Towards a Theory of Events

K. Mani Chandy
Michel Charpentier
Agostino Capponi
California Institute of Technology

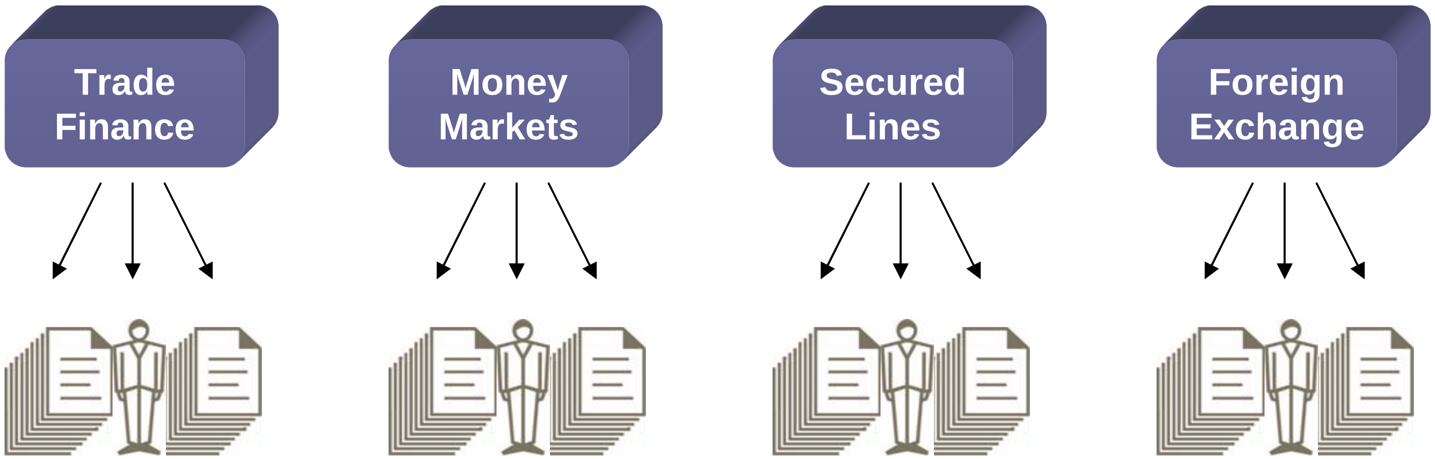
Need for a Theory of Events

- Event processing will become a discipline when there is a unifying theory.
- Consider other disciplines:
  - Control systems: Control theory
  - Database systems: Relational algebra
  - Concurrent computation: temporal logic

Classes of Problems

Respond rapidly to changing conditions

Classes of Problems: Risk Management



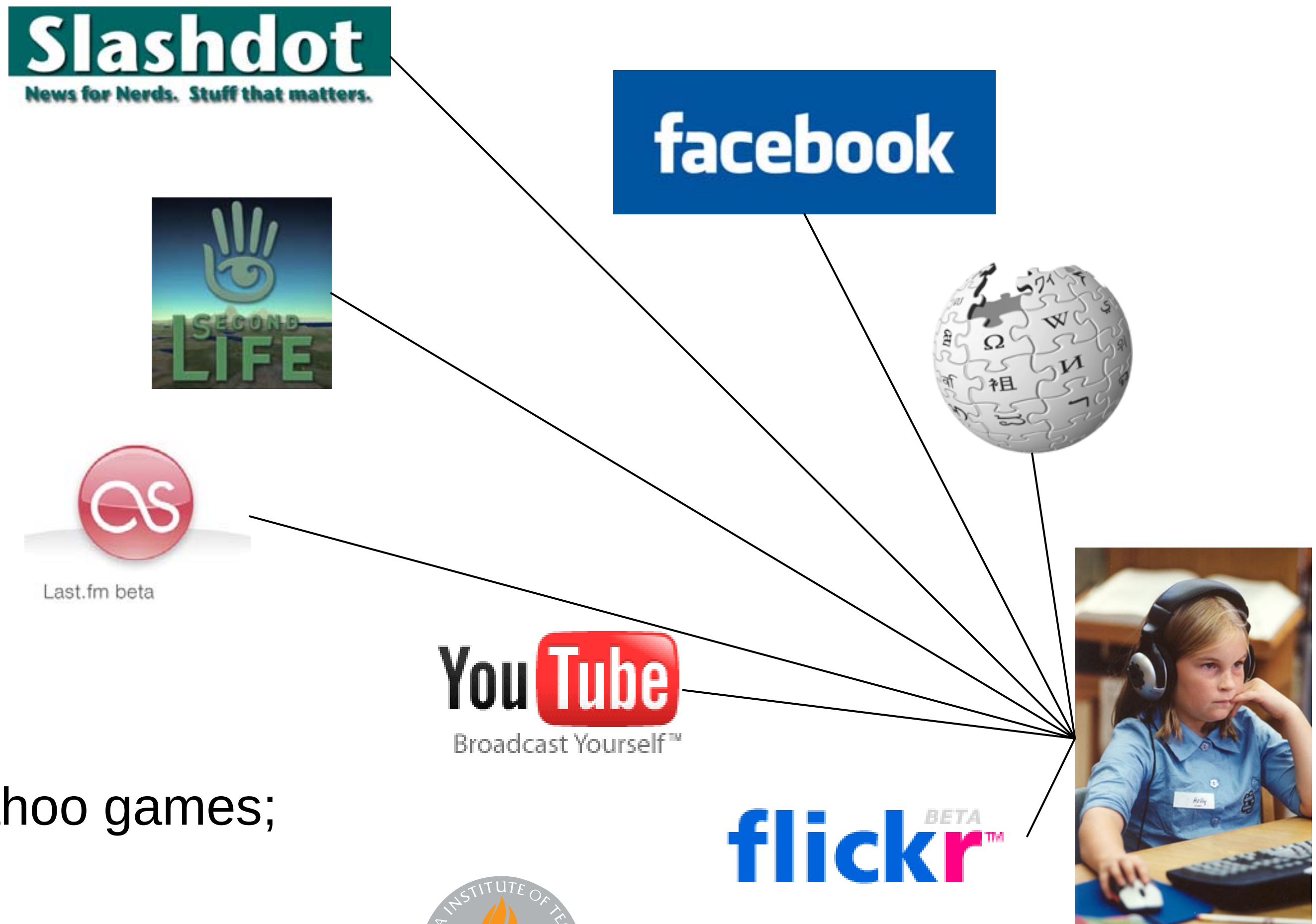
Problem: ***Elapsed time to ascertain limit breach of total credit exposure to single customer or industry segment is 90+ days***

Result: ***Asset impairment leading to increased loss reserves***

Classes of Problems: Individuals

Entertainment and information:

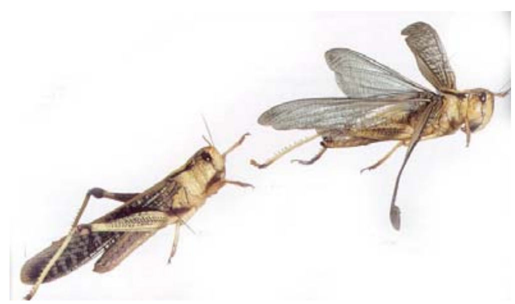
- You Tube
- Face Book
- Second Life
- wikipedia
- eBay
- Amazon
- Flickr
- Google
- Last.fm
- myspace
- msn games; yahoo games; …



Theory Applied to Analysis

Locust swarm: **Global behavior**

Individual locust: **local behavior**



Technology Trends

- Costs of computers, storage, communication dropping rapidly, exponentially and continuously.
- Sensors becoming much more widely available.
- Responder mechanisms widely available: whom to communicate with, how, when
- Miniaturization
- Location sensing; geographical data

What are the scarce resources?

Given rapid exponential decrease in costs of:

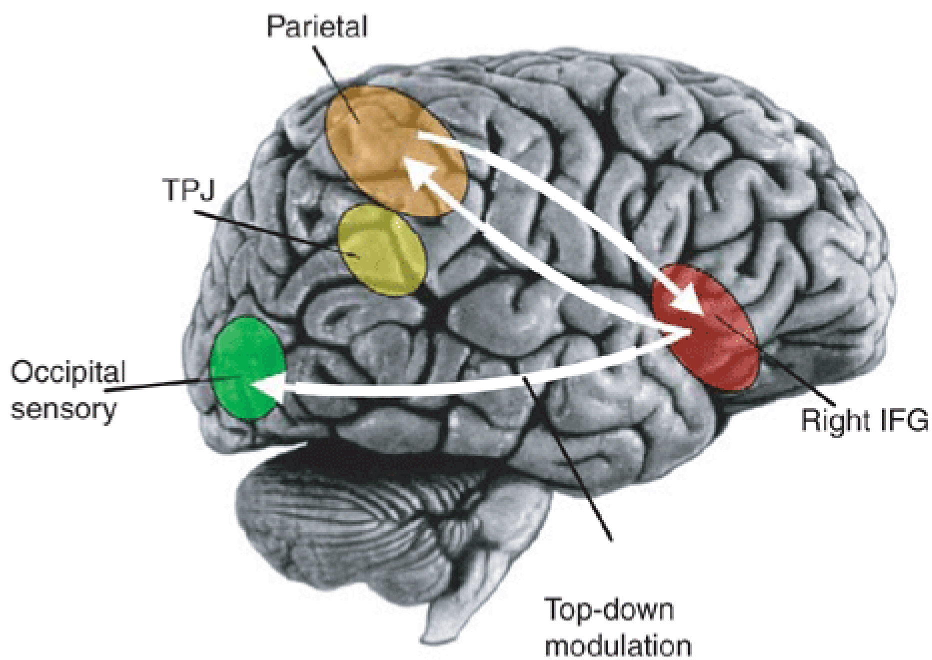
- Storage
- Bandwidth
- Computing capacity
- Energy requirements

Given pressure to:

- Respond even more proactively and rapidly
- React to conditions ***outside*** the organization:
- Handle huge data volumes and creation rates

What are the scarce resources?

- Time and attention



What is the Value of a Theory?

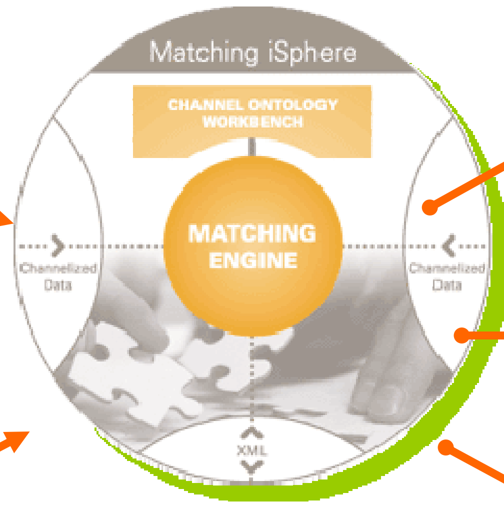
- Systematic analysis of a class of problems
- Systematic design of architecture and its components.

- Analysis of what class of problems?
- Designs of what components? What types of architectures?

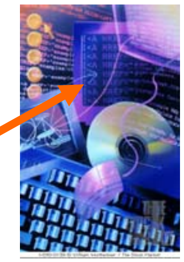
Components of an EP Application

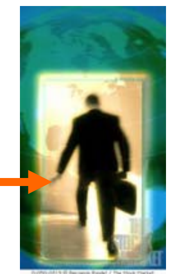
External Event

Information acquisition



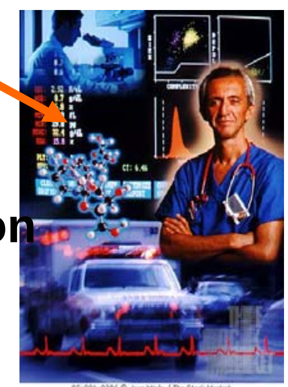
Information Fusion Simulation





Actuator

Actuation



Theory Applied to Design

Design

1. Network
2. Sensors
3. Compute engines and databases
   Event processors; simulators; optimizers,..
4. Responders
5. EPMS (management layer)

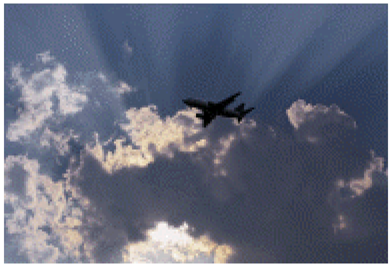
to optimize scarce resources

Types of Sense & Respond Applications

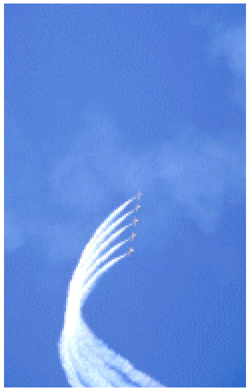
- Monitoring


Environmental monitoring

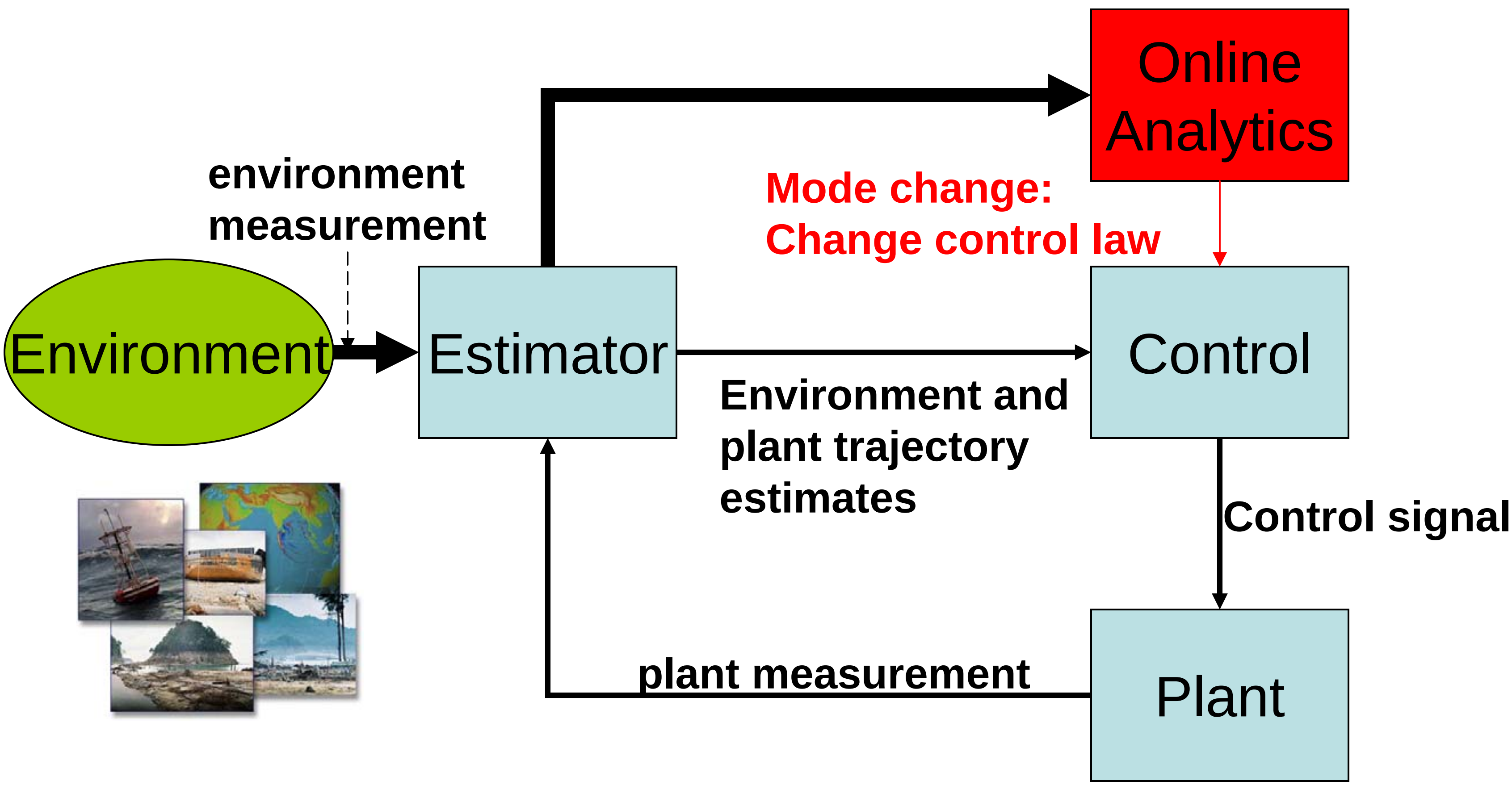
- Continuous control


Coasting plane

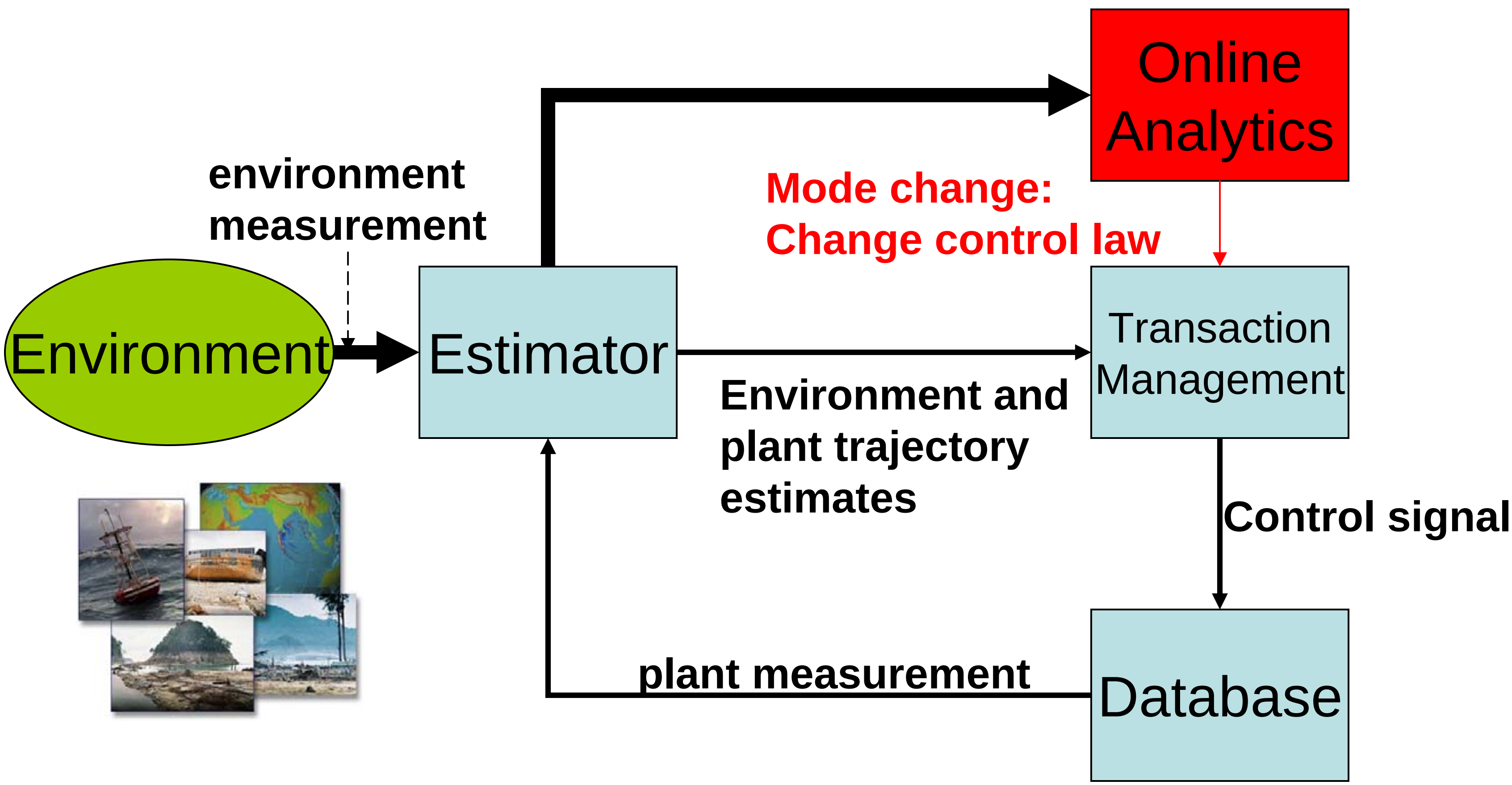
- Mode change


Plane changing mode

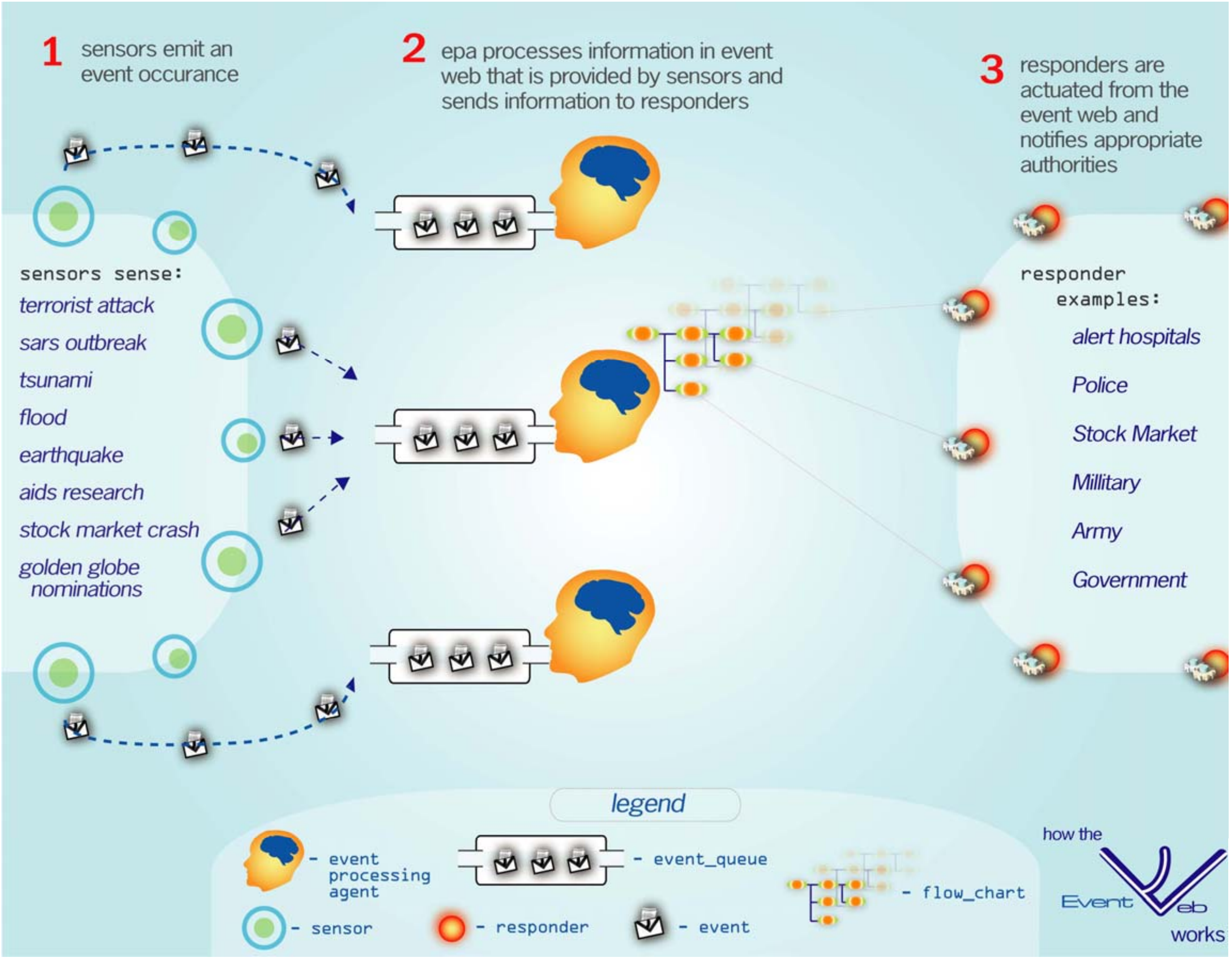
Event Processing



Event Processing



Distributed Mode Changes



Information Fusion: Heterogeneous Data


The Daily

Sunday, August 30, 2006

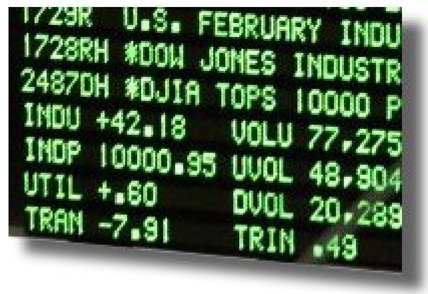
Martians invade earth

Incredible as it may seem, it has been confirmed that a large martian invasion fleet has landed on earth tonight.

First vessels were sighted over Great Britain, Denmark and Norway already in the late evening from where, as further reports indicate, the fleet headed towards the North Pole and Santa Claus was taken hostage by the invaders.

Afterwards they split apart in order to approach most major cities around the earth. The streets filled as thousands fled their homes, many only wearing their pajamas...

Text



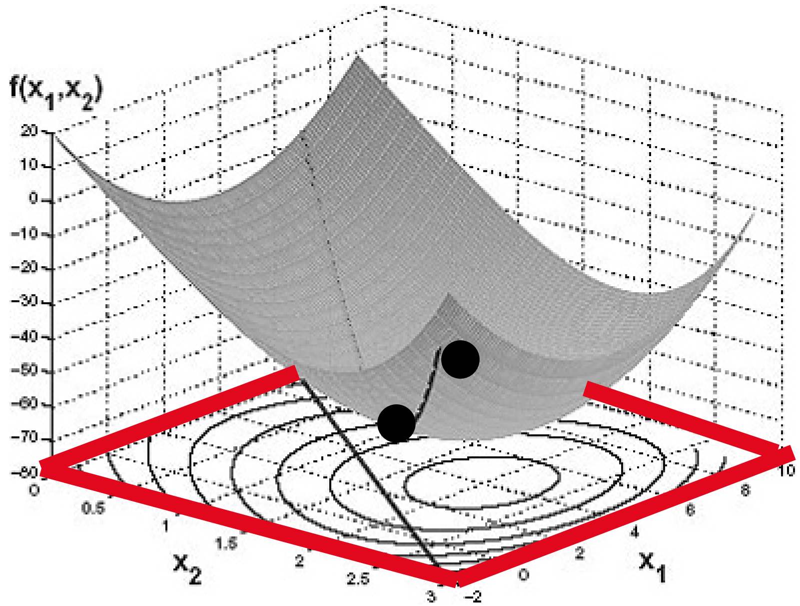
Numeric

Analog

Images

Voice & image

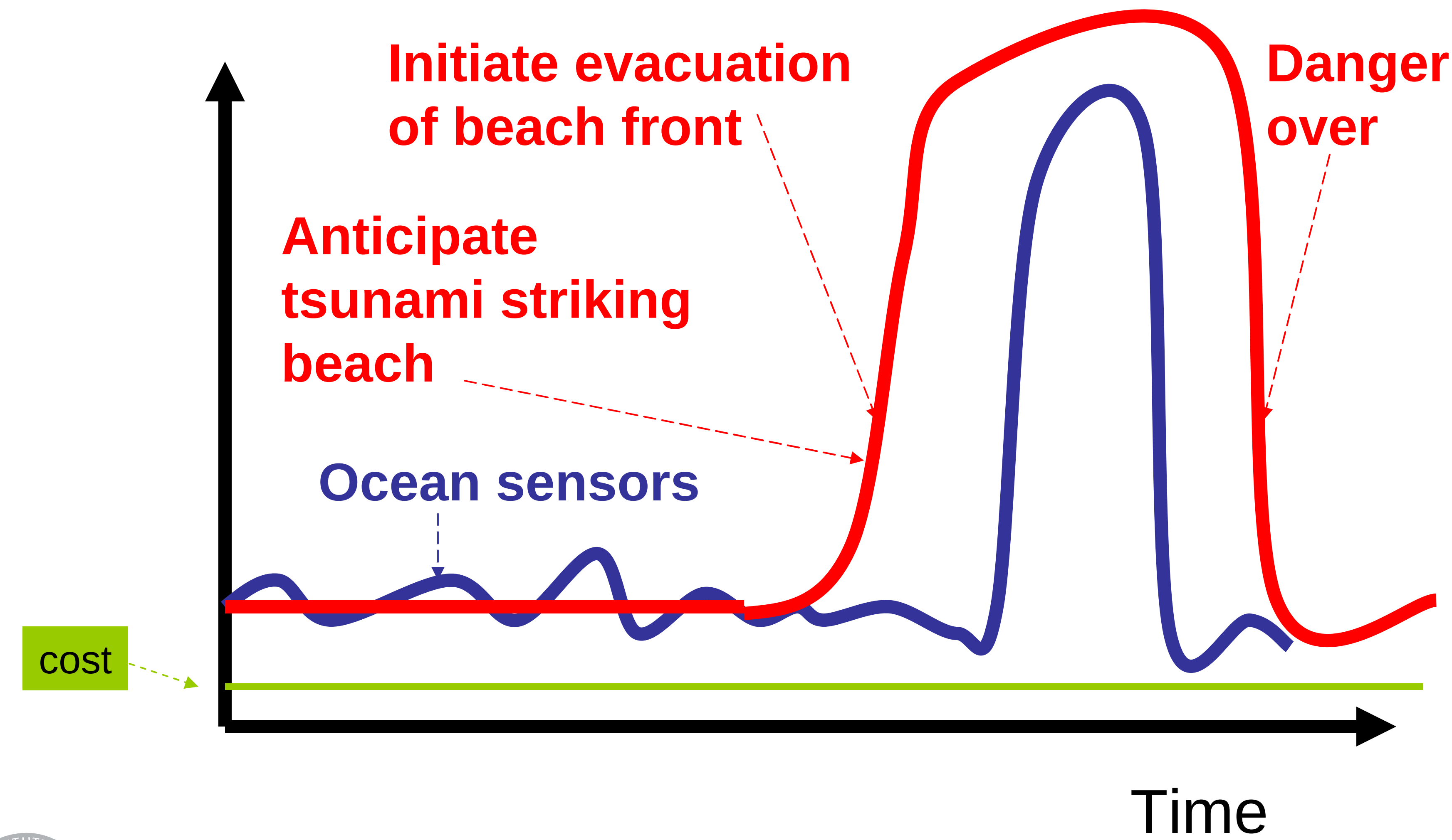
Design/Analysis of Event Systems as Constrained Optimizations



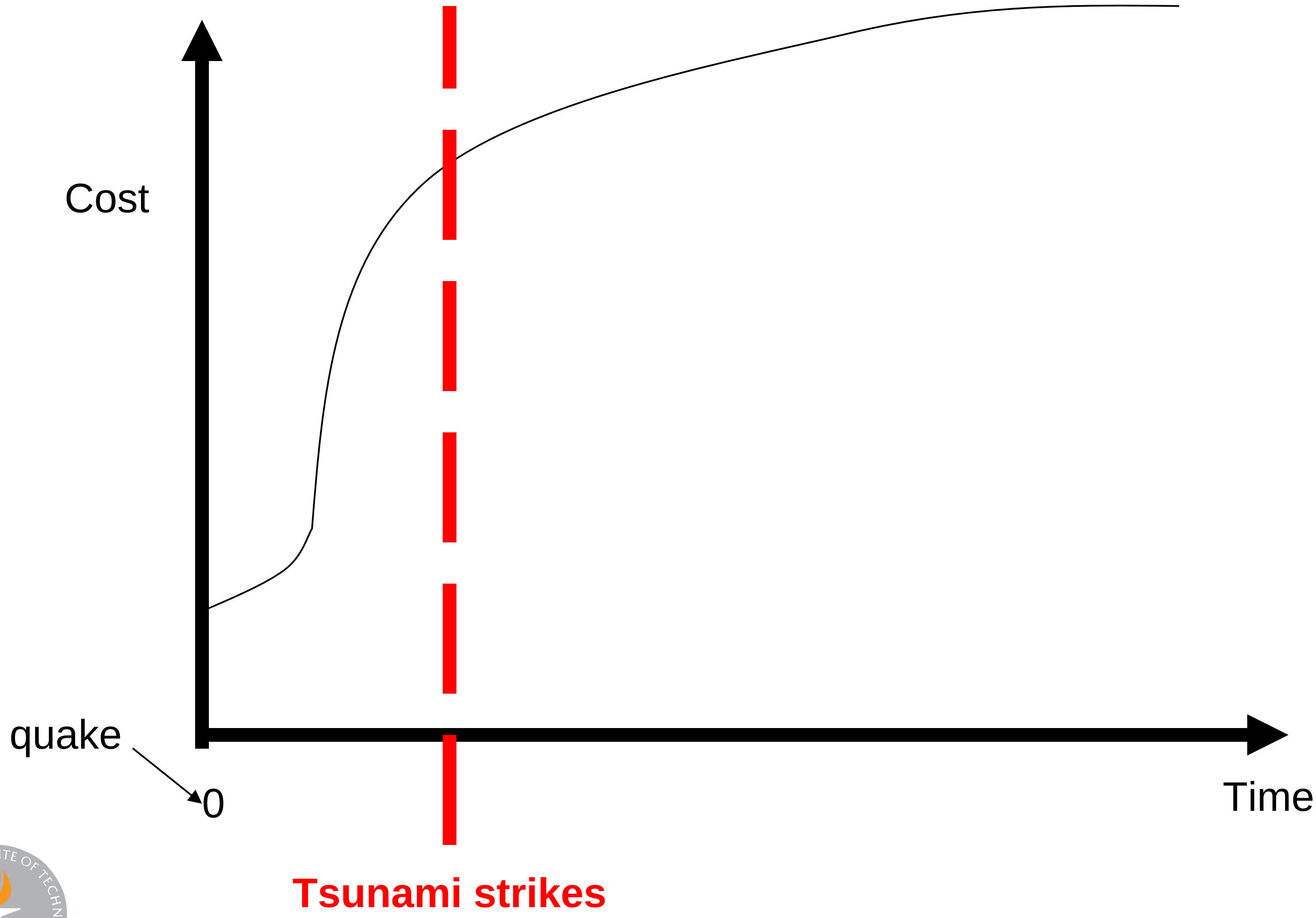
Idea from optimal control

Minimize cost
Subject to constraints

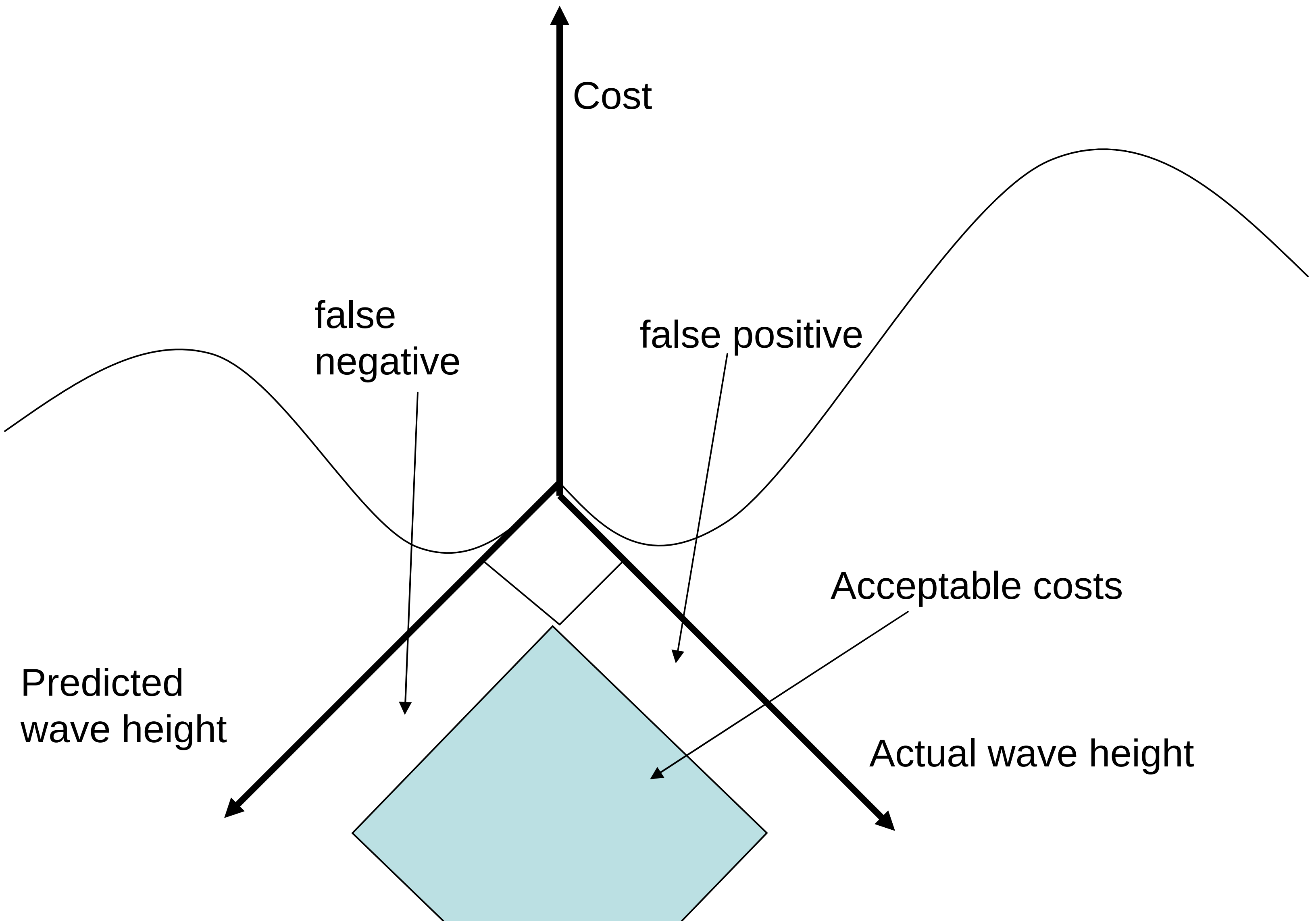
Costs: Tsunami Warning



Cost as a function of time

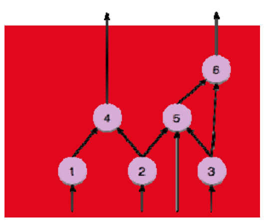


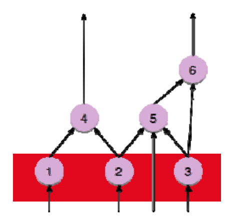
Costs of Error



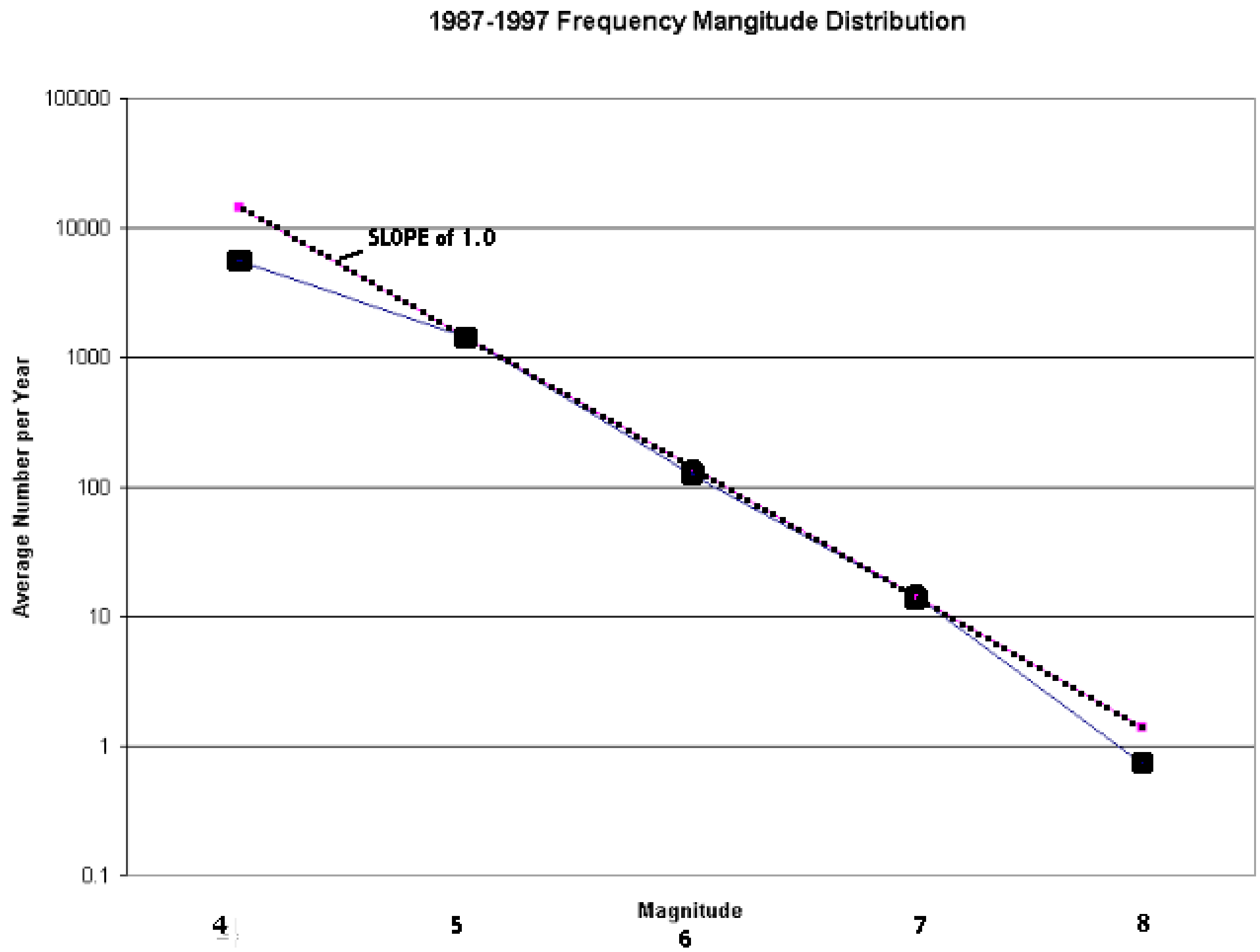
Costs and Benefits of S&R Systems

False Positives: **Occurrence: Occasional** **Cost per event: High**	**False Negatives:** **Occurrence: Very rare** **Cost per event: Enormous**
True Positives: **Occurrence: Rare** **Benefit / event: Enormous**	**True Negatives:** **Occurrence: All the time** **Benefit / event: Low**

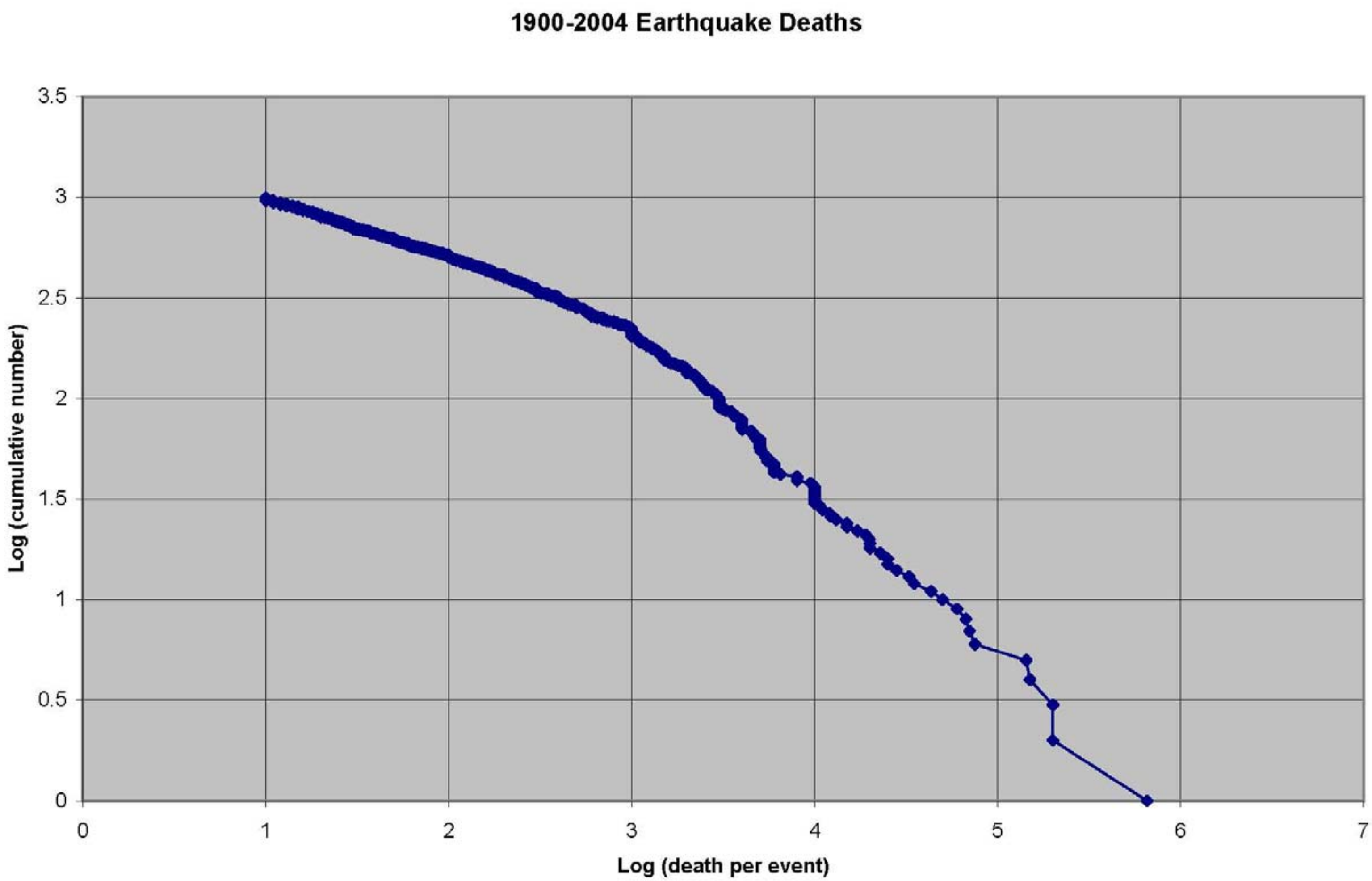




Rarity of Event: Tom Heaton, Caltech

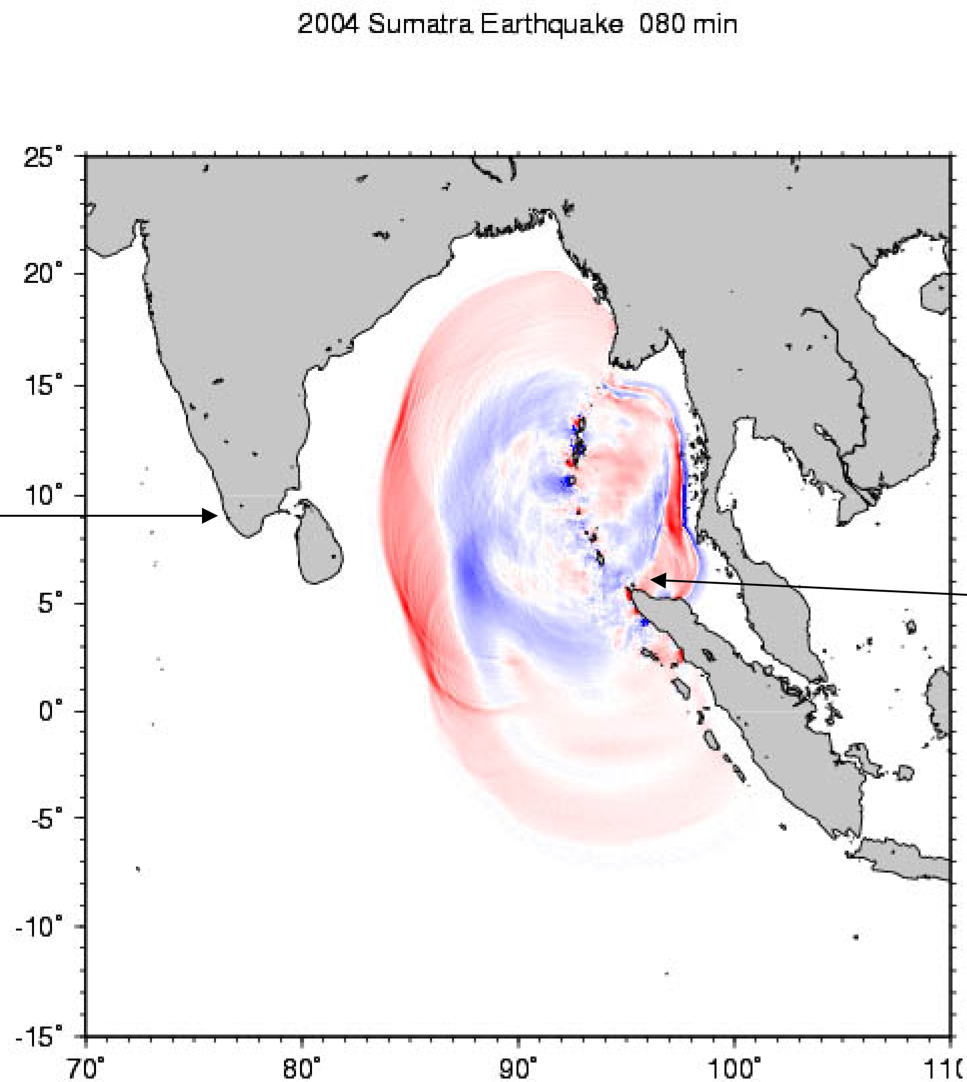


Costs of Events: Tom Heaton, Caltech



$$\left(\text{frequency of occurrences}\right) \propto \left(\text{number of deaths in an event}\right)^{-0.86}$$

Inevitable Costs



Lower cost system

High cost system

Design/ Analysis as Constrained Optimization

Minimize expected amortized cost over lifetime

Subject to constraints about

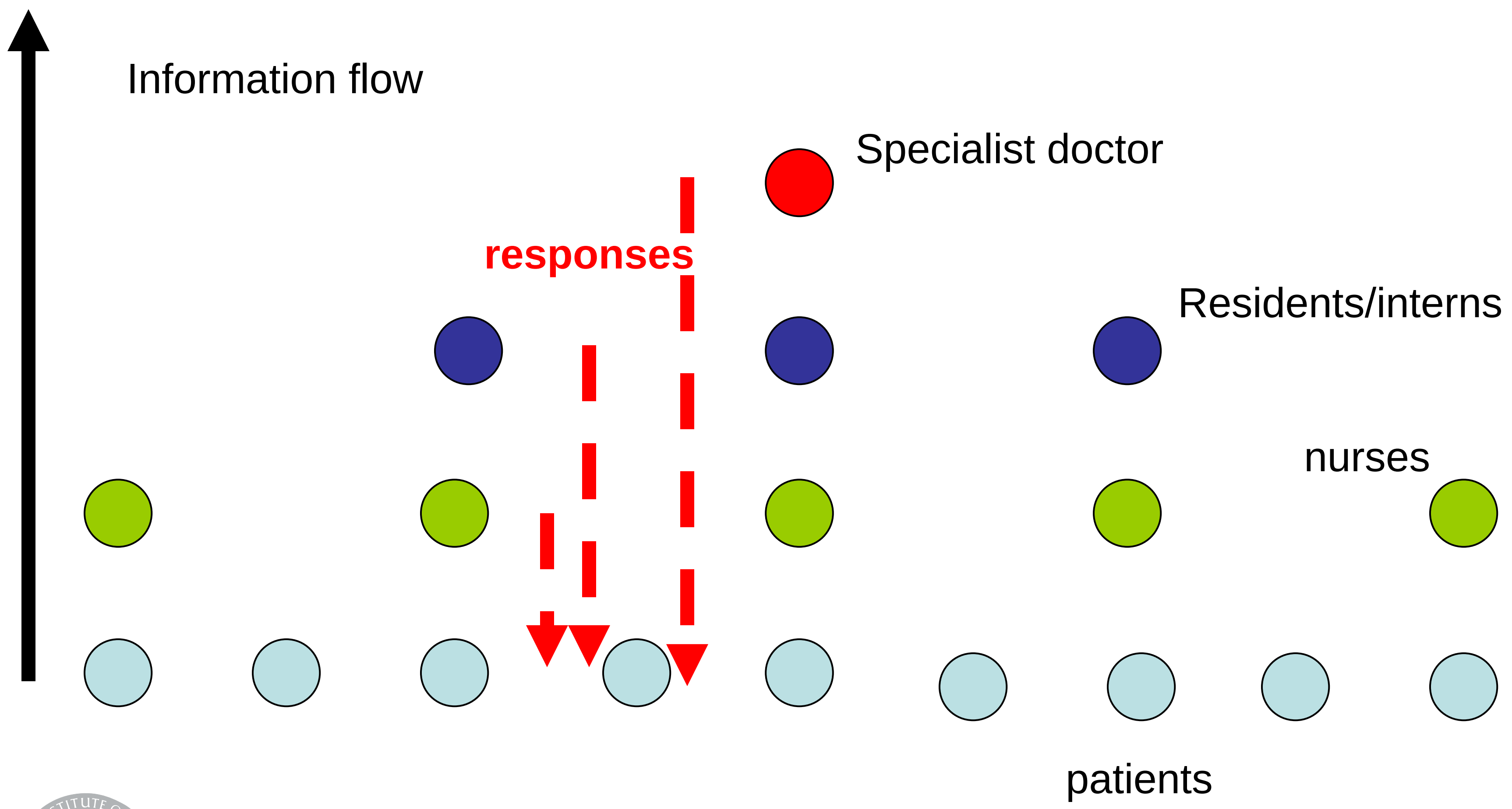
1. Costs of system development
2. Costs of maintenance
3. Types of nodes

Nodes of a System

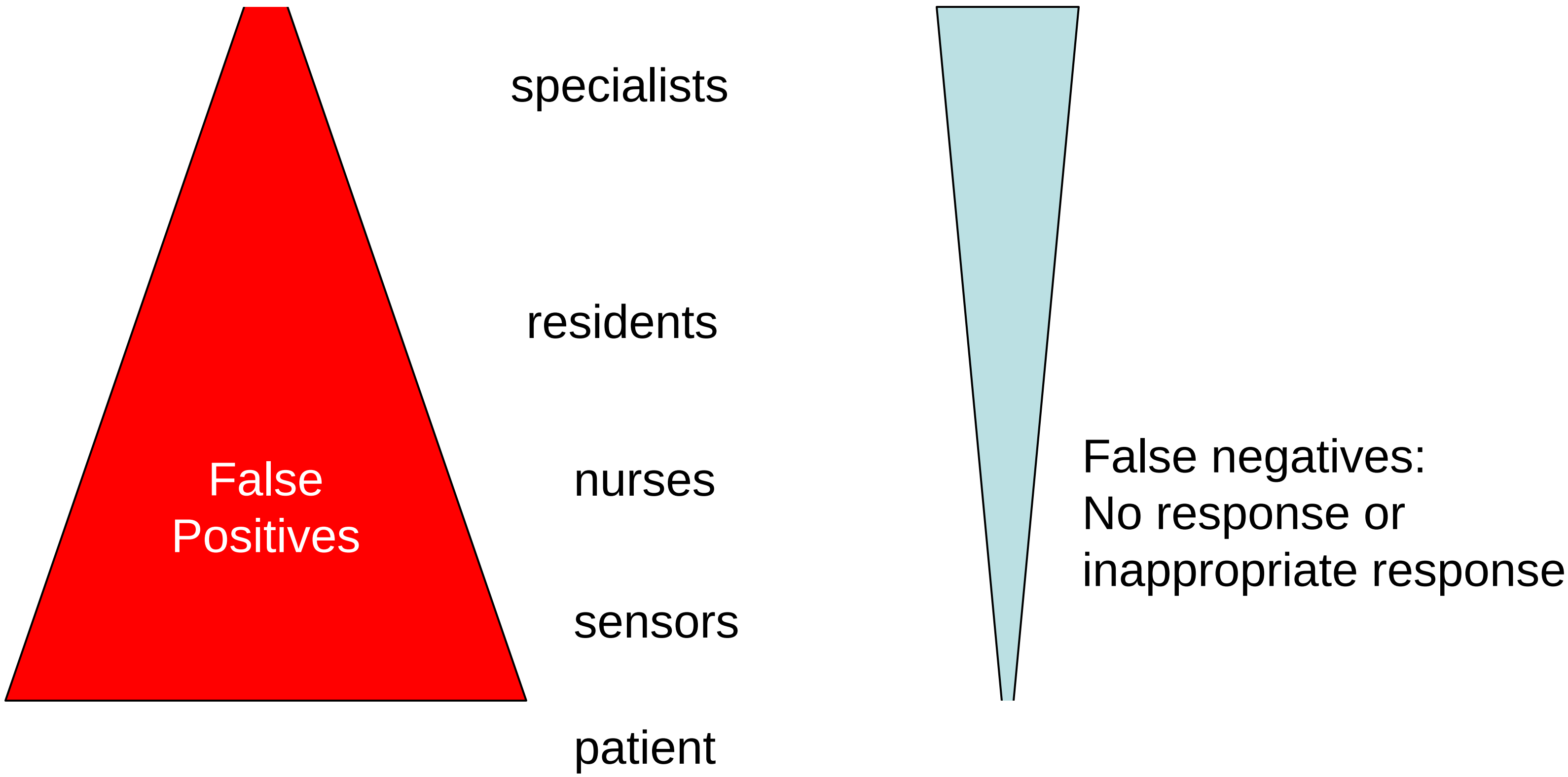
- What are the different node types? What are their costs and capabilities in sensing, responding, computing?

- Consider a hospital with doctors, residents, nurses, volunteers, sensors, patients. What are the capabilities and “costs” of each?

Nodes of a Hospital System



Nodes of a Hospital System



Types of Communication

1. CEO and VP meet Monday at 9AM – Schedule Driven; ***timed***

2. CEO calls VP to check on status of manufacturing – SOA; ***pull***

3. VP manufacturing calls CEO when there is a fire in a factory. – EDA; ***push***

Schedule-Driven Communication: Advantages

- Checks “health” of agents.
  - if an agent doesn’t participate then that agent is probably not functional.
- Energy spent when needed, i.e., when “woken” up by schedule;
  - important for certain types of wireless devices.
- Effective interaction with a group of agents scheduled to meet at the same time

Schedule-Driven Communication Disadvantages

- CEO and VP have group meetings every Monday morning.
- There is a fire in a factory on Monday evening. Does the CEO hear about the fire only next Monday morning?

Push

Advantage

- Service monitors “reality” – data sources – and proactively informs consumer when necessary.

Disadvantage

- Some messages may be irrelevant (false positives) and some conditions may be missed (false negatives)

Pull

Advantage:

- Requestor often has a good idea of information that the requestor needs.
  - e.g., Doctor asks nurse for specific information about blood sugar

Disadvantage

- Requestor doesn’t know when to pull the information.

Ideal: Combination of Communication Types

Example of a hospital:

- **Push**: From sensors to nurses to residents to specialist doctors
- **Pull**: Doctors ask for specific information from specific people and databases to help with determining the doctor's actions
- **Schedule**: Every morning specialists, residents, nurses meet to discuss cases

Database: Interaction Types

- **Push**:
  - Trigger; continuous query
- **Pull**: Typical database operations
  - Query, Updates
- **Schedule**:
  - Time-driven queries and updates

Key Points

- Most systems benefit from a combination of schedule, push and pull
  - what information is communicated with each type of operation, and when?

- Human organizations use combinations of schedule, push and pull. Consider:
  - healthcare, supply chain, trading

- IT has been slow to support integration of ***push*** with pull and schedule-driven interaction

What is an Event?

- An event is a state change that is significant to some agent.
- Significant means the agent should take an action.
- The action may be merely registering the information.
- Passage of time is a state change.
- Absence of change in a parameter over time may be an event.

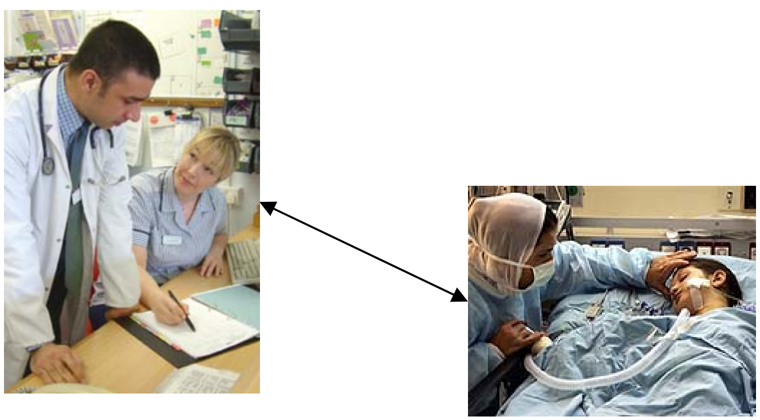
Other Basic Definitions

- **Event Object**: An object containing a description of an event, e.g., a message
- **Latent event objects**: Data from which event objects can be generated
- ***Event objects can be exchanged by push, pull, and scheduled interactions***.

A fundamental problem

Producers and consumers of information do not share state.

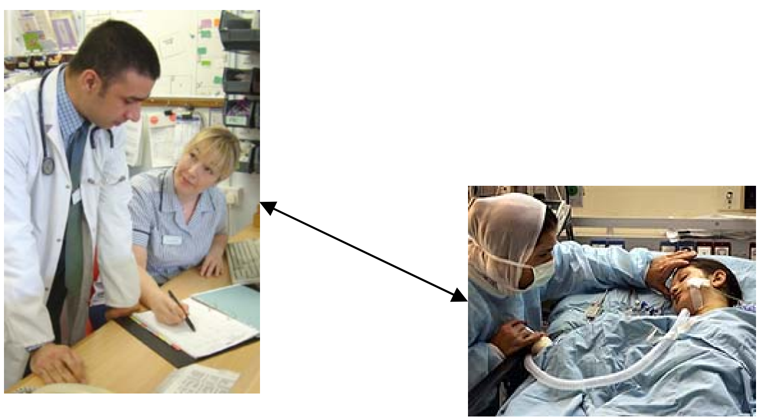
- **Pull**: consumer doesn't know producer's state
- **Push**: producer doesn't know consumer's state.
- **Schedule**: Neither knows state except at scheduled times



A fundamental problem

Producers and consumers of information do not share state.

- Producers and consumers need a shared model of what is a significant state change
- Shared model can be specification of the event or what is normal (the negation of the event)



The Central Issue in Design

- What is the ***value*** of information?

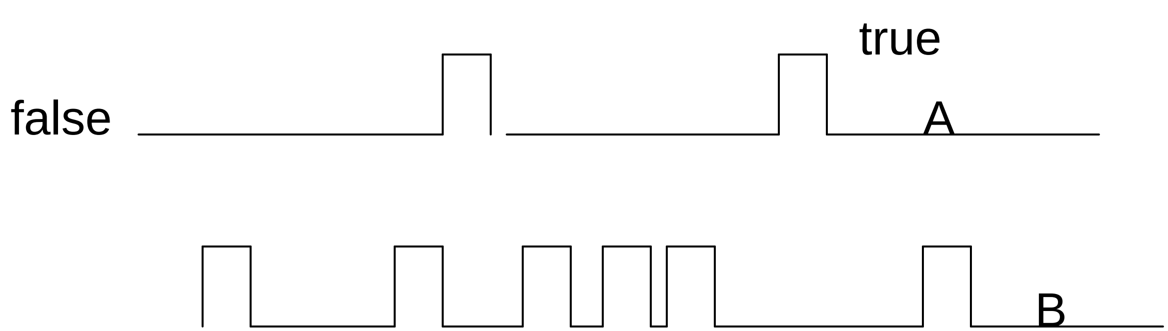
- The value of a piece of information is the difference in costs of responses:
  - Without the information
  - With the information

Example

- Generate an alert when X AND Y becomes true.

- X changes infrequently
- Y changes frequently

- What is the value of information about Y when X is false?



Design / Analysis in terms of Theory

- Push, pull, schedule? When? What information?
  - Depends on value of information and types of nodes

- Query language? Pub/sub mechanism?
  - What is the shared model? Is there a model mismatch? Can the model be trained?
  - Depends on amortizing flow of information across multiple subscribers / producers

Examples: Radiation Detection

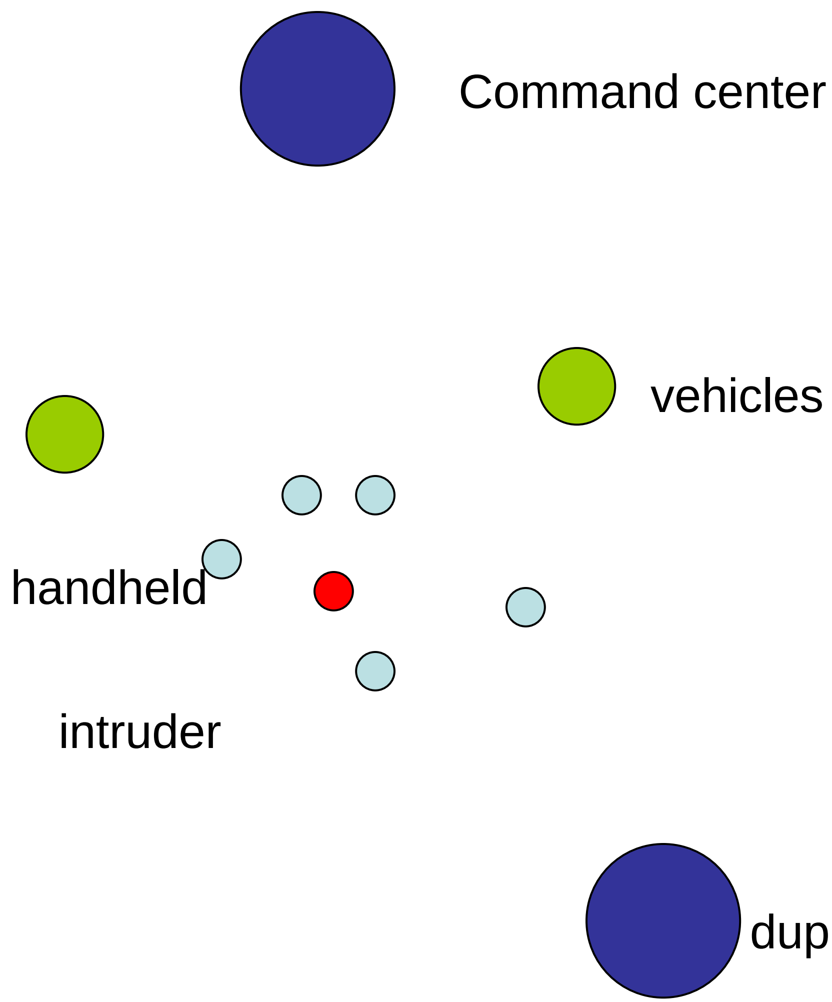
Goal: Intercept terrorist carrying radiation material in a backpack at a political rally.

Nodes of system

- Sensors: detect particles
- Handheld sensors with security personnel; sensors, computers in vehicles; command centers;

- Design the system: network; push/pull/schedule; query language.

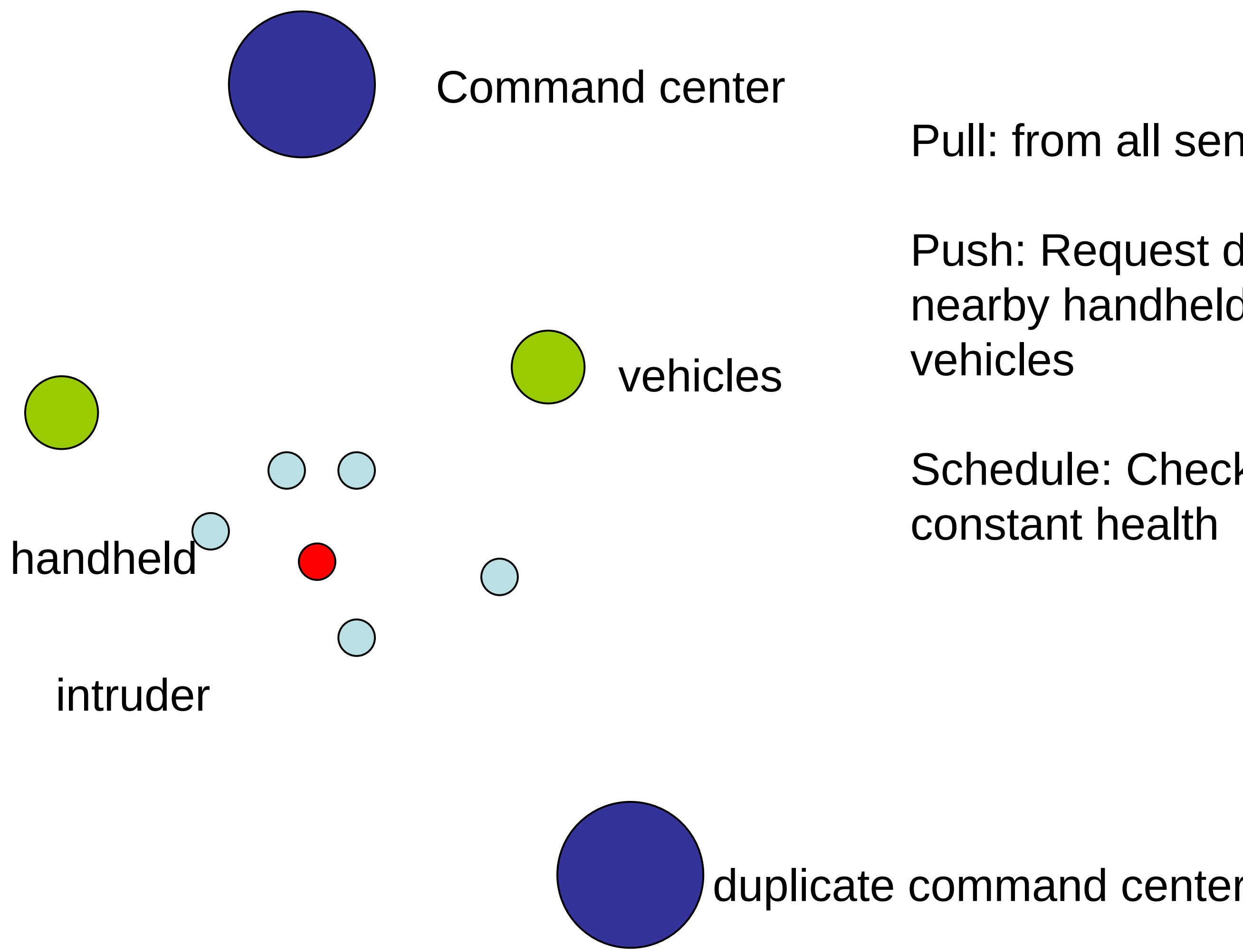
Network Design



Network structure depends on what information gets published.

If publication is --- possible intruder at location x,y --- then send data directly to command center

Component Interaction

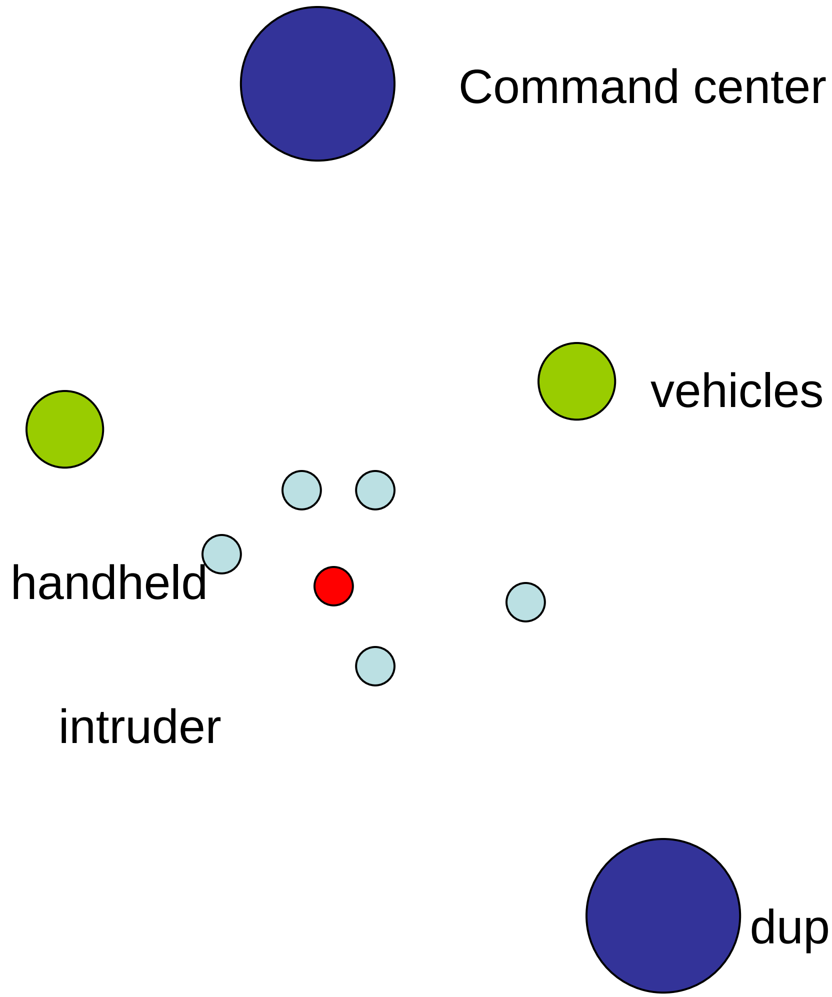


Pull: from all sensors

Push: Request data from nearby handhelds, move vehicles

Schedule: Check constant health

Query Language: Shared Model



Shared model:
- Event is detection of high particle rate
- Event is unusual behavior

- What is the query notation? SQL?

Examples: Baggage Handling at Airports

Goal: Ensure baggage gets routed correctly

Nodes:

- Sensors: RFID readers
- Errors: Multiple RFID tags; multiple vehicles

Design system: network; push/pull/schedule; query language.

Examples: Program Trading

Goal: Trades to maximize profit subject to limited risk

Nodes:

- Stock ticks
- Trader desks

Design system: network; push/pull/schedule; query language.

Design / Analysis in Terms of Theory

- What is the optimum network of an RFID system? A telecom system? A hospital system?
- When is aggregate information optimum?
- When is approximate information optimum?
- What sorts of sampling techniques should be used?
- What information should be stored? Where?

All these questions can be answered systematically from a theoretical framework

Event Processing Theory

- Encompasses time; accuracy; value of appropriate actions; value of information; query planning

- Doesn't deal with transaction management; continuous control, channel capacity.

- Good news/ bad news:
- More complex than relational algebra, optimal control, information theory.
- Uses probability, statistics, optimization, distributed systems, security, model checking, scheduling,..

Summary

Event Processing needs a unifying framework for it to become a "discipline."

The unifying framework should span all EP applications.

There may be more than one framework. Let's work together in building them.