

# Dynamic Observations in Distributed Discrete Event Systems

Weilin Wang, Stéphane Lafortune, Feng Lin, and Anouck Girard

The University of Michigan, Ann Arbor, USA

DISC Workshop on Distributed Discrete Event Systems at  
*WODES 2010*  
Berlin, Germany

2 September 2010

- Introduction: Dynamic Observations and Sensor Activation
- Problem 1: Optimizing Information Acquisition in Distributed Event Diagnosis
- Problem 2: Transformation of Coobservability to Codiagnosability

# Modeling

Discrete Event System (DES) modeled by (monolithic) automaton  $G$

- $\mathcal{L}(G)$  denotes the language generated by  $G$

*Decentralized* information structure:

- $\mathcal{A}$ : The set of agents (diagnosers or controllers)
- $E_{o,i}$ : Event set whose elements are *directly* observable by  $i \in \mathcal{A}$
- $E_{uo,i}$ : Event set whose elements cannot be *directly* observed by  $i \in \mathcal{A}$

# Dynamic Observations

What does *Dynamic Observations* mean?

- The observation of event occurrences by each agent are *trajectory-dependent*

This arises in many problem instances:

- Agents may turn their sensors on/off dynamically
- Agents may communicate with one another
- Observability properties of an event may depend on the system state; e.g., reconnaissance unmanned aerial vehicle (UAV)

# Dynamic Observations: Notation

- Formally, whether or not an event occurrence is observable by agent  $i$ ,  $i \in \mathcal{A}$ , is captured by the *observation mapping*

$$\omega_i : \mathcal{L}(G) \rightarrow 2^{E_o}$$

- After the occurrence of  $s$ , the next event  $e$  is *seen* or *observed* by agent  $i$  when it occurs after  $s$  if and only if it is in  $\omega_i(s)$

# Dynamic Observations: Information Mapping

- Using the observation mapping  $\omega_i$ ,  $i \in \mathcal{A}$ , we define the corresponding *information mapping* (or projection)

$$\theta^{\omega_i} : \mathcal{L}(G) \rightarrow E_o^*$$

recursively as follows

- For the empty string  $\varepsilon$ ,  $\theta^{\omega_i}(\varepsilon) = \varepsilon$ , and
- for all  $s, se \in \mathcal{L}(G)$  with  $e \in E$ ,

$$\theta^{\omega_i}(se) = \begin{cases} \theta^{\omega_i}(s)e & \text{if } e \in \omega_i(s) \\ \theta^{\omega_i}(s) & \text{otherwise} \end{cases}$$

# The Lack of Monotonicity in Dynamic Observations

Sometimes, if an agent “sees more,” this may cause *distinguishable* trajectories to become *indistinguishable*!

Example:

- $L = \{aab, ab\}$
- If one occurrence of event  $a$  in  $aab$  is not “seen” by an agent, then  $aab$  and  $ab$  looks the same
- But if both occurrences of  $a$  in  $aab$  are not “seen” by the agent, then the agent sees  $aab$  as  $b$  but  $ab$  as  $ab$   
 $\Rightarrow$  the agent can distinguish these two trajectories!
- This is possible when another agent who directly observes all occurrences of  $a$  and  $b$  selectively communicates such occurrences to the given agent

But: *If the agent is responsible for turning its sensors on/off, then this cannot happen!*

- ✓ Introduction: Dynamic Observations and Sensor Activation
  - Problem 1: Optimizing Information Acquisition in Distributed Event Diagnosis  
Reference:  
“Optimal sensor activation for diagnosing discrete event systems,” by Weilin Wang, Stéphane Lafortune, Anouck Girard, and Feng Lin, *Automatica*, 46 (2010) pp. 1165–1175.
  - Problem 2: Transformation of Coobservability to Codiagnosability

# Motivation

Information acquisition (dynamic sensor activation) may be *costly*, e.g., if sensors are operated in environments with limited resources (energy, bandwidth, security, etc.)

- In a UAV system, making a measurement may cost hours of flight
- In a radar system, emitting radar signals can be dangerous
- The life span of a sensor often depends on how often it is used



Figure: Predator



Figure: UAV Pilot

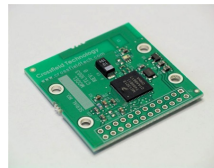


Figure: Wireless Sensor

# Information Acquisition in Partially Observed DES

Estimation and *information acquisition* (IA) are agent-wise interdependent

- What you acquire in the past affects your estimation ( $\theta_i$ )
- What you estimate influences your future acquisition decisions ( $\omega_i$ )

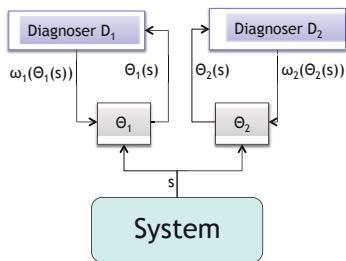


Figure: IA for Distributed Diagnosis

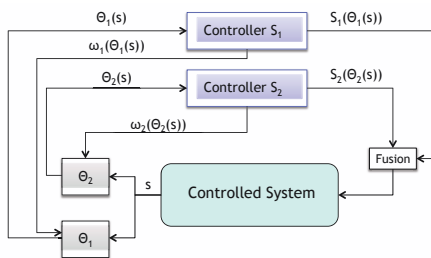


Figure: IA for Distributed Control

# Information Acquisition in Partially Observed DES

- Static sensor optimization:  
[Haji-Valizadeh et al., 1996], [Debouk et al., 2002], [Jiang et al., 2003]
- Dynamic sensor activation for diagnosis or opacity:  
[Thorsley-Teneketzis, 2007], [Cassez-Tripakis, 2008], [Cassez et al., 2009]
- Optimizing sensor activation for state disambiguation (control):  
[Wang et al., 2010]
  
- Communication for control and diagnosis in decentralized control problems:  
[Wong-van Schuppen, 1996], [Ricker-Rudie, 1999], [Barrett-Lafortune, 2000], [Rudie et al., 2000], [van Schuppen, 1998, 2004], [Boel-van Schuppen, 2002], [Wang et al., 2008], [Ricker-Caillaud, 2009], and more.

# Information Acquisition Policy

*Information acquisition policy* (IAP) is a specific class of observation mapping

- 1 Instead of  $2^{E_o}$ , the range of observation mapping becomes  $2^{E_{o,i}}$ :

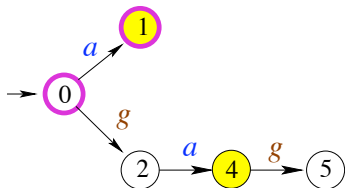
$$\omega_i : \mathcal{L}(G) \rightarrow 2^{E_{o,i}}$$

- 2 *Feasibility in the context of Information Acquisition*: two trajectories that have the same information mapping must be followed by the same acquisition decisions:  $(\forall s, t \in \mathcal{L}(G))$

$$\theta^{\omega_i}(s) = \theta^{\omega_i}(t) \implies \omega_i(s) = \omega_i(t)$$

# Information Acquisition Policy: Feasibility

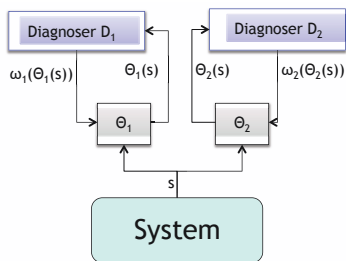
- If two strings “look the same,” then they must have same activation decision on a *common possible event*
- If  $a$  is activated initially, then:  
Not activating  $g$  initially **and**  $a$  after string  $g$  is not *feasible*!  
→ after  $\epsilon$  and  $g$ , we must have the same activation decision for  $a$
- This is called the *feasibility* requirement of IAP



# Problem Statement (Informal)

Since information acquisition is “costly,” it is desirable to find a *minimal, feasible* IAP while maintaining the property of *codiagnosability*.

→ The enforcement of codiagnosability is one approach to break the dependency between estimation and information acquisition.



Issues:

- Codiagnosability
- Optimization space
- Monotonicity

Figure: IA for Distributed Diagnosis

# Codiagnosability

- $E_F$ : Set of fault events to be diagnosed
- $\Pi_F$  is obtained by partitioning  $E_F$  into disjoint sets of different fault types:

$$E_F = E_{f_1} \dot{\cup} \dots \dot{\cup} E_{f_K}$$

- $s \in \Psi(f_k)$  denotes that the last event of  $s \in \mathcal{L}(G)$  is a fault event of type  $f_k$
- $\mathcal{L}(G)/s = \{t \in E^* : st \in \mathcal{L}(G)\}$  is the postlanguage of  $\mathcal{L}(G)$  after  $s$
- $f_k \in s$  means that  $s$  contains an event in  $E_{f_k}$

# Codiagnosability

## Definition (Codiagnosability)

A prefix-closed and live language  $\mathcal{L}(G)$  is said to be codiagnosable with respect to  $\theta_i$ ,  $i \in \mathcal{A}$ , and  $\Pi_F$  on  $E_F$  if:

$$(\forall k \in \Pi_F)(\exists n_k \in \mathbb{N})(\forall s \in \Psi(f_k))(\forall t \in \mathcal{L}(G)/s)[|t| \geq n_k \Rightarrow CD]$$

where the codiagnosability condition  $CD$  is

$$(\exists i \in \mathcal{A})(\forall \mu \in \mathcal{L}(G))\theta_i(\mu) = \theta_i(st) \Rightarrow f_k \in \mu$$

In words, under information mappings  $\theta_i$ ,  $i \in \mathcal{A}$ ,  $\mathcal{L}(G)$  is codiagnosable if after any  $s \in \mathcal{L}(G)$  ending with an event in  $E_{f_k}$  occurs, there is always an agent  $i \in \mathcal{A}$  that can infer, after a finite number of event occurrences, that an event in  $E_{f_k}$  has occurred.

# Language-based Partition

Why partition the language?

- The domain of IAP

$$\omega_i : \mathcal{L}(G) \rightarrow 2^{E_{o,i}}$$

typically has infinite cardinality

- The solution space for IAP should be finite

How to partition the language?

- Could use the state space of  $G$ : *state-based* or *transition-based* solutions
- More general approach: Define IAP over a *finite partition* of  $\mathcal{L}(G)$
- Constraint: *All strings in the same element of the partition must have the same last event and the same acquisition decision for their last event*

# Language-based Partition

- $\Delta$  is Language-based Partition (LBP) if its elements  $\delta_j$ ,  $j = 0, 1, \dots, m$ , satisfy
  - $\Delta$  is a partition of  $\mathcal{L}(G)$
  - $\delta_0 = \{\varepsilon\}$  is a singleton
  - Any strings in the same element  $\delta_j \in \Delta$  must have the same last event
- IAP:  $\Omega \subseteq \Delta$ 
  - For all  $\delta_i \in \Omega$ , when the last event of a string in  $\delta_i$  occurs, the information for that event is acquired
- Notation:
  - In a distributed system with  $\mathcal{A} = \{1, \dots, N\}$ ,  $\bar{\Delta} = [\Delta_1, \dots, \Delta_N]$ ,  $\bar{\omega} = [\omega_1, \dots, \omega_N]$ , and  $\bar{\Omega} = [\Omega_1, \dots, \Omega_N]$

# Problem Statement

- For a system  $G$  with set of agents  $\mathcal{A}$  and a given vector of LBP  $\bar{\Delta}$ , find a *minimal* and feasible IAP  $\bar{\Omega}^*$  under which  $G$  is codiagnosable.
- IAP  $\bar{\Omega}^*$  is minimal if  $G$  is not codiagnosable under all other feasible IAP  $\bar{\Omega} \subset \bar{\Omega}^*$
- $\bar{\Omega} \subset \bar{\Omega}^*$  means that, for all  $i \in \mathcal{A}$ ,  $\Omega_i \subseteq \Omega_i^*$  and, for at least one  $j \in \mathcal{A}$ ,  $\Omega_j \subset \Omega_j^*$

# Main Theorems

## Theorem (Monotonicity of IAP in Diagnosis)

*Let feasible IAPs  $\bar{\omega}'$  and  $\bar{\omega}''$  satisfy  $\bar{\omega}' \subseteq \bar{\omega}''$ . Then,  $\mathcal{L}(G)$  codiagnosable under  $\bar{\omega}'$  implies  $\mathcal{L}(G)$  codiagnosable under  $\bar{\omega}''$ .*

- Recall that for general observation mappings, monotonicity is not always true
  - Monotonicity may not hold when agents communicate event occurrences
- IAP satisfies monotonicity because
  - An agent must make the same acquisition decisions right after trajectories that look the same to that agent and
  - An agent changes to its IAP only affect its own observations

# Main Theorems

## Theorem (Maximum Feasible Subpolicy for an Agent)

*Given  $G$  with a fixed LBP  $\Delta$ , let  $\Omega \subset \Delta$  be an observation mapping that is not feasible. Then, among all feasible IAP  $\Omega' \subseteq \Omega$ , there exists a maximum feasible IAP  $\Omega^{\uparrow F} \subseteq \Omega$  s.t.  $\Omega' \subseteq \Omega^{\uparrow F}$ .*

- The proof follows from the facts that  $\Delta$  is finite and that the union of two feasible IAPs is also a feasible IAP, where:
  - under the union of two IAPs, the agent decides to acquire the information of the next occurrence of an event iff the agent decides to acquire such information under either or both of these two IAPs

# Algorithm MAIN

INPUT: System  $G$ , agent set  $\mathcal{A}$ , observable events set  $E_{o,i}$ , vector of LBP  $\bar{\Delta}$ , sets of fault events  $E_{f_k}$ , and a feasible IAP  $\bar{\Omega}$  (vector of nonempty sets) under which the system is codiagnosable

OUTPUT: Minimal sensor activation policy  $\bar{\Omega}^*$

ITERATIONS:

- Pick one agent and remove “one” set of sensor activations by assigning one element of its IAP to “no activation”  
→ Recall that IAP is a subset of LPB:  $\Omega \subseteq \Delta$
- Use the existence of the Maximum Feasible Subpolicy  
→ Run algorithm for  $\uparrow F$  after a removal
- Test codiagnosability
- Repeat

# Algorithm MAIN

## ITERATIONS:

- ① Initialization: Set  $D = [D_1, \dots, D_N]$  being the vector of  $N$  empty sets
- ① Select an agent  $i \in \mathcal{A}$  and a  $\delta_{i,k} \in \Omega_i$  that has not been examined yet.  
 Let  $\Omega'_i$  be the result of deleting  $\delta_{i,k}$  from  $\Omega_i$ .  
 Calculate  $\Omega_i^{\uparrow F}$  (the maximum IAP of  $\Omega'_i$ ).  
 Let  $\bar{\Omega}'$  be the policy obtained by replacing  $\Omega_i$  with  $\Omega_i^{\uparrow F}$  in  $\bar{\Omega}$
- ② Test codiagnosability for  $\bar{\Omega}'$ .  
 If it holds, set  $\bar{\Omega} \leftarrow \bar{\Omega}'$ .  
 Otherwise, set  $D_i \leftarrow D_i \cup \{\delta_{i,k}\}$
- ③ Repeat Step 1 to 2 until  $\bar{D} = \bar{\Omega}$ .  
 Then, set  $\bar{\Omega}^* \leftarrow \bar{\Omega}$

# Intuitive Explanation of Algorithm MAIN

- We try to remove  $\delta_{i,k}$  from  $\Omega_i$  by setting  $\Omega'_i \leftarrow \Omega_i \setminus \{\delta_{i,k}\}$
- Since  $\Omega_i^{\uparrow F}$  is the maximum subpolicy of  $\Omega'_i$ , by the monotonicity theorem, if the test of codiagnosability fails under  $\bar{\Omega}'$ ,  $\delta_{i,k}$  *cannot be removed* from  $\Omega_i$  without increasing acquisitions of some other agent(s)
- We add  $\delta_{i,k}$  into  $D_i$  to record that the removal failed
- If the codiag test passes and moreover nothing from  $D_i$  was removed by  $\uparrow F$ , we shrink  $\bar{\Omega}$  by setting it to be  $\bar{\Omega}'$
- Since we always reduce acquisition and reducing acquisition always harms codiagnosability, for all  $i \in \mathcal{A}$ , all elements in  $D_i$  can never be removed when the algorithm proceeds further
- Hence, we have  $\bar{\Omega} = \bar{D}$  within finite iterations and when  $\bar{\Omega} = \bar{D}$  we know that a minimal solution has been obtained

# Algorithm MAIN: Discussion

Comments:

- $\bar{\Omega}^*$  depends on the order in which elements in  $\bar{\Delta}$  are considered
- In principle, all minimal solutions could be computed
- In practice, one could follow a priority ordering for the sensors and agents

Still to be specified:

- Algorithm to do  $\uparrow F$
- Algorithm to test Codiagnosability [Wang et al., ECC 2009]

# Maximum Feasible Subpolicy: Window Partitions

- The calculation of maximum subpolicy  $\Omega_i^{\uparrow F}$  depends on how language  $\mathcal{L}(G)$  was partitioned by an LPB
- We developed a method to partition language  $\mathcal{L}(G)$ , called *Window Partition*  $\Delta^w$ , for which  $\uparrow F$  can be computed
- Intuition: For  $\delta_i \in \Delta^w$ , whether a trace  $s \in \delta_i$  or not is determined by both *the state reached before the last event in  $s$*  and *the suffix of the last  $n$  event occurrences of  $s$*
- Reasons:
  - Current state and recent event occurrences are important in practice
  - Window partitions are amenable to refinement (choice of  $n$ ), which achieves balance between computational effort and quality of solutions
  - If  $n = 1$ , then we recover “transition-based” approaches

# Window Partitions: Formal Definition

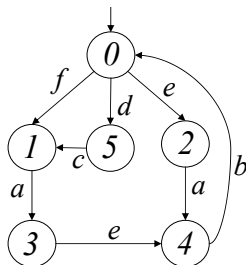
The set  $\Delta^w = \{\delta_i \subseteq \mathcal{L}(G) : i = 0, \dots, m\}$  is called an  $n$ -Window-Partition of  $\mathcal{L}(G)$  if

- For all  $u \in \mathcal{L}(G)$ , if  $\|u\| < n$  then  $\{u\} \in \Delta^w$ .
- For all  $u, v \in \mathcal{L}(G)$  with their length larger or equal to  $n$ ,  $u$  and  $v$  are in the same element of the partition iff
  - they have same subsequence of last  $n$  event occurrences and,
  - before the last event occurrence, they reach the same state in  $G$

# Window Partitions: Example

- Window-Partition with  $n = 2$ :

$$\Delta^w = \{ \{ \epsilon \}, \\ \{ f \}, \{ d \}, \{ e \}, \\ (b, 0, f), (b, 0, d), (b, 0, e), \\ (f, 1, a), (c, 1, a), \\ (e, 2, a), \\ (a, 3, e), \\ (e, 4, b), (a, 4, b), \\ (d, 5, c) \}$$



# Maximum Subpolicy: Window Partitions

## Algorithm for $\uparrow F^I$ with Window Partition

- Intuitive Steps:
  - 1 Calculate the possible confusable pairs of elements in  $\Delta^w \times \Delta^w$  by considering current estimation of  $\Omega^{\uparrow F^I}$  and feasibility requirement
  - 2 Using these possible confusable pairs, remove the sensor activations that cause a violation of feasibility
  - 3 Iterate above steps. Stop iteration when no more confusable pair can be found and no more sensor activation should be removed
- Polynomial in the cardinality of  $\Delta^w$

# Example 1: Algorithm MAIN

- $E_{o,1} = \{a, b, d\}$ ,  $E_{o,2} = \{a, c, d\}$ , and  $E_f = \{f\}$
- For simplicity,  $n = 1$  in Window Partition  $\Rightarrow$  transition-based
- $[-]$ : deactivated transition (unobservable);  $(-)$ : activated (observable)

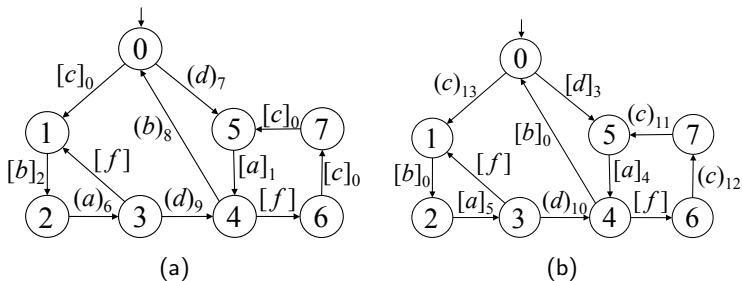


Figure: (a) For Diagnoser 1; (b) For Diagnoser 2

## Example 2: Effect of Window Partition Refinement

- Two-agent system:  $E_{o,1} = \{b, c, d\}$ ,  $E_{o,2} = \{a, d\}$ , and  $E_f = \{f\}$
- An  $\bar{\Omega}^*$  is  $\Omega_1^* = \{(0, c), (3, d), (4, d)\}$  and  $\Omega_2^* = \{(0, a), (3, d), (4, d)\}$
- Refining window partition ( $n = 2$ ), the policy  $\bar{\Omega}$  equivalent to  $\bar{\Omega}^*$  is  $\Omega_1 = \{\{c\}, (f, 3, d), (d, 3, d), (b, 4, d), (e, 4, d)\}$  and  $\Omega_2 = \{\{a\}, (f, 3, d), (d, 3, d), (b, 4, d), (e, 4, d)\}$
- Running Algorithm MAIN starting with this  $\bar{\Omega}$ , we get the “improved” solution:  
 $\Omega_1^* = \{\{c\}, (f, 3, d), (b, 4, d)\}$  and  $\Omega_2^* = \{\{a\}, (f, 3, d), (b, 4, d)\}$

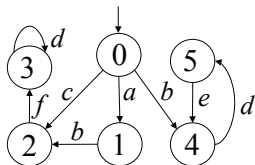


Figure:  $G$

# Summary – Problem 1

- We formulated the problem of dynamic information acquisition in the context of distributed event diagnosis  
*Feasibility implies monotonicity*
- Algorithms were developed for the optimization of information acquisition policies that preserve codiagnosability  
*Minimal solutions can be computed using the notion of “Maximum Feasible Subpolicy”*
- The window partition technique is used to balance the computational efforts and the quality of the solution  
*Algorithm for  $\uparrow F$*
- For a fixed number of agents, the algorithms are of polynomial complexity in the size of the window partition
- From diagnosis to control...

- ✓ Introduction: Dynamic Observations and Sensor Activation
- ✓ Problem 1: Optimizing Information Acquisition in Distributed Event Diagnosis
- Problem 2: Transformation of Coobservability to Codiagnosability  
Reference:  
*Results submitted for publication. Please contact the authors.*

## Decentralized Supervisory Control

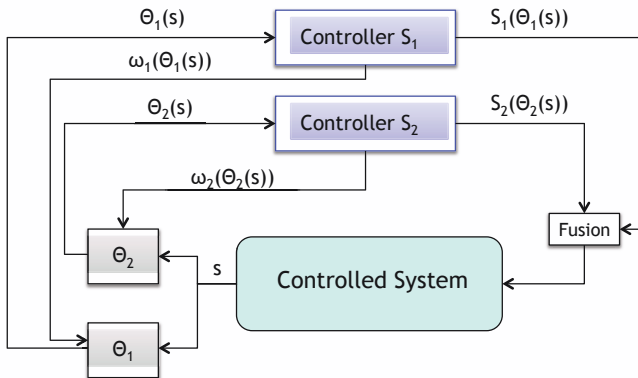


Figure: Decentralized Supervisory Control

# Coobservability

## Definition (Coobservability)

A prefix-closed language  $K = \mathcal{L}(H) \subseteq \mathcal{L}(G)$  is *coobservable* with respect to  $\mathcal{L}(G)$ ,  $\omega_i$ , and  $E_{c,i}$ ,  $i \in \mathcal{A}$ , if for all  $s \in K$  and  $e \in E_c$  with  $se \in \mathcal{L}(G)$ , the existence of  $s_i e \in K$  with  $\theta_i(s_i) = \theta_i(s)$  for all  $i \in A^c(e)$  implies that  $se \in K$ , where  $\theta_i$  is the information mapping for supervisor  $i$  corresponding to  $\omega_i$

In words, under information mapping  $\theta_i$ ,  $i \in \mathcal{A}$ ,  $\mathcal{L}(H)$  coobservable with respect to  $\mathcal{L}(G)$  means the following:

*if an occurrence of  $e$  after  $s \in \mathcal{L}(H)$  is possible but illegal, then there must exist an agent that can safely disable  $e$ , i.e., if a legal  $s_i$  is followed by  $e$  and looks the same as  $s$  for that agent, then  $s_i e$  must also be illegal*

# Context

- Previously: Coobservability and Codiagnosability have been investigated separately
- Our objective: To apply solutions developed for diagnosis problems (codiagnosability) for solving control problems (coobservability)
- Contribution: We show that a given problem of coobservability is reducible to a problem of codiagnosability

# Summary – Problem 2

- We have considered the notions of codiagnosability and coobservability in the general scenario of dynamic observations
- We have presented a polynomial algorithm to transform the problem of coobservability to the problem of codiagnosability
- This enables us to leverage the large literature available for codiagnosability to solve problems of coobservability

# Conclusion

- Many different formulations of the Dynamic Information Acquisition Problem
  - Solution spaces vary; notions of LBP and WBP
  - Look for “structure”; existence of  $\uparrow F$
  - “Information State”
  - Quantitative approaches
  - Computational efficiency is a must
- Cross-fertilization diagnosis – control
  - Does there exist a mapping from codiagnosability to coobservability?
- From IAP to communicating agents...