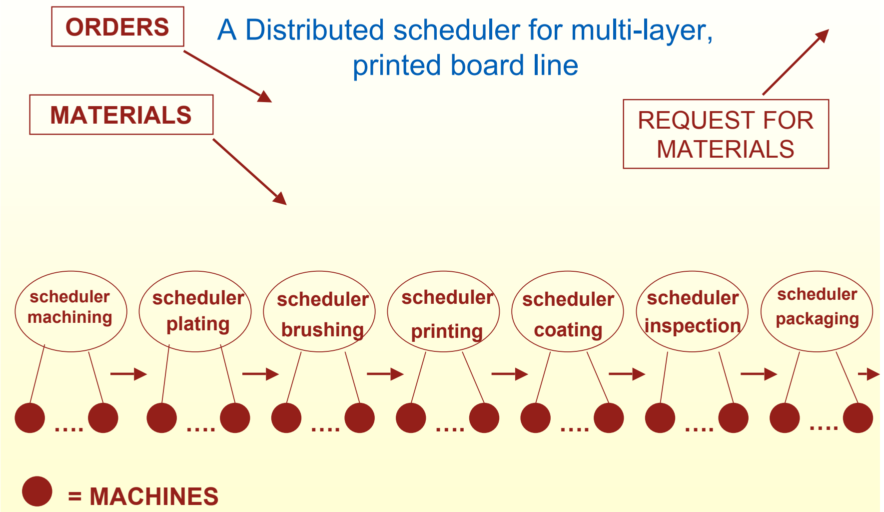
Spatial decomposition (of information) distributed sensor network:



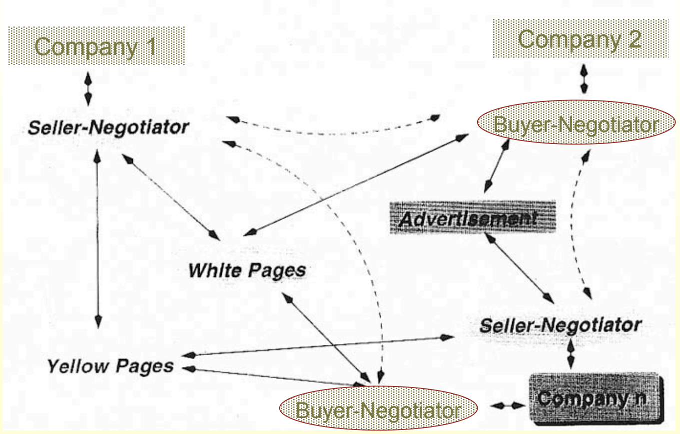
Functional decomposition (of knowledge) group of experts:



Temporal Decomposition (of processing)

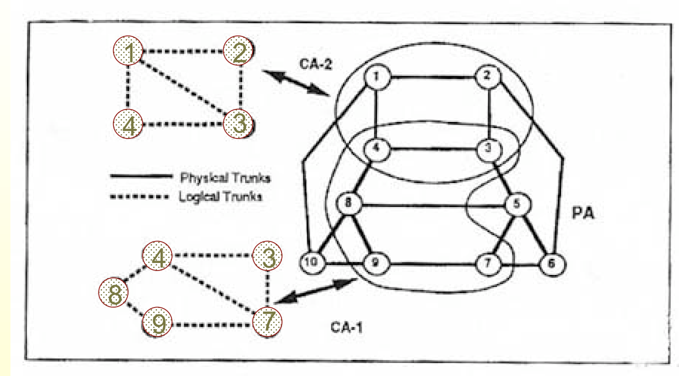


Separate Lines of Authority: Electronic Commerce



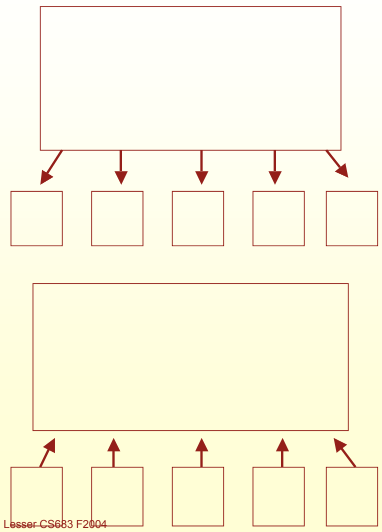
Layered Systems' Decomposition

Multi-agent Customer Network Control: Domain Model



Three-agent Example: Basic Topology
Virtual -- Physical

Designing a Multi-Agent System



- Reductionist Perspective
  - Centralized system that is decomposed/partitioned into a number of agents
  - Encourages a search for ways of “pulling apart” existing centralized systems
- Constructionist Perspective
  - A distributed system that is synthesized from individual systems operating at each node
  - Encourages a search for ways of organizing agents into a society

MAS Applications

- **Distributed situation assessment applications, such as distributed network diagnosis**
  - emphasizes how (diagnostic) agents with different spheres of awareness and control (network segments) should share their local interpretations to arrive at consistent and comprehensive explanations and responses.
  - information gathering on the Internet, distributed sensor networks
- **Distributed resource planning and allocation applications, such as distributed factory scheduling**
  - emphasizes how (scheduling) agents (associated with each workcell) should coordinate their schedules to avoid and resolve conflicts over resources, and to maximize system output
  - electronic commerce, enterprise integration, network management

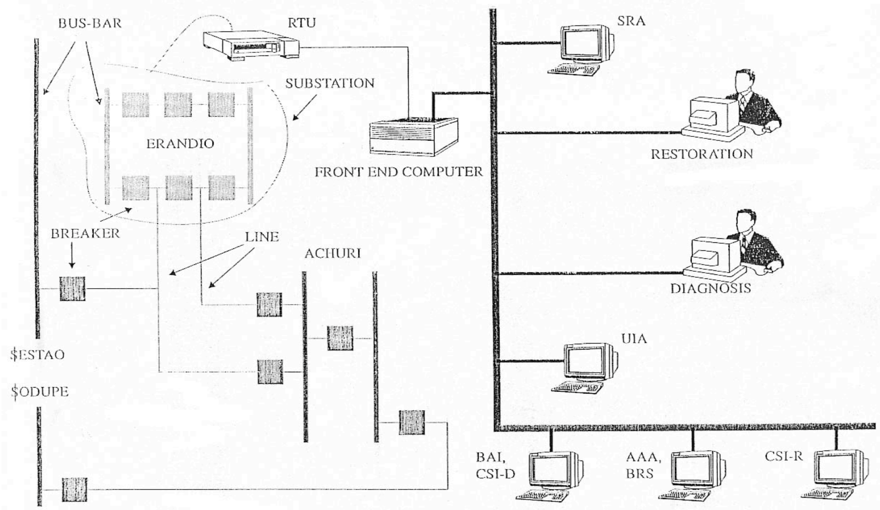
MAS Applications (continued)

- **Distributed expert systems applications, such as concurrent engineering**
  - emphasizes how agents negotiate over collective solutions (designs) given their different expertise and criteria
  - Agent mediated cooperative work
- **The next generation of applications alluded to above will probably involve all of the emphases of these generic applications, and more.**

Cooperating Expert Networks

- **Comprised of semi-autonomous knowledge-based systems with some or all of the following characteristics:**
  - Different areas of expertise (possibly overlapping)
  - Different problem-solving architectures
  - Incomplete or globally inconsistent knowledge
  - Conflicting local goals
- **No central authority able to make good decisions:**
  - Not enough technical expertise
  - No access to private system information

The Multi-Agent System



Cooperation (1)

- **CSI, AAA & BAI**
  - CSI detects fault and informs AAA & BAI
  - AAA performs approximate fault diagnosis
  - BAI produces initial black out area
  - AAA
    - Starts detailed and time consuming validation
    - Uses black out area to prune search space when (if) it arrives

Different problem solving paradigms-
AAA traditional diagnosis approach
BAI monitoring approach

Cooperation (2)

- **AAA & BRS**
  - Both trying to diagnose fault in network
  - Exchange partial results to focus searches towards promising areas and away from unpromising ones
  - Exchange final results to increase/decrease confidence in solutions

Different data models-
AAA non-chronological alarms;
BRS chronological alarms

Cooperation (3)

- **SRA, BAI, AAA & BRS**
  - AAA & BRS inform the SRA of likely diagnosis
  - SRA devises restoration plan
  - BAI checks execution of plan is as it should be
    - AAA & BRS monitor their diagnosis
    - SRA informed of unexpected events which may lead to a re-plan

What is Coordination?

"Coordination is the process of managing interdependencies between activities"

- Malone and Crowston, 1991

Coordination problems occur when:

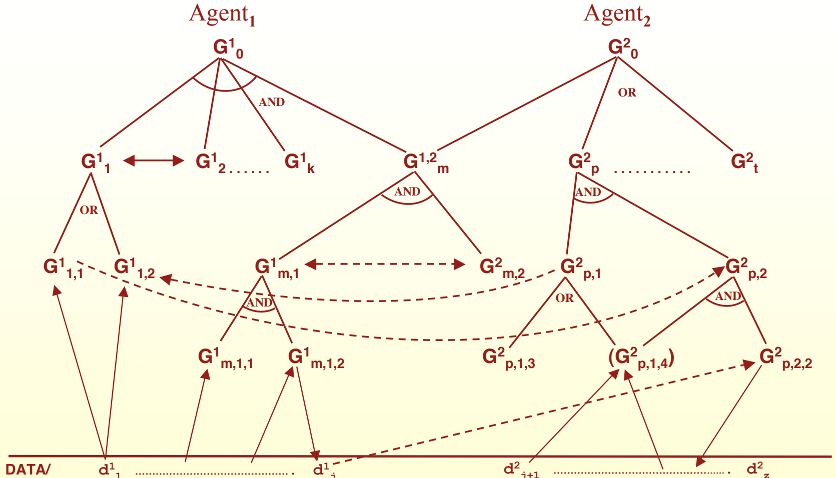
- An agent has a choice in its actions within some task, and the choice affects performance
- The order in which actions are carried out affects performance
- The time at which actions are carried out affects performance

Subproblem Interdependencies

Engenders the Need for Coordination

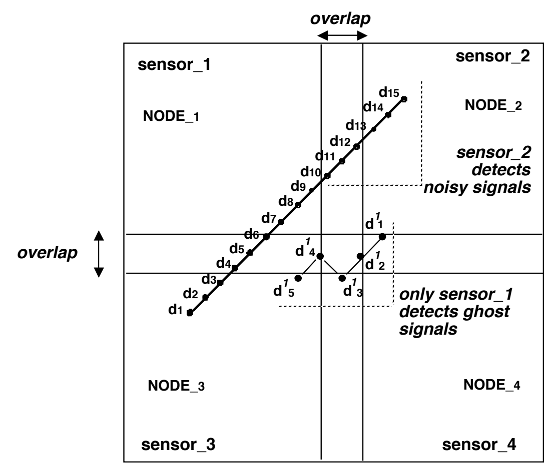
- Contention for resources or through relationships among the subproblems.
- Subproblems are the same/overlapping, but different agents have either alternative methods or data that can be used to generate a solution.
- Two subproblems are part of a larger problem in which a solution to the larger problem requires that certain constraints exist among the solutions to its subproblems.
- Not possible to decompose the problem into a set of subproblems such that there is a perfect fit between the location of information, expertise, processing, and communication capabilities in the agent network and the computational needs for effectively solving each subproblem.

Coordinating Distributed Search



A distributed goal search tree involving Agent$_1$ and Agent$_2$. The dotted arrows indicate interdependencies between goals and data/resources in different agents, solid arrows dependencies within an agent. The superscripts associated with goals and data indicate the agent which contains them (Jennings, 1993 based on Lesser, 1990).

DSN:A Situation Needing Several Styles of Cooperation



The four overlapping sensors detect signal data at particular locations for discrete sensed times (the dots with associated times). Sensor_2 is faulty and not only generates signal data at the correct frequencies for each location but also detects noisy signals at spurious frequencies for each location. Node 1 is connected to sensor 1.

Cooperative Interactions in DSN

- Exploiting Predictive Information
  - **Node 1 should send information to Node 2 to help resolve ambiguity and speed up processing**
- Avoiding Redundant Activity
  - **Nodes 1–3 should avoid redundant work on overlapping sensor data**
- Task Sharing
  - **Node 4 should take on tasks (coordinating or domain) because it is idle**
- Resolve Ambiguity
  - **Nodes 1–4 should work together to detect ghost signals**
- Answer Construction
  - **Nodes 1–3 should develop plan, based on load, about how to integrate results into area-wide map**

Centralization vs. Decentralization

- **Degree of control/data centralization**
  - Optimality of decision based on amount of non-local context exploited
    - How important is optimality?
  - Cost of acquiring non-local context
    - End-to-End Delays
    - Overloading of communication channels
  - Computational Processing required to analyze larger context
- **Different Issues need different degrees of centralization**
  - System Architecture may mix and match different mechanisms on an issue by issue basis to achieve an appropriate levels of control centralization

Cooperative vs. Self-Interested Agents

- Cooperative…

  …agents work toward common goal

- Self-interested…

  …agents work toward own goals but require help from other agents to complete them

Cooperative vs. Self-Interested Agents, p.2

- Does this lead to different approaches?
  - cooperative agents may disagree because of conflicting local perspectives
  - cooperative agents may contribute to common goals by following own local goals (skeptical nodes)
  - self-interested agents may be willing to share information if there is a lot of uncertainty in their decisions
- Is the utility function for evaluating actions the only difference between cooperative and self-interested agents?
  - doing an optimal calculation with complete (global) and up-to-date information may be impractical for either approach
  - approximate calculations with partial and out-of-date information blur the boundaries

Provides a Model for Computation in the 21st Century

Network of cooperating, intelligent agents (people/machines)

- constructionist perspective
  - build out of heterogeneous systems
  - high-level artificial language for cooperation
  - problem solving for effective cooperation will be as or more sophisticated than the actual domain problem solving
    - reasoning about the goals, plans, intentions, and knowledge of other agents

A Model for Computations in the 21st Century, p. 2

- operate in a "satisficing" mode
  - do the best they can within available resource constraints
  - deal with uncertainty as integral part of network problem solving
    - due process (Hewitt)/negotiation
  - complex organizational relationships among agents
    - Scaling to 100s and 1000s of agents

A Model for Computations in the 21st Century, p. 3

- Highly adaptive/highly reliable
  - learning will be an important part of their structure (short-term/long-term)
  - able to adapt their problem-solving structure to respond to changing task/environmental conditions

Profound implications for AI and Computer Science!

Course Summary

Overall Goal of This Course

Understanding the nature of computational structures to support intelligent reasoning in 'Open Environments"

- **Dealing with Uncertainty in all its Guises**
  - Search to resolve uncertain control information
  - Reasoning with Uncertain Information
  - Making Decision Under Uncertainty
  - Learning Information to Resolve Uncertainty

How Does This Course Relate to This Model?

- **"Satisficing" Computation/Resource-Bounded Reasoning**
  - Computational framework that allows you to trade off the quality of the answer derived with the amount of resources used to derive it
- **Uncertainty/Inconsistency as Integral part of problem solving**
  - Computational framework that allows you to live with it — rather than eliminate it
- **Intelligent Control**
  - Computational framework that allows you to effectively manage your resources to satisfy the given goals
- **Agency/Semi-Autonomous Agent**
  - Computational framework that allows agents to interact autonomously with the world in terms of sensing, perceiving, planning, effecting and communicating

Course objectives revisited

After this course you will be able to...

- understand state-of-the-art AI techniques
- construct simple AI systems
- read the AI literature
- evaluate AI-related technology claims
- apply AI techniques in non-AI settings
- pursue specialized AI courses and research

Understanding New directions in AI

- **Abstraction and approximation**
- **Resource-bounded techniques**
- **Meta-level reasoning and control**
- **Issues of Scale and Openness**
  - Large knowledge bases
- **Multi-agent systems**
- **On-line learning**

Basis for Taking Advanced AI Courses

- Machine Learning
- Reinforcement Learning
- Case-Based Reasoning
- Empirical Methods
- Robotics
- Natural Language Processing
- Computer Vision (2)
- Multiagent Systems
- Reasoning and Acting under Uncertainty
- Combinatorial Optimization

Topics We've Covered

- Advanced Search Techniques
  - Research-bounded techniques
  - Multi-level
  - Subproblem interaction
  - Non-monotonic domains
  - Meta-level control
- Reasoning about Uncertain Information
  - Bayes-Nets
  - Fuzzy-Logic
  - Shafer-Dempster
- Rational Decision Making under Uncertainty
  - Utility Theory
  - Value of Information
  - Decision Networks/Influence Diagrams
- Learning
- Agent and Multi-Agent Architectures (sort of?)
  - Reasoure-bounded reasoning

Important Areas Not Covered

- Knowledge Representation
- Non-monotonic reasoning
- Planning
- Scheduling
- Expert Systems
- Hidden Markov Models for Perception

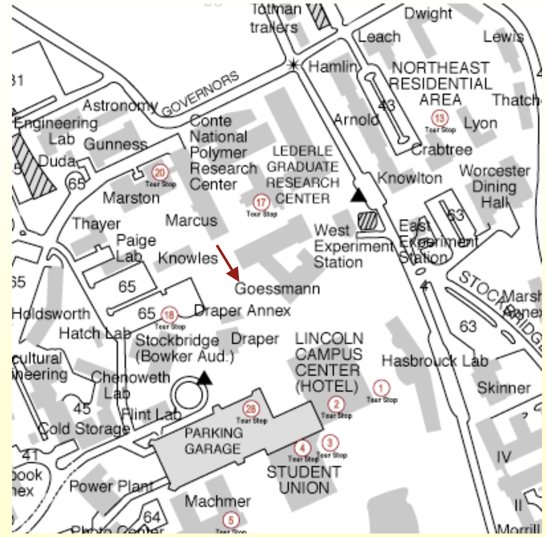
THE END!!!!

Good Luck on Your Exam

Exam

- **Time**
  - Friday 12/17 8-10am.
- **Location**
  - GSMN 51 Goessmann Lab
- **Open Book**
- **Only on Material not covered on Midterm**

Exam Location



Material for Exam

- **Rational Decision Making under Uncertainty**
  - Utility Theory
  - Value of Information
  - Decision Networks/Influence Diagrams
- **Learning**
  - Decision trees
  - Reinforcement learning
    - Dynamic programming
  - Neural networks
  - Instance-based learning
    - Case-based learning
  - Analytic learning
    - EBL
  - Relational learning (guest lecture)
- **Resource Bounded Reasoning**
- **Multi-Agent Systems**