Knowledge Capture in the New Millennium: How the Semantic Web Changed Everything

Yolanda Gil
Interactive Knowledge Capture Group

www.isi.edu/ikcap
ai.isi.edu

August 2008
In the beginning…

…there was the Knowledge Acquisition Bottleneck
The Infamous Knowledge Acquisition (KA) Bottleneck
Our Goal

Knowledge-Based System
Our Goal

EXPECT

Knowledge-Based System

expert
EXPECT’s Agenda Interface (circa ’94)

Agenda shows errors
Detailed context of error
Suggested fixes
EXPECT: A User-Centered Framework for Developing Knowledge-Based Systems

**EXPECT**

- Method instantiator
- Domain dependent KBS
- KBS compiler
- Interdependency Model (IM)
- Knowledge-Based System
- Knowledge Base
  - Domain ontologies and factual knowledge
  - Problem solving methods
- KA tools
  - EMeD
  - PSMTool
  - NL Editor

**Ontologies and Method libraries**

- CYC/Sensus Upper
- Plans (PLANET)
- Evaluations and Critiques
- Resources (OZONE)
- Evaluation PSMs

**Instrumentation**
The Essence of Knowledge Capture Research: Finding and Fixing Errors in Knowledge Base

[Mitchell 86]

- Incomplete
  - Missing knowledge

- Incorrect
  - Inaccurate knowledge

- Intractable
  - Computational complexity of search space
The Essence of Knowledge Capture Research: Finding and Fixing Errors in Knowledge Base

- Incomplete
  - Missing knowledge
- Incorrect
  - Inaccurate knowledge
- Intractable
  - Computational complexity of search space

[Mitchell 86]

[Stupid 78]

 ignorant
 stupid
 slow

[Mitchell 86]

[Davies 78]
The Essence of Knowledge Capture Research: Finding and Fixing Errors in Knowledge Base

 incomplete
- Missing knowledge

 incorrect
- Inaccurate knowledge

 intractable
- Computational complexity of search space

 inadequate
- Limited expressive power in k representation language

 immature
- Missing key terms needed to express new ones

 ignorant

 stupid

 slow
KANAL [Kim & Gil 03, 04, 05]

Test Knowledge
Knowledge Analysis for: Bridge-COA-1

Summary of analysis

- Warnings were found in the following steps:
  - the Main-Attack-Task called Object-771
  - the Reserve-Task and Supporting-Attack-Task called Object-774
  - the Main-Attack-Task called Object-772
  - the Supporting-Attack-Task called Object-773

Step: the Fire-Support-Task and Attack-by-Fire-Artillery on Mvr Unit called Object-659

- For agent B3rdDSArtBu (Direct-Support-Artillery-Battalions), (Allegiance is Blue), its equipment is (_38-Howitz-155mm5138) so the default combat power is 1.02.
- Since its remaining strength is 0.85499996 the relative combat power is (1.02 * 0.85499996) = 0.87299994
- For enemy B1stBnBde (Mechanized-Battalions), (Allegiance is Red), its equipment is (_RBT-6051650) so the default combat power is 0.41.
- Since its remaining strength is 0.64000005 the relative combat power is (0.41 * 0.64000005) = 0.26240003
- So the available force ratio of the Fire-Support-Task and Attack-by-Fire-Artillery-on-Mvr-Unit called Object-659 it (sum of agent relative-combat-power) (sum of enemy relative-combat-power) = sum of (0.87299994) sum of (0.26240003) = 3.335512 The required force ratio of the task is 0.5

1. The available-force-ratio (3.335512) >= The required-force-ratio (0.5)

- Click [here] to see other checks

The above condition(s) succeeded

The following becomes false:

1. the remaining strength of B3rdDSArtBu is 0.85499996
2. the remaining strength of R1stBnBde is 0.64000005

The following becomes true:

1. the remaining strength of B3rdDSArtBu is 0.83935
2. the remaining strength of R1stBnBde is 0.512
Great Progress

Users can complete complex tasks only with acquisition tools.

Tools help whatever the level of sophistication of user.

Users enter knowledge faster, especially if it is complex.
Great Progress

- Unsophisticated users can enter knowledge about complex processes in expert-level domains (biology, military planning, etc.)
- … Only the knowledge is 99% correct
- … Only the system needs the remaining 1% to reason appropriately

> Many research challenges remain

SHAKEN [Clark et al 05]
Then...

...along came the (semantic) web
Semantic Web
<table>
<thead>
<tr>
<th>Year</th>
<th>Research (etc., etc!)</th>
<th>Common Formats</th>
<th>Web Standards</th>
<th>Wide deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>PCA, etc</td>
<td>cwm, KIF</td>
<td>logic framework, crypto</td>
<td>Trusted systems, Inter-engine interop</td>
</tr>
<tr>
<td>2003/1</td>
<td>etc</td>
<td>n3, RuleML</td>
<td>rules...</td>
<td>Declarative ebusiness, Rule indexes</td>
</tr>
<tr>
<td></td>
<td>cyc, KR, DL</td>
<td>DAML, OIL</td>
<td>OWL</td>
<td>Web of meaning, Cross-App interop</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MCF</td>
<td>RDF-S, RDF</td>
<td>Interop within App</td>
</tr>
<tr>
<td></td>
<td>GML, troff</td>
<td>SGML</td>
<td>XML</td>
<td></td>
</tr>
</tbody>
</table>
novaspiwww.typepad.com -- Nova Spivak, Radar Networks
What Changed in the New Millennium

Knowledge Systems
- One user
- User must be trained
- One system
- One knowledge base
- One reasoner
- Expert user

Semantic Web
- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases
- Many reasoners
- Varying quality and coverage
Knowledge Capture in the New Millennium

Semantic Web

- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases
- Many reasoners
- Varying quality and coverage

Research Topics
Learner: Acquiring Knowledge from Volunteer Contributors [Chklovski 02, 03]

Volunteers are numerous and willing to put in the time, BUT:
- Owe us nothing & are free to leave
- Are unwilling to undergo extensive training upfront
- Vary widely in expectations, helpfulness, knowledge, views, …
Collecting from Volunteers: Research Questions

- Managing volunteer effort:
  - How to draw in volunteer contributors?
  - How to minimize/eliminate contributor training?
  - How to set up targeted collection efforts which focus acquisition?

- Towards organizing knowledge repositories:
  - How to ensure acquired knowledge is easier to interpret?
  - How to best structure the acquired knowledge?

- How to integrate knowledge acquired from contributors with knowledge mined from the web or engineered knowledge bases?

- Using the collected knowledge:
  - How to validate and refine the acquired and available knowledge?
  - How to best use the knowledge repository for practical applications?
Quantitative Analysis of Knowledge Repository

[Chklovski & Gil AAAI’05]

- **Coverage**
  - Spontaneous contributions misallocate contributor effort: some statements come with very high redundancy
  - As the collection grows, there are diminishing returns in the coverage

- **Acceptability**
  - Redundancy of a statement is a useful indicator of its acceptability
  - Typical statements generated by many users (high gf), may not be of better acceptability
  - Quality varies across statements. Evaluation and revision are needed
    - Recent results on validation by other users: with 2 users can identify ~80-95% of spurious entries, detect ~70-80% needing qualification, and reject only ~3% of good entries
Evolving Collection Strategies in Learner

[Chklovski & Gil K-CAP’05]

Carefully designed templates to constrain semantics of the input (1)

Guidance on form and type of answer sought (2)

Knowledge is acquired incrementally, using follow-up questions (3)

Knowledge automatically postprocessed to discard malformed entities (4)

Feedback on whether answer is useful and conforms to guidance given (2)

Multiple contributors evaluate previously entered statements (5)
Addressing Brittleness of Knowledge Bases in Real-World Environments

Problem: missing general common knowledge
- Impossible to fully engineer CALO’s knowledge base in advance (i.e., define all terms and specify all relations)
  - Especially in broad-domain, dynamic environments
- Some missing knowledge needs to come from the user
  - Interactive knowledge acquisition tools like Tailor can be used, but:
  - Busy users should not need to define widely known terms or commonly known relations between them

Approach: Draw from a broad-coverage repository to relate new terms to existing terms in its knowledge base
- CALO can hypothesize appropriate knowledge base extensions
- Broad-coverage knowledge repository can be acquired from volunteer contributors over the web
  - Collection can focus on specific topics of interest (e.g. office environments), specific types of knowledge, etc
  - Acquisition system can proactively guide contributors based on what it already knows
  - Repository extensions can be steered for currency/breadth/density
BEAM Knowledge Sources for NL Interpretation and Assistance

From Web Volunteers
- Action Paraphrases Repository: “plan X ⇔ schedule X” “lease car ⇔ rent car”
- Substeps Repository: “Reserve X has-substep find X”
- Repairs Repository: “If projector not working, try a new bulb”

From Text Extraction
- Verb Relations Repository: “schedule happens-before reschedule”
- Shadow Ontology “CPK is-a restaurant”

From Knowledge Engineers
- Ontologies

From Volunteers in the Organization
- Organization-specific Task knowledge
  “Airport pickup of visitors is common, but not here”

User To Do Entries
- Find hotel w/ pool for Joe
- Reserve conf room for talk
- Buy bread on the way home
- Get talk abstract from Joe
- Announce room for talk
- Set airport pickup for Joe

Added by user
Marked by BEAM as user-only
Executed & monitored
Added by BEAM
Deleted by BEAM because unnecessary

Year 3
Year 4
Year 5

BEAM
- Map entries to task procedures
- Initiate and report execution
- Anticipate missing entries and suggest them
- Group and organize entries
- Assist with how activities are done in the organization

CALO Task Ontology: Catalog of Automated Procedures
- Reserve accommodations
- Host a visitor
- Arrange meeting

Execution and Monitoring
Instrumented Desktop
Task Learning

UCS INFORMATION SCIENCES INSTITUTE

Yolanda Gil
Learning Common Knowledge from Web Volunteers To Assist Users with To Do List Management in CALO

Result: Interconnected structured statements
- setting up a videoconference
  - camera: switch on
  - computer: turn on, start up, link up
  - microphone: turn on, test, adjust

Problems & remedies:
- during a meeting
  - projector not available
  - locate a portable projector

Answering Descriptive Questions:
- What subtasks may be required for the task of setting up a meeting in room N?
  - locating a portable projector

Research Challenges:
- Structure and formalize contributions by combining knowledge acquisition and language processing techniques
- Develop inference algorithms to exploit semi-structured knowledge

Results to Date:
- BEAM system uses collected knowledge to assist users with To Do lists:
  - Learned paraphrase knowledge is currently being used to map ToDo list entries into CALO task ontology

700,000+ statements collected from over 3,000 users
Beamer: Interpreting To-Do Lists Through Paraphrasing [Gil & Ratnakar AAAI’08; Gil & Ratnakar IUI’08]

First system developed to assist users with to-do lists
- To Do lists are pervasive [Kirsh 01; Norman 91], used by more than 60% of people for personal information [Jones & Thomas 97], used more than calendars, contact lists, etc.)
- Many commercial/web tools: tadalist.com, voo2do.com,

BEAMER learns new paraphrase patterns of agent capabilities in CALO Task Ontology when user tasks agents by hand
- E.g., for to-do entry “Set up discussion with Bill on IUI paper” the user tasks calendar agent with task: ScheduleMeeting +person=“Bill” +topic=“IUI paper”
  => Paraphrase pattern learned: Set up discussion with +person on +topic
- Could acquire paraphrases from CALO or web volunteers [Chklovski & Gil, AAAI’05; Chklovski, K-CAP’05]

Uses learned paraphrase patterns to interpret to-do list entries

Evaluation shows great performance
- Collected corpus of 2400 to-do entries from CALO users
- 86.7% accuracy in the relevance task, over 90% correctness in task selection from early stages of learning, and very high correctness in task association so that only 0.2 to 0.4 edits are required from a user for a given to-do entry

Benefits:
- Can handle unusual structure of to-do lists
- Can improve system performance and adapt to user by learning patterns
- Can be applied to any domain or task set (in principle)
Knowledge Capture in the New Millennium

Semantic Web

- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases
- Many reasoners
- Varying quality and coverage

Research Topics

- Harnessing volunteer contributions
To setup for a game of solitaire, take a deck of cards and deal seven cards. Layout the cards horizontally on the table with the first card face up. Using one less card each time, repeat dealing and laying out the cards.
What Makes Good and Bad Instruction?

1. begin lesson
2. start SetupKlondikeSolitaire hasId 8887
   // To setup for a game of solitaire,
3. resultIs name=solitaireGameSetup isa type=GameSetup
   // take a deck of cards and
4. initSetup name=deck isa type=CardDeck
   // deal seven cards.
5. doThis name=Deal basedOn deck, 7 expect name=hand
6. name=hand isa type=Hand
   // Layout the cards horizontally on the table with the first card face up
7. doThis name=Layout basedOn hand
   // Using one less card each time, repeat dealing and laying out the cards
8. name=numOfCards isa type=Integer
9. name=numOfCards value=7
10. repeat
    doThis name=Decrement basedOn numOfCards
        expect numOfCards
    doThis name=Deal basedOn deck, numOfCards
        expect hand
    name=hand isa type=Hand
    doThis name=Layout basedOn hand
    until
    name=Equals basedOn numOfCards, 1
11. end lesson
Results: Solitaire

OWL Representation

<j:0:Procedure rdf:about="#SetupKlondikeSolitaire">
  <j:0:hasSteps rdf:parseType="Collection">
    <j:0:Procedure rdf:about="#Deal"/>
    <j:0:Procedure rdf:about="#Layout"/>
    <j:0:Loop>
      <j:0:until rdf:resource="#Equals"/>
      <j:0:repeat rdf:parseType="Collection">
        <j:0:Procedure rdf:about="#Decrement"/>
        <j:0:Procedure rdf:about="#Deal"/>
        <j:0:Procedure rdf:about="#Layout"/>
      </j:0:repeat>
    </j:0:Loop>
  </j:0:hasSteps>
  <j:0:hasInput rdf:resource="#deck"/>
  <j:0:hasResult rdf:resource="#solitaireGameSetup"/>
</j:0:Procedure>

<rdf:Description rdf:about="#Deal">
  <j:1:hasResult rdf:resource="#hand"/>
  <rdf:type rdf:resource="#Procedure"/>
  <j:1:hasInput rdf:resource="#deck"/>
  <j:1:hasInput rdf:resource="#numOfCards"/>
</rdf:Description>

<rdf:Description rdf:about="../models/#Equals">
  <rdf:type rdf:resource="#Procedure"/>
  <j:1:hasInput rdf:resource="../models/#numOfCards"/>
  <j:1:hasInput rdf:resource="../models/#1"/>
</rdf:Description>

<rdf:Description rdf:about="#hand">
  <rdf:type rdf:resource="#Hand"/>
  <rdf:type rdf:resource="#Entity"/>
</rdf:Description>

<rdf:Description rdf:about="#deck"/>
Results: Solitaire

OWL Representation

```xml
<j:0:Procedure rdf:about="#SetupKlondikeSolitaire">
  <j:0:hasSteps rdf:parseType="Collection">
    <j:0:Procedure rdf:about="#Deal"/>
    <j:0:Procedure rdf:about="#Layout"/>
    <j:0:Loop>
      <j:0:until rdf:resource="#Equals"/>
      <j:0:repeat rdf:parseType="Collection">
        <j:0:Procedure rdf:about="#Decrement"/>
        <j:0:Procedure rdf:about="#Deal"/>
        <j:0:Procedure rdf:about="#Layout"/>
      </j:0:repeat>
    </j:0:Loop>
    <j:0:hasInput rdf:resource="#deck"/>
    <j:0:hasResult rdf:resource="#solitaireGameSetup"/>
  </j:0:hasSteps>
</j:0:Procedure>

 rz:Deal { 
  <j:1:hasResult rdf:resource="#hand"/>
  <rdf:type rdf:resource="#Procedure"/>
  <j:1:hasInput rdf:resource="#deck"/>
  <j:1:hasInput rdf:resource="#numOfCards"/>
}

rz:Equals { 
  <rdf:type rdf:resource="#Procedure"/>
  <j:1:hasInput rdf:resource="../models/#numOfCards"/>
  <j:1:hasInput rdf:resource="../models/#1"/>
}

rz:hand { 
  <rdf:type rdf:resource="#Hand"/>
  <rdf:type rdf:resource="#Entity"/>
}
```

Process Learned

\[
\text{S:SetupKlondikeSolitaire (Deck ?deck)} \\
\{
  \text{while (NotEquals(_numOfCards, 1))} \\
  \{
    \text{A:Decrement ();} \\
    \text{A:Deal (?deck, _numOfCards);} \\
    \text{S:Layout (_hand);} \\
  \}
\}
\]
Knowledge Capture in the New Millennium

Semantic Web

- Many, many, MANY users
- No time for training
- Many distributed subsystems

Research Topics

- Harnessing volunteer contributions
- Natural tutorial instruction
- Many knowledge bases
- Many reasoners
- Varying quality and coverage
The learning process involves complex decision strategies and multi-threaded and finer-grained activities.

Learning goals and decision strategies:
- Increase capability (k of rules and exceptions)
- Find hypotheses about all actions
- Select among alternative hypotheses based on capability
- Cumulate planning and execution capability estimates
- Increase confidence
- Test on a variety of situations
- Increase coverage
- Experiment with problems within a class
- Increase competence
- Explore solution variants
Maven: A Learning Meta-Reasoner [Kim & Gil 08]

**Intention:** Intended learning goals

**Plans**
- Plans for Achieving Learning Goals
- Strategies for Goal Selection
- Strategies for Progress Assessment

**Reasoner**

**Desire**

**Maven**

**Learning History**
History of tasks, activities, and goal achievement

**Top-down**

**Agent Tasks**

**Blackboard**

**Shared Belief:** Knowledge Base
- Hypotheses
  - e.g. Hyp-A: step1 before step2
  - Hyp-B: step2 before step1
- Annotations on hypotheses
  - (e.g. Hyp-A supersedes Hyp-B)

**Bottom-up**

**Learning Goals posted by Maven**

**Learning Goals posted by Learning Agents**
Knowledge Capture in the New Millennium

**Semantic Web**
- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases

**Research Topics**
- Harnessing volunteer contributions
- Natural tutorial instruction
- Meta-level reasoning
- Many reasoners
- Varying quality and coverage
Scaling and Integrating AI Algorithms through Distributed Workflows

- Many scientific disciplines benefit from large-scale processing
  - Computational workflows to manage large complex distributed applications [Deelman & Gil 06; Deelman et al 05; Deelman et al 06; Gil & Deelman 06]
- In contrast, AI research is largely done in small scale
  - Typically confined to desktop computations with modest data sizes

Multi-disciplinary experiments

High-level workflow

Executable workflow

Model integration leads to new discoveries

Shared, large-scale resources

A satellite-generated Interferometric Synthetic Radar (InSAR) image of the 1999 Hector Mine earthquake.

- Shows the displacement field in the direction of radar imaging
- Each fringe (e.g., from red to red) corresponds to a few centimeters of displacement.
Data Analysis and Knowledge Discovery through Distributed Workflows

- Focus on data mining and knowledge discovery algorithms
  - Support large-scale machine learning
  - Complex analysis through multiple algorithms
  - Provide community with high-end computing resources

TECHNICAL CHALLENGES:

- Analysis composed of multiple algorithms
  - Reuse of known-to-work compositions of algorithms as workflow templates

- Algorithm choices based on data
  - Select appropriate algorithms for the given data set characteristics to create workflow instances
  - Integrate standard data transformation steps

- Manage distributed execution
  - Map to resources to create executable workflows

- Scale, robustness
  - Concurrent execution of many analysis workflows on allocated resources
A Workflow for Ensemble Learning [Caruana et al 07]

- Data
  - k
  - Partition
    - k subsamples
    - SelectValidationData
      - k validation sets
      - NaiveBayesModeler
        - Calibration
          - NaiveBayesClassifier
            - k results
            - PerformanceEvaluation
              - k training sets of k-1 subsamples
              - ID3
              - SVM
              - SelectionAndCombination
                - Results
Weather-SM-2007-Data.csv

SamplingIntervalParameterVar

20

maxJavaHeapSizeModelerParameterVar

maxJavaHeapSizeClassifierParameterVar

SystematicSample

TrainingSampleDataVariable

TrainingDataVariable

SamplingNode

ModelerNode

ID3-Modeler

ClassIndexParameterVar

5

ModelDataVariable

ClassificationDataVariable

SamplingIntervalParameterVar

20

maxJavaHeapSizeModelerParameterVar

maxJavaHeapSizeClassifierParameterVar

ID3-Classifier

ClassificationDataVariable

TestIdDataVariable

ModelDataVariable

Legend

node

Link

Variable

component

ArgumentID

Data object
Workflows and Requests in Wings

Reusable workflow template includes constraints

Request describes a partial workflow that is automatically completed by the system
An Architecture with Distributed Workflow Reasoners [Gil et al 08]
Workflow System Architecture

**Workflow System**

- **Query Manager**
  - Request as Template
  - Request Submission
  - Query Portfolios
  - Request Elaboration

**Workflow Generation & Execution**

1. **Step 1:** Template Selection
2. **Step 2:** Algorithm Selection
3. **Step 3:** Data Selection
4. **Step 4:** Workflow Instantiation
5. **Step 5:** Performance Evaluation & Ranking
6. **Step 6:** Workflow Mapping & Execution
7. **Step 7:** Workflow Mapping
8. **Step 8:** Workflow Execution

**Data & Algorithm Selection**

- Step 1: Template Selection
- Step 2: Algorithm Selection
- Step 3: Data Selection
- Step 4: Workflow Instantiation

**Workflow Evaluation & Ranking**

- Step 5: Performance Evaluation
- Data Reuse

**Workflow Mapping & Execution**

- Step 6: Workflow Ranking
- Ingestion & Accuracy

**Failure Handling & Reporting**

- Failure Analysis

**Active Workflows**

**Queue**

**Workflow Generation Processes**

**Ensemble Manager**

**Workflow System Instru-**

- **Step 1-3 Alt-A:** Creation of Template Variants
- **Step 1-3 Alt-B:** First-principles Workflow Assembly

**User Interface Services**

- **Basic Request & Tracking Interface**
- **Full Editing & Advanced Tracking Interface**

**Expert Interface**

**End User Interface**

**Workflow System Logs**

**Execution Logs**

**Other svcs**

**Learning From Human Feedback**

**Autonomous Learning**

**Experiential Memory**

**Site Cat**

**Pattern Cat**

**Data Cat**

**Comp Cat**

**Data Services**

**Comp Services**

**Grid**

**Software Deployment Support**
Workflow System

**LEGEND:**
- All software is open source
- National Middleware Infrastructure (NMI) software

**Wings**
- Workflow validation
- Data/Comp selection
- Workflow generation
- Metadata generation

**Pegasus**
- Site selection
- Replica selection
- Workflow optimization
- Workflow submission

**Condor**
- DAGMan execution engine
- Condor-G job manager

**Globus**
- RLS replica mgmt
- GRAM remote submission
- GridFTP data transfer
Workflows for Seismic Hazard Analysis

[Gil et al 06; Kim et al 06; Gil et al 07]

- Input data: a site and an earthquake forecast model
  - thousands of possible fault ruptures and rupture variations, each a file, unevenly distributed
  - ~110,000 rupture variations to be simulated for a given site

- High-level template combines 11 application codes

- 8048 application nodes in the workflow instance generated by Wings

- 24,135 nodes in the executable workflow generated by Pegasus, including:
  - data stage-in jobs, data stage-out jobs, data registration jobs

- Executed in USC HPCC cluster, 1820 nodes w/dual processors but only <144 available
  - Including MPI jobs, each runs on hundreds of processors for 25-33 hours
  - Runtime was 1.9 CPU years

- Provenance records kept throughout the generation and execution process for 100,000 workflow data products
SCEC workflows run each week using Pegasus and DAGMan on the TeraGrid and USC resources. Cumulatively, the workflows consisted of over half a million tasks and used over 2.5 CPU Years, Largest workflow $O(100,000)$ nodes

Knowledge Capture in the New Millennium

Semantic Web

- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases
- Many reasoners
- Varying quality and coverage

Research Topics

- Harnessing volunteer contributions
- Natural tutorial instruction
- Meta-level reasoning
- OWL and semantic web languages
- Varying quality and coverage
Reasoning to Create Workflow Instance

Wings DAX Generator

- Data Selection for Workflow Instance Creation
- Workflow Instance (Compact)
- Template Data Instantiation
- Workflow Instance (Expanded)
- Workflow Node Unrolling For DAX Generation

- DAX

OWL-DL Reasoner (Jena OWL Micro Reasoner)

Workflow Templates Library

No-inference Reasoner (Jena OWL Mem Writer)

Metadata Catalog (Data/File Descriptions)

Domain Component Library + Metadata Definitions

Core Component + DataSet Ontologies

Process Flow

Ontology Import

Knowledge Capture in the New Millennium

Semantic Web

- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases
- Many reasoners
- Varying quality and coverage

Research Topics

- Harnessing volunteer contributions
- Natural tutorial instruction
- Meta-level reasoning
- OWL and semantic web languages
- Semantic web reasoners
Annotating Knowledge Bases with Provenance to Facilitate Evolution and Extensions

Complex concepts typically end up with initial representations that need to be extended over time as the KB is used

“…The first step a cell takes in reading out part of its genetic instructions is to copy the required portion of the nucleotide sequence of DNA – the gene – into a nucleotide sequence of RNA. The process is called transcription because the information, though copied into another chemical form, is still written in essentially the same language – the language of nucleotides. Like DNA, RNA is a linear polymer made of four different types of nucleotides subunits linked together by phosphodiester bonds. It differs from DNA chemically in two respects: (1) the nucleotides in RNA are ribonucleotides – that is, they contain the sugar ribose (hence the name ribonucleic acid) rather than deoxyribose; (2) although, like DNA, RNA contains the bases adenine (A), guanine (G), and cytosine (C), it contains uracil (U) instead of the thymine (T) in DNA. Since U, like T, can base-pair by hydrogen bonding with A, the base-pairing properties described for DNA also apply to RNA…”

-- Essential Cell Biology, Alberts et al. 1992
IKRAFT: Annotating Knowledge Bases with Provenance [Gil & Ratnakar 02]

The first step a cell takes in reading out part of its genetic instructions is to copy the required portion of the nucleotide sequence of DNA - the gene - into a nucleotide sequence of RNA. The process is called transcription because the information, though copied into another chemical form, is still written in essentially the same language - the language of nucleotides. Like DNA, RNA is a linear polymer made of four different types of nucleotides subunits linked together by phosphodiester bonds. It differs from DNA chemically in two respects: (1) the nucleotides in RNA are ribonucleotides - that is, they contain the sugar ribose (hence the name ribonucleic acid) rather than deoxyribose; (2) although, like DNA, RNA contains the bases adenine (A), guanine (G), and cytosine (C), it contains uracil (U) instead of the thymine (T) in DNA. Since U, like T, can base-pair by hydrogen-bonding with A, the base-pairing properties described for DNA also apply to RNA.

Transcription is the process of copying a portion of DNA into RNA

TO BE DEFINED

Events

- copying
- contains

Objects

- Transcription
- process
- portion
- DNA
IKRAFT: Handling Complementary/Contradictory Text Sources [Gil & Ratnakar 02]

Among some 30 biological agents that have been studied as potential weapons, anthrax may be the most likely choice for terrorists because it is easier to acquire than most and is so lethal, killing 30 to 90 percent of all unvaccinated people who are not treated promptly. The anthrax spore is also very durable, able to survive for decades in the soil or other areas protected from direct sunlight. However, anthrax does not spread from one victim to another. That limits its impact to those who inhale the aerosols. Individuals can be protected by a vaccine given in a series of six doses followed by annual boosters, a cumberson regimen. Anthrax can also be treated by antibiotics, including Cipro, penicillin, and doxycycline, if they are administered within days of exposure.

Smallpox is an even bigger worry for some experts because of the global pandemic it could trigger. Unlike anthrax, smallpox is contagious, spreading
Seismic Hazard Analysis: Should Engineer Override Constraint Specified by Model Developer

### Campbell02

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ground-motion-param</td>
<td>3</td>
</tr>
<tr>
<td>type-of-fault</td>
<td>Reverse</td>
</tr>
<tr>
<td>the-site-type</td>
<td>Soft Rock</td>
</tr>
<tr>
<td>magnitude-of-earthquake</td>
<td>4</td>
</tr>
<tr>
<td>distance-to-rupture</td>
<td></td>
</tr>
</tbody>
</table>

#### Range

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 &gt; magnitude-of-earthquake &gt; 4.7</td>
<td>Click for more details</td>
</tr>
<tr>
<td>7.7 &gt; magnitude-of-earthquake &gt; 0</td>
<td></td>
</tr>
<tr>
<td>-1 &lt; ground-motion-param</td>
<td></td>
</tr>
<tr>
<td>10 &gt; ground-motion-param</td>
<td></td>
</tr>
</tbody>
</table>

#### Dependencies

<table>
<thead>
<tr>
<th>Dependency</th>
</tr>
</thead>
</table>

| magnitude-of-earthquake                 |
| ground-motion-param                    |
| ground-motion-param                    |

#### Details

<table>
<thead>
<tr>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click for more details</td>
</tr>
</tbody>
</table>
Examining the Provenance of Axioms in the Knowledge Base
Trellis: Source Description & Qualifications

[Gil and Ratnakar ISWC’02]

Automatically deriving global content trust in sources based on trust ratings from a community of users.

Dublin Core metadata standard to specify attribution

Source Reliability and Credibility Ratings

Ex: NY Times article from REUTERS reporting “At a press conference last Monday, Buckingham Palace was adamant that Prince Larry did not inhale”.

根据来源: Buckingham Palace,考虑到他们想要挽回王室声誉。

根据来源: BBC News,这是完全可靠的。

更多毒品问题
Automatically Deriving Global Trust Ratings [Gil and Ratnakar, ISWC’02]

- Statements attributed to sources can be:
  - Used to reach a conclusion
  - Dismissed in the derivation (tainted)
  - Not used to reach a conclusion

- For each source-statement pair (s,a):
  - C(s,a): credibility rating
  - R(s,a): reliability rating
  - U(s,a): # times used to reach a conclusion
  - T(s,a): # times found tainted
  - N(s,a): # times not used

- Overall rating:
  - \( O(s,a) = k1 \times (C + R) + k2 \times U - k3 \times T - k4 \times N \)
  - \( O(s) = \frac{1}{N} \sum_{1 \text{ to } N} O(s, a) \)
Using Automated Trust Ratings to Help Users Select Sources

### Relevant Sources

<table>
<thead>
<tr>
<th>Statement</th>
<th>Source</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature must be between 90 and 50 °F</td>
<td>Special Operations Site</td>
<td>(8/10)</td>
</tr>
<tr>
<td>Gibraltar temperature 60 °F</td>
<td>NWS Internet Weather Source</td>
<td>(9.3/10)</td>
</tr>
<tr>
<td>Gibraltar avg temperature in March is 60 °F</td>
<td>METOC Rota</td>
<td>(10/10)</td>
</tr>
<tr>
<td>Dublin temperature is 48 °F</td>
<td>NWS Internet Weather Source</td>
<td>(9.3/10)</td>
</tr>
<tr>
<td>Avg Dublin temperature in March is 46 °F</td>
<td>METOC Rota</td>
<td>(10/10)</td>
</tr>
<tr>
<td>Water Temperature is 80 °F</td>
<td>NWS Internet Weather Source</td>
<td>(9.3/10)</td>
</tr>
<tr>
<td>Temperature must be between 90 and 50 °F</td>
<td>Unofficial Special Operations Site</td>
<td>(9.3/10)</td>
</tr>
</tbody>
</table>

### Search Sources by Keyword

<table>
<thead>
<tr>
<th>Statement</th>
<th>Source</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>temperature</td>
<td>NWS Internet Weather Source</td>
<td>(9.2/10)</td>
</tr>
<tr>
<td>METOC Rota</td>
<td>NWS Internet Weather Source</td>
<td>(9.2/10)</td>
</tr>
<tr>
<td>ERDS operations manual</td>
<td>NWS Internet Weather Source</td>
<td>(9.2/10)</td>
</tr>
<tr>
<td>Dennis Reimer Training library</td>
<td>NWS Internet Weather Source</td>
<td>(9.2/10)</td>
</tr>
</tbody>
</table>

### Rating Details for NWS Internet Weather Source: Overall Rating of (9.2/10)

<table>
<thead>
<tr>
<th>Statement</th>
<th>Credibility</th>
<th>Reliability</th>
<th>Used for conclusion</th>
<th>Tainted</th>
<th>Not used</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visibility is around 6 miles</td>
<td>6/6</td>
<td>5/6</td>
<td>1/1</td>
<td>0/1</td>
<td>0/1</td>
<td>9.3/10</td>
</tr>
<tr>
<td>Winds from the east at 10 Knots</td>
<td>6/6</td>
<td>5.5/6</td>
<td>2/2</td>
<td>0/1</td>
<td>0/2</td>
<td>9.6/10</td>
</tr>
<tr>
<td>Gibraltar temperature 60 °F</td>
<td>6/6</td>
<td>5/6</td>
<td>1/1</td>
<td>0/1</td>
<td>0/1</td>
<td>9.3/10</td>
</tr>
<tr>
<td>Wind from SSE at 8 Knots</td>
<td>5/6</td>
<td>5/6</td>
<td>1/1</td>
<td>0/1</td>
<td>0/1</td>
<td>8.6/10</td>
</tr>
</tbody>
</table>
Identified a New Research Area: Content Trust [Gil & Artz JWS’07; Artz & Gil JWS’07]

- Content trust identified as a new research area
- Given trust is voluntary assessment to take a risk based on a prediction of another entity’s behavior, we define:
  - **Content trust**: Voluntary assessment to take a risk based on the information (content) provided by another entity

- Different from existing research on entity trust
  - Entity trust does not examine the content being supplied by the entity
  - Focus on access control and security, reputation, and trust propagation metrics
  - Survey on existing trust research in computer science at [Artz and Gil 06]

- Content trust is related, but different
  - If low trust entities provide same content, that content may be trusted
  - A high trust entity may provide content that contradicts what many low trust entities are providing, so that content may not be trusted

- An important new area of research:
  - What factors that influence content trust can be captured in practice?
  - How can these factors be combined into an overall content trust value?
  - Can users supply valuable information about trust as they analyze sources?
Factors that Affect Content Trust Decisions [Gil & Artz 07]

1. *Topic considered*
2. *Context and criticality of the need for information*
3. *Popularity of the resource*
4. *Recognized authority of associations*
5. *Reputation by direct experience*
6. *Referrals by other users*
7. *Association by other trusted resources (eg citations)*
8. *Expertise of the user*
9. *Perceived bias of source*
10. *Perceived incentive in providing accurate information*
11. *Absence of other alternative resources*
12. *Agreement with other resources*
13. *Provenance and pedigree*
14. *Precise and specific content*
15. *Likelihood of content being correct given what is known*
16. *Time of creation of the content*
17. *Professional appearance*
18. *Likelihood of deceptive behavior*
19. *Recency of factors under consideration*
Dominant Factors of Content Trust [Gil & Artz JWS’07; Gil & Artz WWW’06]

- Study on representative scenarios
  - Users marked sources as: I (Irrelevant), T (Trusted), L (Limited trust), D (Distrusted)

- Results:
  - Results on label distributions:
    - Majority of sources (66%) rated with limited trust, very few with trust (10%) or distrust (14%)
  - Results on associations:
    - At least one association was considered per source
    - Unable to determine majority of associations (66%), esp authority
      - But there are Web authority algorithms
    - In very few cases associations ratings were opposite
      - Negative in authority and positive in provenance
  - Results on deception:
    - Able to determine biases in majority of cases (72%)
      - Compared to 33% rate for associations
    - Bias is highly correlated with trust and distrust decisions
    - Bias alone not a good a predictor of trust and distrust decisions
    - Association bias is largely indicative of bias

- Conclusions:
  - Most salient factors to judge content trust: associations and bias
  - Bias factors combined into single bias value
  - Bias is mostly associations bias
  - Associations are much harder to determine than bias

- Associations
  - Authority
  - Provenance
  - Related Resources

- Bias
  - Bias
  - Incentive
  - Deception
Simulating Collection of User Trust Feedback

[Gil and Artz 06; Gil and Artz 07]

- Trust estimates are better as user feedback increases in:
  - Trust and Distrust scenario
  - Distrust Only scenario
  - Sparse Trust and Distrust scenario.

- Continuous user feedback is better than binary user feedback.
Knowledge Capture in the New Millennium

Semantic Web

- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases
- Many reasoners
- Varying quality and coverage

Research Topics

- Harnessing volunteer contributions
- Natural tutorial instruction
- Meta-level reasoning
- OWL and semantic web languages
- Semantic web reasoners
- Provenance and trust

Semantic Web

- Many, many, MANY users
- No time for training
- Many distributed subsystems
- Many knowledge bases
- Many reasoners
- Varying quality and coverage
More Information at www.isi.edu/ikcap

The "Interactive Knowledge Capture" Research Group at ISI

List of Projects

- Tailor (Helping users to examine and modify procedural knowledge using short sentences)
- Wings (Workflow Instance Generation and Selection)
- Pegasus (Planning and Execution for Grids)
- Cognitive Grids (Intelligent Middleware for Distributed Problem Solving)
- Learner (Acquiring world knowledge from volunteer contributors)
- Windward (Automatic Generation of Workflows for Large-Scale Distributed Data Analysis)
- Beamer (Assisting Users with To-Do Lists)
- TELMe (Learning Procedures from Scaffolded Tutorial Instruction)
- MathTrust (Mathematical Analysis of Trust and Deception in Information Sources)
- MetaLearning (Meta-reasoning to Coordinate Multiple Learning Methods)
- PrivatEe (Protecting Data through a Privacy-Aware Workflow System)

Other Projects

- CAT (An Intelligent Assistant for Workflow Composition)
- SCETIT (Information Infrastructure for Earthquake Research)
- Stick (Skills for Learning to Interactively Capturing Knowledge)
- Echo (Memory-based meta-Level reasoning tool for Interactive Knowledge Capture)
- Trellis (Web-based tool enabling argumentation which leverages information from multiple sources)
- VerbOcean (Extracting fine-grained semantic relations between verbs)
- Expect (A reflective architecture for knowledge acquisition)
- KANAL (Knowledge Analysis on Process Models)
- Constable (Helping users to examine and modify constraints)
- Docker (Helping people connect software)
- Phosphorus (A Knowledge and Experience-Based Agent Capabilities Matcher)
- Electric Elves