Getting Started

We now have dress shirts on sale for men with 16 necks [at a department store]

Ice cream souveniers [on a billboard on I-24, Nashville]

Thou shall commit adultery [The Wicked Bible, 1623, England]

Slow children at play [in residential areas]

🎉 Please wait for hostess to be seated [at restaurants]

We will sell gasoline to anyone in a glass container [Sania Fe gas station]

Don't kill your wife. [In the window of a Kentucky appliance store]

Let our washing machine do the dirty work.

Dinner Special - Turkey $2.35; Chicken or Beef $2.25;
                 Children $2.00

Will the last person to leave please see that the perpetual light is extinguished [In the vestry of a New England church]

✝️ When two trains approach each other at a crossing, both shall come to a full stop and neither shall start up again until the other is gone. [Kansas legislature, early 1890’s]

Trespassers will be prosecuted to the full extent of the law - Sisters of Mercy [on the wall of a Baltimore estate]

Man, honest. Will take anything [an ad]
Requirements Elicitation

- Critical Issues
- Desirable Properties of Requirements
- Some Elicitation Techniques
  - Ethnomethodology
  - Domain Analysis
  - Problem Frames
  - Data/Information Elicitation Techniques

©Lawrence Chung
What are RE Processes?

A Framework

for initial model construction & subsequent reengineering

3 fundamental concerns:

- understand 
- (formally) describe 
- attain an agreement on

the problem

User

Elicitation

Knowledge

User reqs.

Request more Knowledge

Domain Knowledge

Problem Domain

Specification

Validation

User Feedback

Reg. models

Validation results

Domain Knowledge

Lawrence Chung

- Elicitation
determine what’s really needed, why needed, whom to talk to acquire as much knowledge as possible

- Specification
produce a (formal) RS model: translate “vague” into “concrete”, etc. make various decisions on what & how

- Validation
assure that the RS model satisfies the users’ needs

early RE

late RE
Relativism is everywhere:
Truth often depends on the observer
- The meanings of the words
- The assumptions about context
- The set of categories for understanding the world
- perceptual limitations
- cognitive ability
- personal values and experience

“In the land of the blind, the one-eyed man is king”

- many sources
  - customers, tellers, other employees
  - databases, files, chapters, literature, standards, legal issues
  - surveys of similar applications
  - (what are they, who are they, how successful are they)

- different views on wants and needs

- different notations, mental models

User

Elicitation

Problem Domain
(acct’q, banking, loan policies, etc.)

- many sources
  - domain experts (accountants, bankers, loan managers, etc.)
  - databases, files, chapters, literature, standards, legal issues

©Lawrence Chung
Requirements creeping rate = percentage of change/time

Can you cope with 50%/month?
Requirements Elicitation

- Critical Issues
- Desirable Properties of Requirements
- Some Elicitation Techniques
  - Ethnomethodology
  - Domain Analysis
  - Problem Frames
  - Data/Information Elicitation Techniques
How to do RE?

Major themes of the course

Types of Errors

Jet Propulsion Laboratory (JPL) [Kelly92]
- nearly 2/3 requirements defects are due to omission of key information
  - techniques of completeness
  - setting the scope
  - identifying system operations
  - identifying goals/objectives

The Naval Research Laboratory
- Navy A-7E aircraft’s operational flight program [Boehm, DeMarco, et al.]
  - ongoing research since the mid-70’s
  - 77% of requirements errors were nonclerical
  - 49% incorrect facts
  - 31% omission
  - 13% inconsistency
  - 5% ambiguity
  - 2% misplacement
  - 33% of requirements errors were detected by manual review automated tools can detect a significant number of errors

Error detection and removal

in defining

Why, What & How

& What else?

Lawrence Chung
**How to do RE?**

*Major themes of the course*

- **Error detection and removal**

<table>
<thead>
<tr>
<th>Conceptuality</th>
<th>Formality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural language</td>
<td>Semi-formal notations</td>
</tr>
<tr>
<td>customer/users</td>
<td></td>
</tr>
<tr>
<td>defects</td>
<td></td>
</tr>
</tbody>
</table>

- **A requirements engineer should:**

  - understand the problem *through interaction*
  - express/translate the needs
  - be knowledgeable in relevant technologies toward possible solutions
  - decide on a solution
  - manage acquisition and evolution

A requirements engineer = SEeer, Customer Engineer, Req/Sys/Prog Analyst, Sys. Engineer, Sys. Engineer for Business, Req. Modeller, Assessor of Safety Control & Protection Systems, System Designer, RE Process Manager, Knowledge Engineer, Specifier, etc.

*Lawrence Chung*
Qualities of a Requirement: Unambiguous

- A requirement is unambiguous if it has only one interpretation.
Explore Ambiguity: Dictionary Definitions

*Mary had a little lamb.*

**had** - Past of have

**have** - 1a: To hold in possession as property  
4a: To acquire or get possession of: OBTAIN (best to be had)  
4c: ACCEPT; to have in marriage  
5a: To be marked or characterized by (have red hair)  
10a: To hold in a position of disadvantage or certain defeat  
10b: TRICK, FOOL (been had by a partner)  
12: BEGET, BEAR (have a baby)  
13: To partake of (have dinner)  
14: BRIBE, SUBORN (can be had for a price)

**lamb** - 1a: A young sheep esp. less than one year old or without permanent teeth  
1b: The young of various other animals (as smaller antelopes)  
2a: A person as gentle or weak as a lamb  
2b: DEAR, PET  
2c: A person easily cheated or deceived especially in trading securities  
3a: The flesh of lamb used as food
## Explore Ambiguity: Analysis

<table>
<thead>
<tr>
<th>Have</th>
<th>Lamb</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>1a</td>
<td>Mary owned a little sheep under one year of age or without permanent teeth.</td>
</tr>
<tr>
<td>4a</td>
<td>1a</td>
<td>Mary acquired a little sheep under one year of age or without permanent teeth.</td>
</tr>
<tr>
<td>5a</td>
<td>1a</td>
<td>Mary is the person who owned a little sheep under one year of age or without permanent teeth.</td>
</tr>
<tr>
<td>10a</td>
<td>1a</td>
<td>Mary held a little sheep under one year of age or without permanent teeth in a position of disadvantage.</td>
</tr>
<tr>
<td>10b</td>
<td>1a</td>
<td>Mary tricked a little sheep under one year of age or without permanent teeth.</td>
</tr>
<tr>
<td>12</td>
<td>1b</td>
<td>Mary gave birth to a young antelope.</td>
</tr>
<tr>
<td>12</td>
<td>2a</td>
<td>Mary is (or was) the mother of a particular small, gentle person.</td>
</tr>
<tr>
<td>13</td>
<td>3a</td>
<td>Mary ate a little of the flesh of lamb.</td>
</tr>
<tr>
<td>14</td>
<td>2c</td>
<td>Mary bribed a small person trading in securities who was easily cheated.</td>
</tr>
</tbody>
</table>
Explore Ambiguity: An Observation

The sweet spot

Understandability

Ambiguity
Qualities of Software Requirements


• **Correct**
  – Is a true statement of something the system must do.

• **Complete**
  – Describes all significant requirements of concern to the user.

• **Consistent**
  – Does not conflict with other requirements.

• **Unambiguous**
  – Is subject to one and only one interpretation.

• **Verifiable**
  – Can be tested cost effectively.

• **Ranked for importance and stability**
  – Can be sorted based on customer importance and stability of the requirement itself.

• **Modifiable**
  – Changes do not affect the structure and style of the set.

• **Traceable**
  – The origin of each requirement can be found.

• **Understandable**
  – Comprehended by users and developers.
Qualities of a Requirement: Verifiable

- A requirement is verifiable if:
  - There exists some finite, cost-effective process with which a person or machine can check that the product meets the requirement.

- The system supports up to 1,000 simultaneous users.
- The system shall respond to an arbitrary query in 500 msec.
- The color shall be a pleasing shade of green.
- The system shall be available 24 x 7.
- The system shall export view data in comma-separated format, according to the IEEE specification.

Are these requirements verifiable? If not, what is a better way to state them?
How to do RE?

Major themes of the course

Error detection and removal

<table>
<thead>
<tr>
<th>Conceptuality</th>
<th>Formality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural language</td>
<td>Semi-formal notations</td>
</tr>
<tr>
<td>customer/users</td>
<td>Formalisms</td>
</tr>
<tr>
<td>defects ↑</td>
<td>modellers/specifiers</td>
</tr>
</tbody>
</table>

Clarity

shall, will, must, should, etc. => formality (criticality)
A or B => formality (inclusive-or/exclusive-or)
mostly A => formality (A in cases C1, C2, ..., Cn)
perhaps/could/may/might A => formality (A in cases C1, C2, ..., Cn)
b very and large, often, frequently A => formality (A in cases C1, C2, ..., Cn)

A or ~A => formally ignore as tautology, but may mean something
A set of formulas $\Phi$ in first-order logic is **consistent** if and only if there is no formula $\phi$ such that $\Phi \vdash \phi$ and $\Phi \vdash \neg \phi$.
How to elicit?

How to do RE?
Major themes of the course

Error detection and removal

<table>
<thead>
<tr>
<th>Conceptuality</th>
<th>Formality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural language</td>
<td>Semi-formal notations</td>
</tr>
<tr>
<td>customer/users</td>
<td>Formalisms modellers/specifiers</td>
</tr>
<tr>
<td>defects↑</td>
<td>defects↓</td>
</tr>
</tbody>
</table>

Correctness (external Consistency)

When two trains approach each other at a crossing, both shall come to a full stop and neither shall start up again until the other is gone. [Kansas legislature, early 1890’s]

=> scenario analysis and/or
=> formalism

Call forwarding: (B -> C); (A -> C); (D -> A)
(B -> C); (C -> B)

How would you detect these potential problems?
How should you deal with “feature interaction problems”?
In philosophy, ontology is the study of being or existence. It seeks to describe or posit the basic categories and relationships of being or existence to define entities and types of entities within its framework. Ontology can be said to study conceptions of reality. http://en.wikipedia.org/wiki/Ontology
Recall

How to do RE?
Major themes of the course

Error detection and removal

<table>
<thead>
<tr>
<th>Conceptuality</th>
<th>Formality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural language</td>
<td>Formalisms</td>
</tr>
<tr>
<td>customer/users</td>
<td>modellers/specifiers</td>
</tr>
<tr>
<td>defects↑</td>
<td>defects↓</td>
</tr>
<tr>
<td>Semi-formal notations</td>
<td></td>
</tr>
</tbody>
</table>

Overspecification
- algorithms (sorting, searching, routing, serialization, normalization, etc.)
- data structures (stack, queue, tree, graph, heap, etc.)

Technically unsolvable problems
- scheduling, pattern recognition, etc.

Also, do not worry at this time about acquiring the resources to build the house. Your first priority is to develop detailed plans and specifications. Once I approve these plans, however, I would expect the house to be under roof within 48 hours.

While you are designing this house specifically for me, keep in mind that sooner or later I will have to sell it to someone else. It therefore should have appeal to a wide variety of potential buyers. Please make sure before you finalize the plans that there is a consensus of the population in my area that they like the features this house has.

I advise you to run up and look at my neighbor’s house he constructed last year. We like it a great deal. It has many features that we would also like in our new home, particularly the 75-foot swimming pool. With careful engineering, I believe that you can design this into our new house without impacting the final cost.

©Lawrence Chung
Requirements Elicitation

- Critical Issues
- Desirable Properties of Requirements
- Some Elicitation Techniques
  - Ethnomethodology
  - Domain Analysis
  - Problem Frames
  - Data/Information Elicitation Techniques

©Lawrence Chung
Ethnomethodology

Why is it difficult?

A wicked problem

the "say-do" problem: people know how to do things they normally don't describe (tacit knowledge); descriptions of such things may be highly inaccurate

So, what should be done then?

List three examples
How to elicit?

Data/Information Elicitation Techniques

- the "say-do" problem: people know how to do things they normally don't describe (tacit knowledge); descriptions of such things may be highly inaccurate
- experts may not want to tell... "say-do" problem

Ethnomethodology (People's methods)

- Sometimes, observation is the best way to understand how things are done
- (esp. where) social order is accomplished on a moment-to-moment basis
- So, OBSERVE in a NATURAL setting

  plan ahead (like in interviewing) → observe & record → analyze

- e.g., stock brokerage (multiple phone calls, computer), HCI
- ethical, legal implications, if video-taping without notification
- observation not in a natural setting, if people are aware of being observed needs maximal natural setting, minimal interruption
- can be too time consuming to analyze the recording gradual identification of critical tasks and focusing
How to elicit?

Knowledge Acquisition: A Relative of Requirements Elicitation

- From AI, largely intended for acquiring expertise (e.g., of doctors, lawyers) practised by "knowledge engineer"
  
  Recall: requirements elicitation -> capturing "knowledge" of domain

- Use of mediating representations:
  
  help bridge the gap between the structure of expert knowledge and formal, computer-based representations (e.g., Text, Note, Diagram, Chart, Table, Frame, Rule, Semantic Net)

Automatic KA techniques

- infer new knowledge from past experience
  
  worksFor(bill, john)  
  worksFor(maria, john)  
  worksFor(george, john)  
  
  forall x  worksFor(x, john)  
  
  For whom does Susan work?

- suggest refinement
  
 forall x, y  (x <> y) -> worksFor(x, y)

- detect inconsistencies
  
 forall x  worksFor(x, john)  
  worksFor(eve, maria)

Issues recognized for KA

- novice K <> expert K  --> diff. types of customers

- experts may not want to tell  --> 'say-do' problem

- expertise (experience) doesn't always translate into 'rules'
  
  --> reqs. analyst: informal -> formal (ethnomethodology)
Requirements should contain *nothing but* information about the environment.
How to elicit?
Data/Information Elicitation Techniques

- Sampling
  - the process of systematically selecting representative elements of a population, often applied to documents ("hard" data, e.g., transaction log)
  - useful as it can minimize costs/overhead during data gathering (only a portion, no direct involvement of customer)
  - sampling tasks:
    - data determination
      - E.g., in building/improving an ATM system
        - how much time/transaction (-> #machines, response time improvement)
        - how many errors before completion (-> UI design, robustness, help fn)
        - correlation between amount and time spent (-> max amt, accuracy assurance)
        - peak period, interval between transactions (-> performance improvement)
        - success/failure ratios (-> bad transaction types, time of day/week)
    - population
      - E.g., transactions
        - transactions in 4 local branches for 1 week
    - type determination
      - purposive sampling choose population elements the analyst considers important with no regard to statistical issues (e.g., only high amount/frequent transactions)
      - random sampling every kth element
    - sample size
      - E.g., consider 1/10th of all transactions (in 4 local branches for 1 week)
      - the bigger the size, the higher the cost of sample collection, but higher confidence level
How to elicit?

Data/Information Elicitation Techniques

Questionnaires

- kinds of information sought: attitudes, beliefs, behavior
  - not normally found through sampling (hard data) or interviews
  But if not anonymous, customers may be reluctant to answer questions

  Have you used any meeting scheduler system before? Y N
  - If yes, are you satisfied with it? 1 2 3 4 5
  - If no, would you try a meeting scheduler system when available? Y N
  Would you encourage other people to use one? Y N
  How much time are you willing to spend in each session?
    - 5 minutes < 10 minutes <= 20 minutes

- avoid open questions
  (because answers to such questions are hard to correlate and interpret)
  - Do you think a new meeting scheduler will succeed?
  - Do you believe a meeting scheduler system should drastically change our daily lives?

- questionnaires should be short
  (otherwise, people may be reluctant to participate with busy schedule)

- administer the questionnaire using simple rules
  - scoring scheme: e.g., a range of from 1 to 5
  - group inter-related questions
    - E.g., Q 1 2 3 represent customer satisfaction with current systems
    - Q 4 5 6 7 represent customer willingness to try a new one

Lawrence Chung
How to elicit?

Data/Information Elicitation Techniques

- **Interviewing**

  - **kinds of information sought:** tacit knowledge as well as hard facts, opinions, feelings, goals
  - **dos:** planning ahead of time
    - See how experienced journalists do it!
  - **don'ts:** needs mastery of skills
    - buzzwords/acronyms to impress
    - unusual body language
      - (unusual tone of voice, facial/body expressions, dress, etc.)

- **public library**
  - read background material
    - (mtgs, scheduling mtgs, mtg scheduler systems)
  - establish interviewing objectives
    - (what are you trying to get out of the interview?)

- **magazines, films**
  - decide whom to interview
    - (people are busy; likely or important customers)

- decide for interview
  - call people ahead of time to prepare;
  - tell the purpose, duration, possibly question types

- decide on interview structure & question types
  - (write down questions & answer them ahead of time)

- hold the interview

- review on-line notes -> disseminate -> difference recording & resolution

Lawrence Chung
How to elicit?

Data/Information Elicitation Techniques

- **Group Meetings**
  - focus groups:
    - kind of group interview, often conducted in terms of "stimulus material" (videos, stories)
    - Success depends on the kinds of participants and moderator

- **E.g., Joint Application Development (JAD)**
  - Joint Requirements Planning (JRP)
    - usually for high-level managers;
    - identify and examine business goals, problems, critical success factors, strategic opportunities
  - Joint Application Design (JAD)
    - identify and examine the end users' needs

- **4 tenets of JAD:**
  - **group dynamics**
    - participants (developers, users/customers)
    - leader/moderator/facilitator
    - recorder/scribe
  - **visual aids**
    - E.g., calendars, participants, equipments, locations
  - **organized, rational process**
    - periodic, democratic, conflict accommodating
  - **WYSIWIG documentation approach**
Feature Oriented Domain Analysis (FODA): a process for domain analysis and establishes specific product for later use. Three basic phases:

- **Context Analysis**: defining the extent (or bounds) of a domain for analysis
- **Domain Modeling**: providing a description of the problem space in the domain that is addressed by software (See *Enterprise Modeling*)
- **Architecture Modeling**: creating the software architecture(s) that implement solutions to the problems in the domain

*Note: The architectural modeling phase was initially defined as part of the FODA methodology. However, the process of integrating FODA products with architectural modeling has become part of the domain design activity in the overall concept of Domain Engineering.*

©Lawrence Chung
How to Elicit?: FODA

- **Context Analysis:**
  - **Structure Diagram** - an informal block diagram in which the domain is placed relative to higher-, lower-, and peer-level domains.
  - **Context Diagram** - a data flow diagram showing data flows between a generalized application within the domain and the other entities and abstractions with which it communicates.
Problem Frames


Anti-requirements, abuse frame, security requirements

Figure 1: A threat described by a generic abuse frame diagram.

Figure 2: A security requirement expressed in Problem Frames.
Problem Frames

The key idea in problem frames is *recurring problem types*.

- Different problems share characteristics.
- If you stand back from the details of your current problem, you may recognize it as a known problem.
- Many known types of problems are already solved.
- A *problem frame* represents a "well-known" type of software problem.

Problem frames are a way of representing certain software–related expertise.

- **The nature of expertise**
  - Experts have about 50,000 chunks of knowledge
  - It takes them about 10 years to become experts
  - This is true across many domains
  - Experts recognize these chunks instead of deriving them
  - What passes for insight or intuition is often recognition

- **The idea in problem frames**
  - Frames represent chunks of knowledge
  - Knowing more variations will allow you to recognize more pre-solved problems and apply prior art
  - This is easier and less risky than analyzing from scratch
Basic steps in applying problem frames

1. Break the context into pieces (called *domains*).
2. Identify the shared phenomena (called *interfaces*) among the domains.
3. Represent the domains and their interfaces in a *context diagram*.
4. Add the conditions (called *requirements*) that the software must bring about.

- A context diagram that has been augmented with requirements is called a *problem diagram*.
- A problem diagram that recurs a lot is called a *problem frame*.
A sample problem diagram: editing a “periods & ranges” database

- **System**: PREdit machine
- **Domain**: Medical staff
- **Interface**: Data entry
- **Shared phenomena**: All machine domains are causal.
- **Requirements**: Sending SQL to the database is causal.

**Notes**:
- Arrowed dashed line = constraining domain’s phenomena
- Dashed line = referencing domain’s phenomena

©Lawrence Chung
The different types of domains

- Causal (C) – has predictable relationships among physical phenomena
  - The machine domain, which has a double stripe, is always a causal domain.
- Biddable (B) – physical but unpredictable
  - Humans are the most common biddable domain.
- Lexical (X) – physical representation of data and symbolic phenomena
  - Designed domains, which have a single stripe, are usually lexical domains.

The different types of phenomena

- Causal (C or E) – the events that one domain initiates in order to influence or control another domain
  - Causal phenomena deal with what happens.
- Symbolic (Y) – values, truths, and states
  - Symbolic phenomena are encodings of other phenomena.

Examples of each type of phenomenon

- Causal (C or E)
  - DrawLine(x1, y1, x2, y2), supported by LCD screens
  - MakeElectricalPulse(voltage), supported by pacemakers
  - FireThrust(duration), supported by nuclear missiles
  - InsertRow(), supported by Microsoft Excel
  - ClickHyperlink(-url), supported by humans

- Symbolic (Y)
  - ListOfParagraphs(), supported by Word documents
  - GridOfCells(), supported by spreadsheets
  - LevelOfEncryption(), supported by bank database
There are 5 basic problem frames.

- **Required behavior**
  Control part of the physical world to satisfy a condition.

- **Commanded behavior**
  Control part of the physical world according to operator instructions.

- **Information display**
  Obtain state/behavior information from the physical world and present it as required.

- **Simple workpieces**
  Build a tool to create & edit persistent information objects

- **Transformation**
  Transform information inputs to required outputs
News article, 20 Oct 1992

AMBULANCE CHIEF QUITS AFTER PATIENTS DIE IN COMPUTER CRASH
By Ian MacKinnon and Stephen Goodwin

The Chief executive of the London Ambulance Service resigned yesterday over allegations that up to 20 people may have died because of the collapse of a new computer system controlling emergency calls. Virginia Bottomley, Secretary of State for Health, was forced to announce an external inquiry into the 36 hours over Monday and Tuesday which led to delays of up to three hours in ambulances arriving.

...
London Ambulance Manual System Problems

- identification of the precise location can be time consuming due to often incomplete or inaccurate details from the caller and the consequent need to explore a number of alternatives through the map books;
- the physical movement of paper forms around the Control Room is inefficient;
- maintaining up to date vehicle status and location information from allocators' intuition and reports from ambulances as relayed to and through the radio operators is a slow and laborious process;
- communicating with ambulances via voice is time consuming and, at peak times, can lead to mobilization queues;
- identifying duplicated calls relies on human judgment and memory. This is error prone;
- dealing with call backs is a labor intensive process as it often involves CA's leaving their posts to talk to the allocators;
- identification of special incidents needing a Rapid Response Unit or the helicopter (or a major incident team) relies totally on human judgment.
Critical Requirements

- **Ambulance dispatch functionality**
  - Calls report incidents and other needs for transport
  - An ambulance arrives at the location of an incident promptly; the ambulance may take patient(s) to hospital

- **Other requirements**
  - Timely response without communication overload
  - Resilience to faulty communication
  - Resilience to independent field decisions by personnel
  - Incremental information about incident
  - Efficient use of resources, efficient response

- **System considerations**
  - Incremental deployment
  - Fit with existing system processes

First cut at context and problem

Commanded behavior

- Ambulance Dispatch Machine
  - a: 911 call
  - b: dispatch message
- Calls
  - b: dispatch message
  - c: requests

Ambulance arrives at incident promptly, may take patient to hospital

©Lawrence Chung

Source: Lecture note by Jim Herbsleb, http://conway.isri.cmu.edu/%7ejdh/MethodsF06/lec/probframes/prob-fr-3b.ppt
Recalling steps in problem frame modeling

Basic steps in applying problem frames

1. Break the context into pieces (called *domains*).
2. Identify the shared phenomena (called *interfaces*) among the domains.
3. Represent the domains and their interfaces in a *context diagram*.
4. Add the conditions (called *requirements*) that the software must bring about.

- A context diagram that has been augmented with requirements is called a *problem diagram*.
- A problem diagram that recurs a lot is called a *problem frame*.

©Lawrence Chung

Problem Domains

- **Calls**: telephone calls from the public and doctors
- **Resources**: ambulances, personnel, special equipment
- **But** …
  - Calls do not correspond directly to incidents
  - Detailed knowledge of geography is required to interpret calls and to know which ambulance to send
- **So add domains** …
- **Incidents**: discrete events that require ambulance response
- **Geography**: Streets, addresses, hospital locations, etc
Ambulance Context

- Calls
- Real World Geography
- Incidents
- Resources

a: 911 call
d: {create, update, close} incident
e: geographic facts
b: dispatch message
c: requests

©Lawrence Chung

Ambulance Problem

Calls

Real World Geography

Incidents

Resources

Ambulance Dispatch Machine

Ambulance arrives at incident promptly, may take patient to hospital

a: 911 call
b: dispatch message
c: requests
d: {create, update, close} incident
e: geographic facts

©Lawrence Chung

Call Taking

- **Calls**
- **Incidents**
  - Prioritizes calls
  - Establishes location of incident
  - Combines multiple calls about each incident

- **Workpiece**
- **Real World Geography**
- **Resources**

- **a**: 911 call
- **d**: {create, update, close} incident

Geographic facts

- Calls
- Geography Machine
- Real World Geography
- Geography Model
- Geography is OK
- Incidents
- Resources

Model domain (ch 7)

©Lawrence Chung

Ambulance Dispatch

- Calls

*Commanded behavior*

- Real World Geography

*Actually dispatches ambulances based on incidents and status of resources*

- Incidents d Dispatch b Resources

Ambulance arrives at incident promptly, may take patient to hospital

©Lawrence Chung

Source: Lecture note by Jim Herbsleb, http://conway.isri.cmu.edu/%7ejdh/MethodsF06/lec/probframes/prob-fr-3b.ppt
Combined Ambulance Dispatch

Call Taking

Incidents reflect info in calls

Incidents

Calls

Geography Machine

Real World Geography

Geog is OK

Geography Model

Dispatch

Resources

Note: Incidents is lexical in CallTaking, biddable in Dispatch

Ambulance arrives at incident promptly, may take patient to hospital

Source: Lecture note by Jim Herbsleb, http://conway.isri.cmu.edu/%7ejdh/MethodsF06/lec/probframes/prob-fr-3b.ppt

©Lawrence Chung