Introduction to (KnEng) Knowledge-Engineering

via

Examples of (engineering) knowledge-based applications

by

Michael Bobak

My background

MS ComputationalScience/AI UIUC
17years of AI in .edu/.gov/.com

Agenda

2 case studies showing Kn-Engineering/Mgt
Giving an intro to tech/concepts:
Matching via Cases/Rules
[CBR/Rule-Based]

2 case-studies

Problem: helping CallServiceRep(CSR)s

1)phone: too much to learn, high turn-over

2)mail: large repetitive backlog

Approach to Kn-Eng/Mgt

Software Eng, focuses on software ecosystem
I focus on the Knowledge-ecosystem

Kn-Acquisition: starting with subj-matter-experts
Build description beyond DB

Solutions

Phone: match caller record against case-library of prototypical problems with associated question-tree for that problem-case, then use record to prune-Qs/simplify the trees, and present a ranked set of cases all before person gets on the line; then choose&follow proper Q-tree

Email: match email text against against a set of (bag of word)cases that represent a hierarchy of needs and products,
&regexp of name/phone/etc.
Rules match on combination of (specific to general) concepts, to provide most likely request concepts; map: need+object → action
Rules map to a response composition: "thank you mr/ms... for your product-info/defect/.. request on ..., I hope this helps.."

Phone case/Qtree prune-rule

```
(defcase
   missingPayment
(daysSincePayment
   ?days&>40)
(resolveWith
   mkPaymentQtree))
```

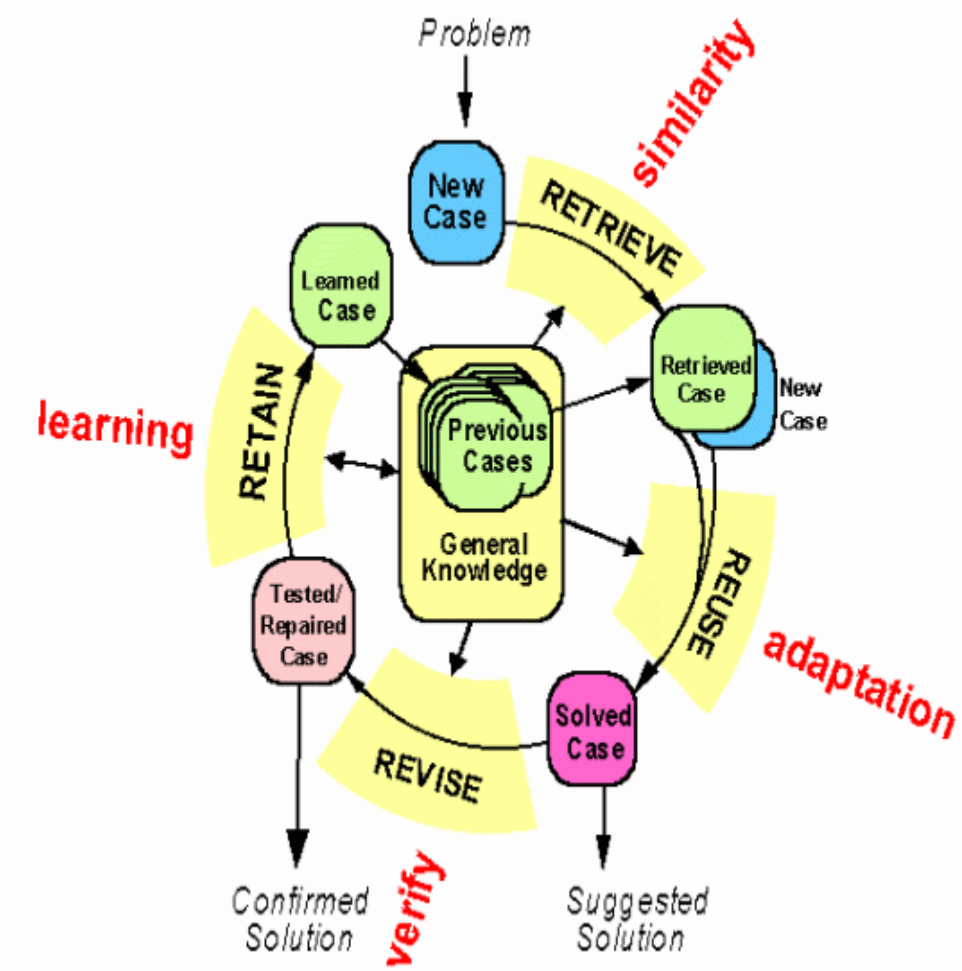
```
(defrule pruneLast3Q
 (loanRec
  (resolveWith
      mkPaymentQtree)
 (lastPayDates
      (nil ok ok)))
      =>
      (removeQ last3Q
          mkPaymentQtree))
```

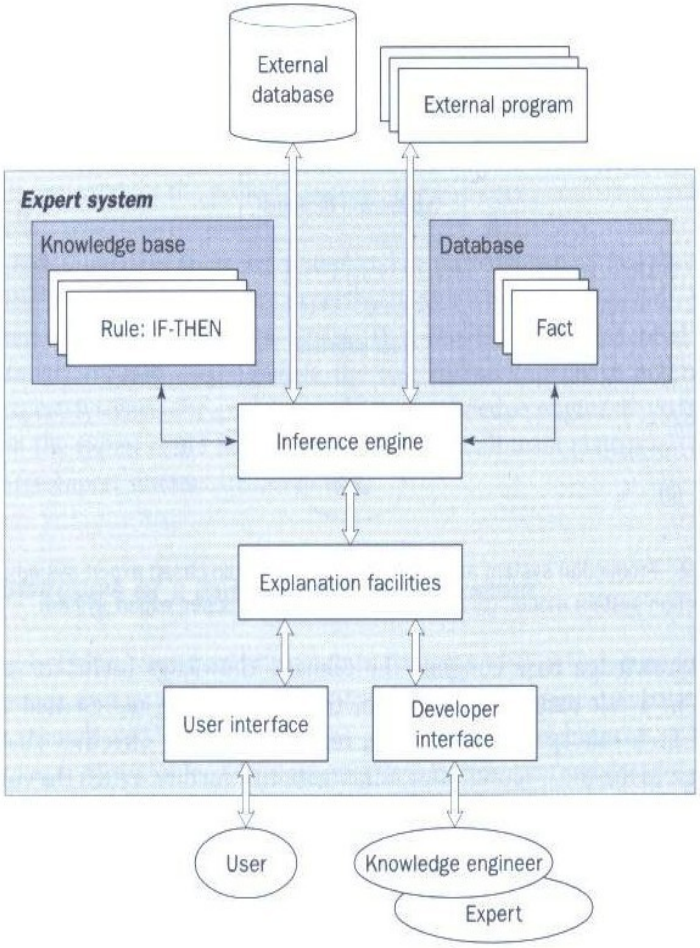
Email case/hierarchy &rules

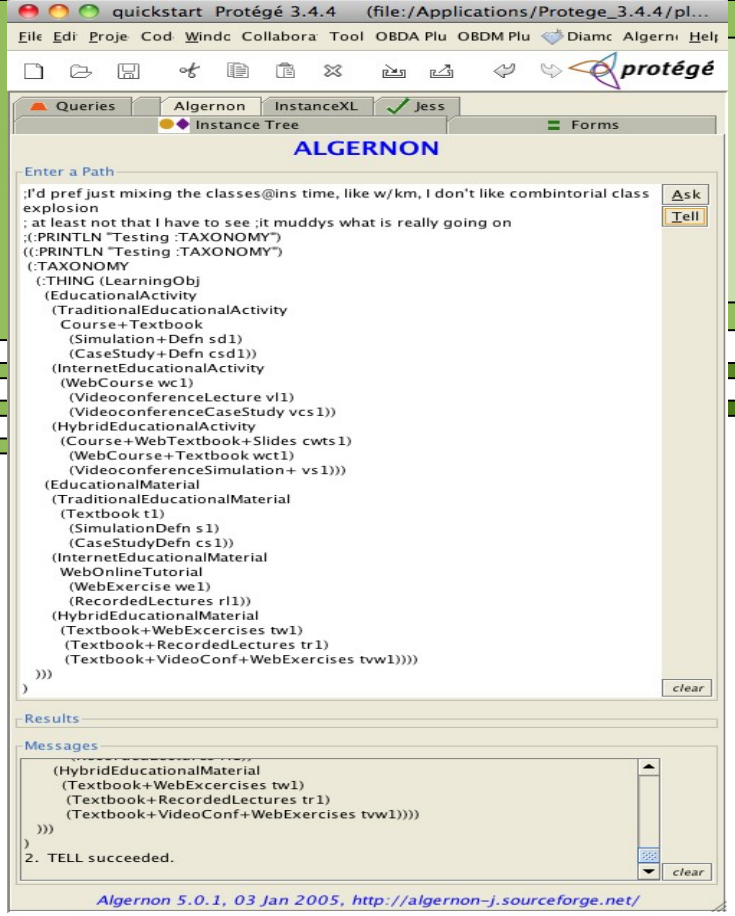
```
(:Taxonomy (:Thing (:cases
 (Query
   (product-info
      your-product that-you-carry)
   (defect-query
      problem-with broken))
 (product-lines
   ((home-needs
        home-product)
      (shampoo
          best-suds))))))
```

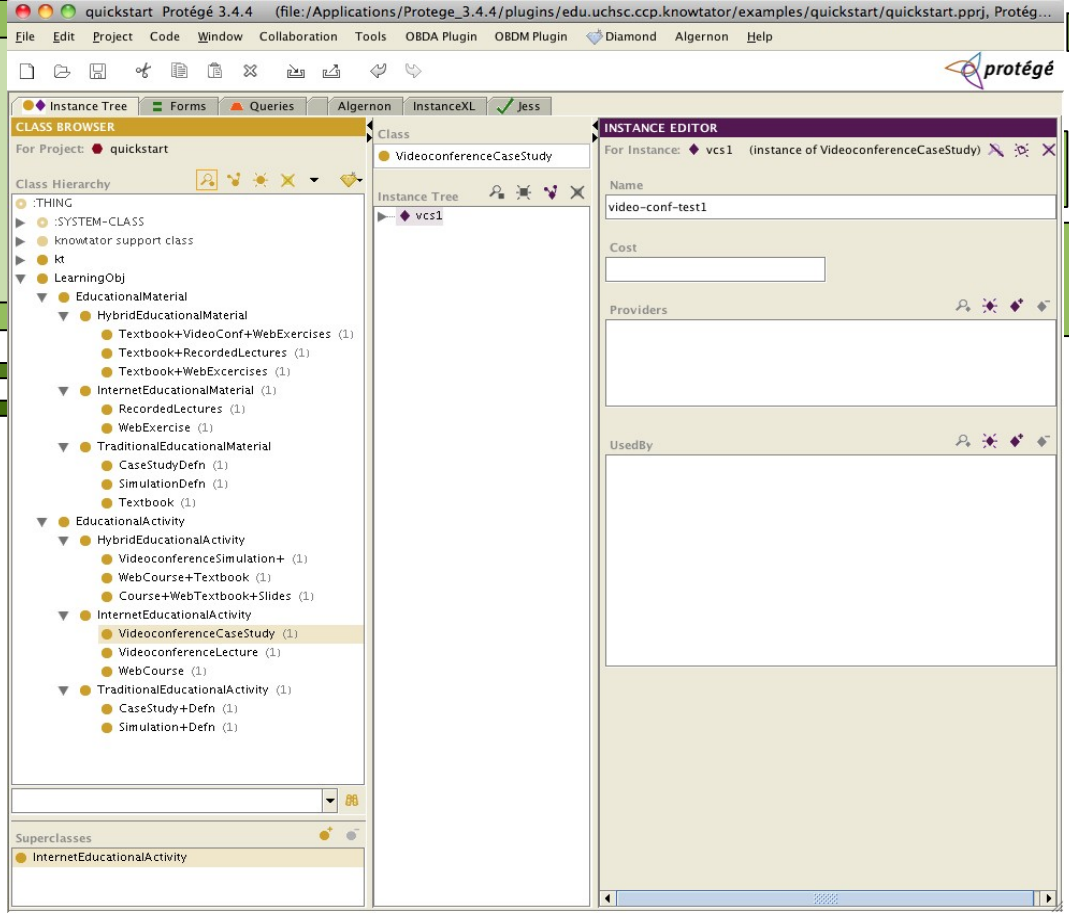
```
(defrule  map2response
 (Query ?qry)
 (product-lines ?prod)
 (customerName ?name)
=>
 (send2 ?name
   (prodInfo ?prod ?qry)))
```

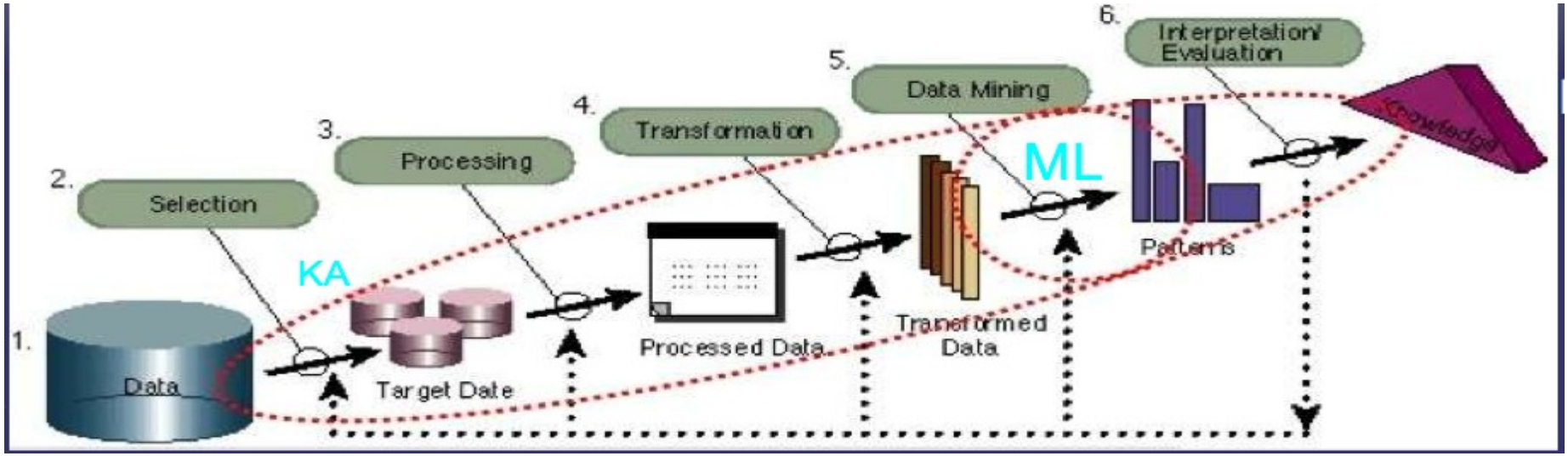
Case/Rule Mgt






quickstart Protégé 3.4.4 (file:/Applications/Protege_3.4.4/pl...
File Edit Project Code Window Collaboration Tool OBDA Plug OBDM Plug Diamond Algernon Help
protégé
Queries
Algernon
InstanceXL
Jess
Instance Tree
Forms
ALGERNON
Enter a Path
;I'd pref just mixing the classes@ins time, like w/km, I don't like combintorial class explosion
; at least not that I have to see ;it muddys what is really going on
;(:PRINTLN "Testing :TAXONOMY")
((:PRINTLN "Testing :TAXONOMY")
(:TAXONOMY
(:THING (LearningObj
(EducationalActivity
(TraditionalEducationalActivity
Course+Textbook
(Simulation+Defn sd1)
(CaseStudy+Defn csd1))
(InternetEducationalActivity
(WebCourse wc1)
(VideoconferenceLecture vl1)
(VideoconferenceCaseStudy vcs1))
(HybridEducationalActivity
(Course+WebTextbook+Slides cwts1)
(WebCourse+Textbook wct1)
(VideoconferenceSimulation+ vs1)))
(EducationalMaterial
(TraditionalEducationalMaterial
(Textbook t1)
(SimulationDefn s1)
(CaseStudyDefn cs1))
(InternetEducationalMaterial
WebOnlineTutorial
(WebExercise we1)
(RecordedLectures rl1))
(HybridEducationalMaterial
(Textbook+WebExcercises tw1)
(Textbook+RecordedLectures tr1)
(Textbook+VideoConf+WebExercises tvw1))))
)))
)
Ask
Tell
clear
Results
Messages
(HybridEducationalMaterial
(Textbook+WebExcercises tw1)
(Textbook+RecordedLectures tr1)
(Textbook+VideoConf+WebExercises tvw1))))
)))
)
2. TELL succeeded.
clear
Algernon 5.0.1, 03 Jan 2005, http://algernon-j.sourceforge.net/


quickstart Protégé 3.4.4 (file:/Applications/Protege_3.4.4/plugins/edu.uchsc.ccp.knowtator/examples/quickstart/quickstart.pprj, Protég...
File Edit Project Code Window Collaboration Tools OBDA Plugin OBDM Plugin Diamond Algernon Help
protégé
Instance Tree
Forms
Queries
Algernon
InstanceXL
Jess
CLASS BROWSER
For Project: quickstart
Class Hierarchy
:THING
:SYSTEM-CLASS
knowtator support class
kt
LearningObj
EducationalMaterial
HybridEducationalMaterial
Textbook+VideoConf+WebExercises (1)
Textbook+RecordedLectures (1)
Textbook+WebExcercises (1)
InternetEducationalMaterial (1)
RecordedLectures (1)
WebExercise (1)
TraditionalEducationalMaterial
CaseStudyDefn (1)
SimulationDefn (1)
Textbook (1)
EducationalActivity
HybridEducationalActivity
VideoconferenceSimulation+ (1)
WebCourse+Textbook (1)
Course+WebTextbook+Slides (1)
InternetEducationalActivity
VideoconferenceCaseStudy (1)
VideoconferenceLecture (1)
WebCourse (1)
TraditionalEducationalActivity (1)
CaseStudy+Defn (1)
Simulation+Defn (1)
Superclasses
InternetEducationalActivity
Class
VideoconferenceCaseStudy
Instance Tree
vcs1
INSTANCE EDITOR
For Instance: vcs1 (instance of VideoconferenceCaseStudy)
Name
video-conf-test1
Cost
Providers
UsedBy


1.
Data
2.
Selection
KA
Target Date
3.
Processing
Processed Data
4.
Transformation
Transformed Data
5.
Data Mining
ML
Patterns
6.
Interpretation/ Evaluation
Knowledge

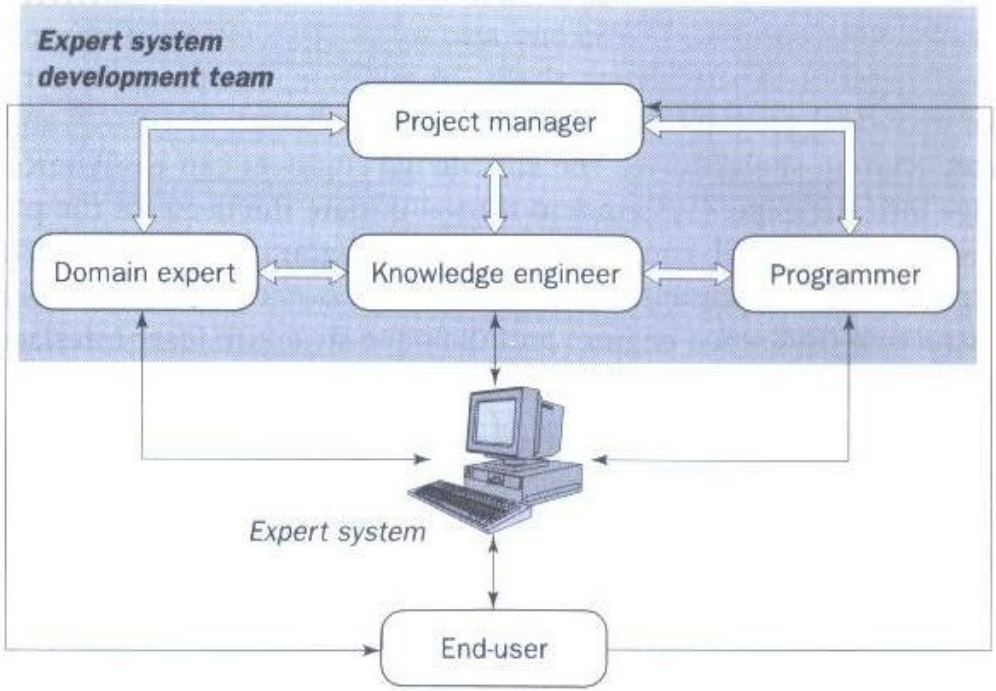
teamwork

Strong motivation for

Having time with
SubjMatterExperts

&having IT help provide
all the needed data
→

the best projects



rollout/measured-return(ROI)

Phone: 6months
Decrease csr learning time & turnover
Increased capacity&profit.

Email: <1year
Increased capacity, as CSR only answered when phones were slow; so huge backlog.
Handled the bulk of (boring repeat) Qs

Other examples/ Questions?

Conceptual search (medical) from free-text

Mortgage sales workflow rules from DB; Fully CBR CSR Q&A

Setup experiment control/log/learning sys

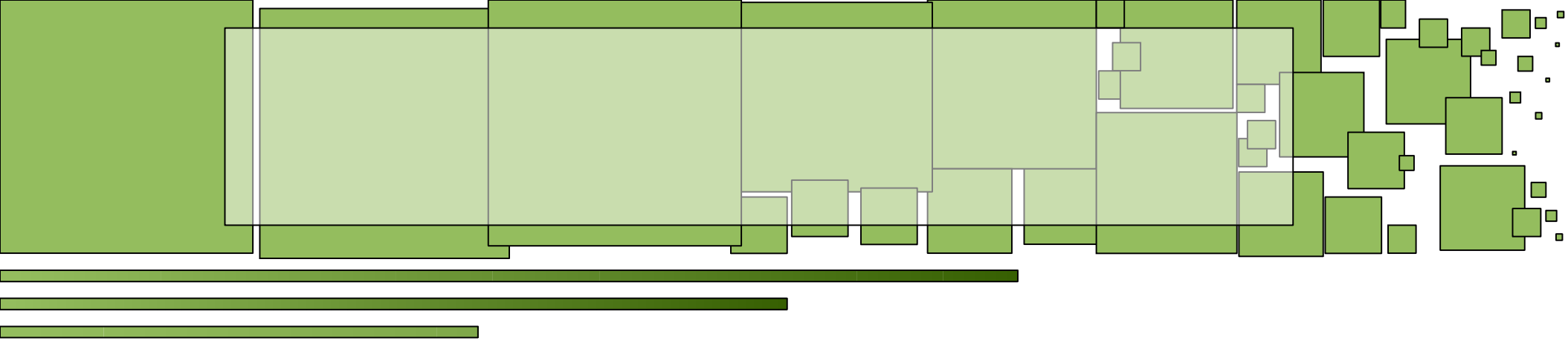
Maintain large telco diagnosis system

Two (specific) Intelligent-Tutoring-Systems (ITS):

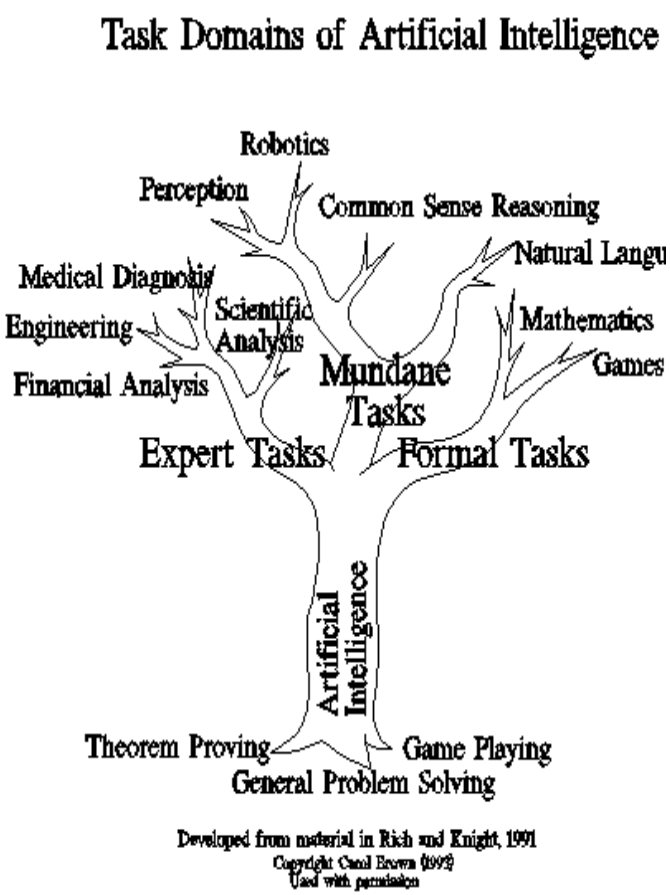
- Thermodynamics tutor: MBR, rules w/TMS (explanations)
- Sim-Ship in distress: sim/viz Prolog, fwd-Rules, BN

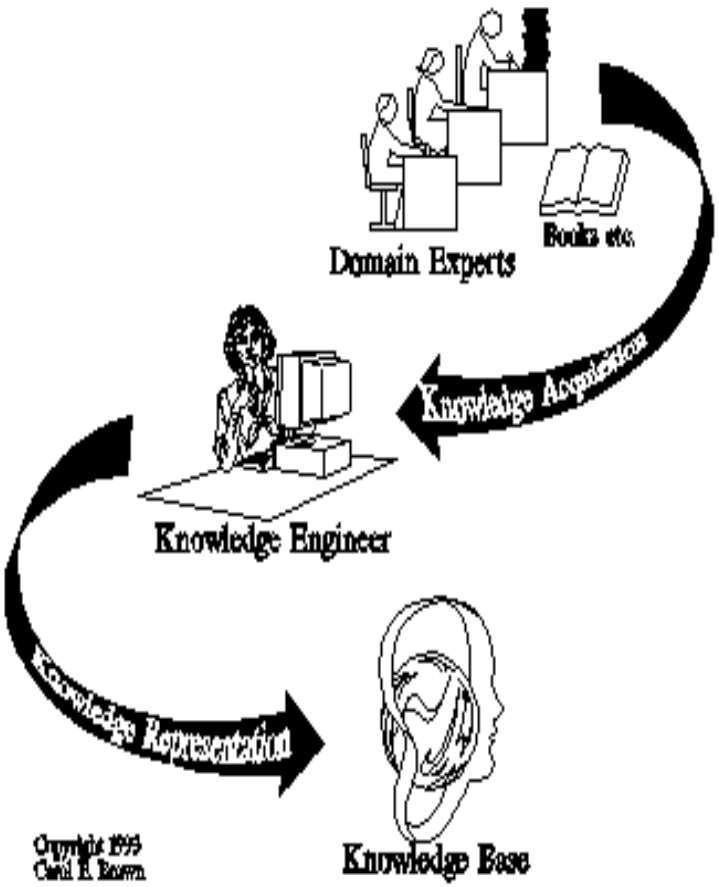
Multi-Env-Sim coordination: sim/CLIPS/PVM (distributed obj/services)

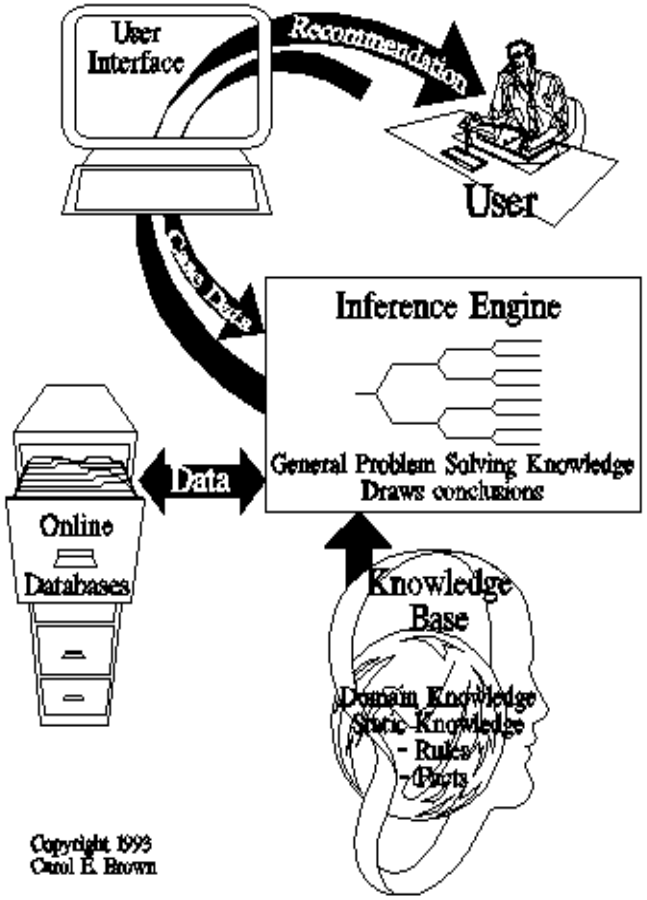
Anything else from my CV, that is of interest &/or General Questions



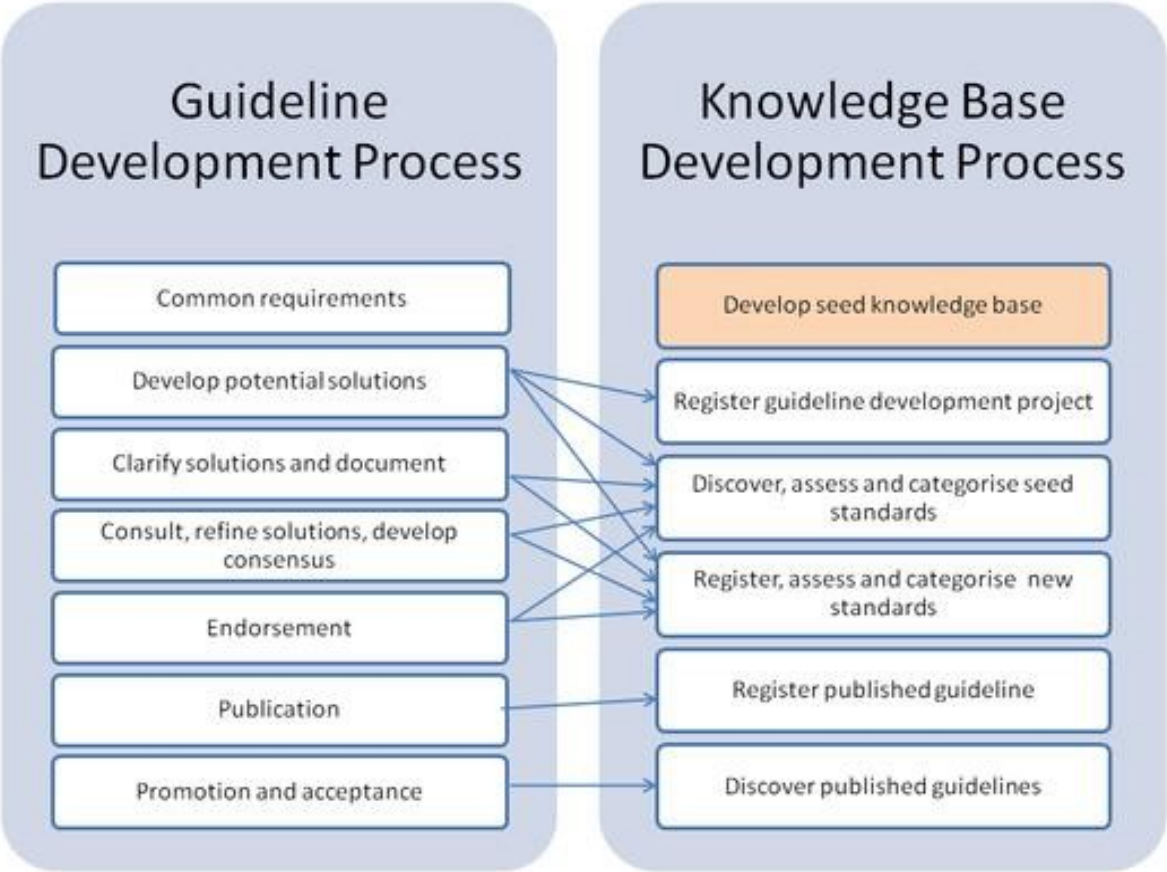
domain-KnAq->KB-sys

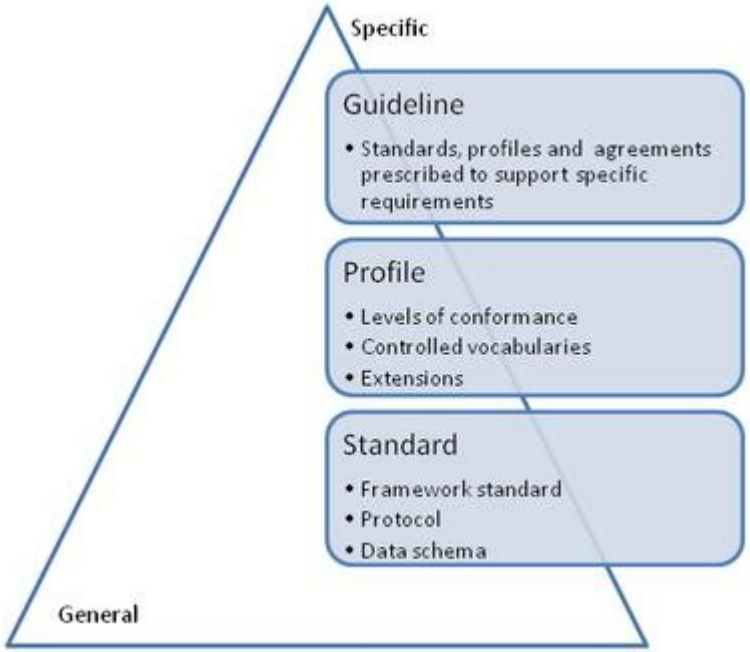






Edu KB-dev example





General path to getting personalized problem solutions

Need to represent your world&break down prob:

- Find problem state description/representations
- Find (sub)solution-*cases* &/or *rules*-of-thumb which can match on instances of (sub)problem
- Have system move from base-data to higher-level descriptions that are more easily mapped to final solutions for the case at hand; just as an expert in the field would do

Organizing Knowledge

- Create concept hierarchies so you can describe what you want to match at the proper level once
- '*Match* → *action*'s evolves the system state
- Match knowledge is very modular (if you)
- Value explicit over implied Kn in procedures
- This makes Kn Acq/Mgt much faster/easier

Human experts	Expert systems	Conventional programs
Use knowledge in the form of rules of thumb or heuristics to solve problems in a narrow domain.	Process knowledge expressed in the form of rules and use symbolic reasoning to solve problems in a **narrow domain**.	Process data and use algorithms, a series of well-defined operations, to solve general numerical problems.
In a human brain, knowledge exists in a compiled form.	Provide a **clear separation of knowledge from its processing**.	Do not separate knowledge from the control structure to process this knowledge.
Capable of explaining a line of reasoning and providing the details.	**Trace the rules fired** during a problem-solving session and **explain how** a particular conclusion was reached and **why** specific data was needed.	Do not explain how a particular result was obtained and why input data was needed.
Use inexact reasoning and can deal with incomplete, uncertain and fuzzy information.	Permit **inexact reasoning** and can deal with incomplete, uncertain and fuzzy data.	Work only on problems where data is complete and exact.
Can make mistakes when information is incomplete or fuzzy.	**Can make mistakes** when data is incomplete or fuzzy.	Provide no solution at all, or a wrong one, when data is incomplete or fuzzy.
Enhance the quality of problem solving via years of learning and practical training. This process is slow, inefficient and expensive.	Enhance the quality of problem solving by adding new rules or adjusting old ones in the knowledge base. When new knowledge is acquired, **changes are easy** to accomplish.	Enhance the quality of problem solving by changing the program code, which affects both the knowledge and its processing, making changes difficult.

Other examples/ Questions?

Conceptual search (medical) from free-text

Mortgage sales workflow rules from DB; Fully CBR CSR Q&A

Setup experiment control/log/learning sys

Maintain large telco diagnosis system

Two (specific) Intelligent-Tutoring-Systems (ITS):

- Thermodynamics tutor: MBR, rules w/TMS (explanations)
- Sim-Ship in distress: sim/viz Prolog, fwd-Rules, BN

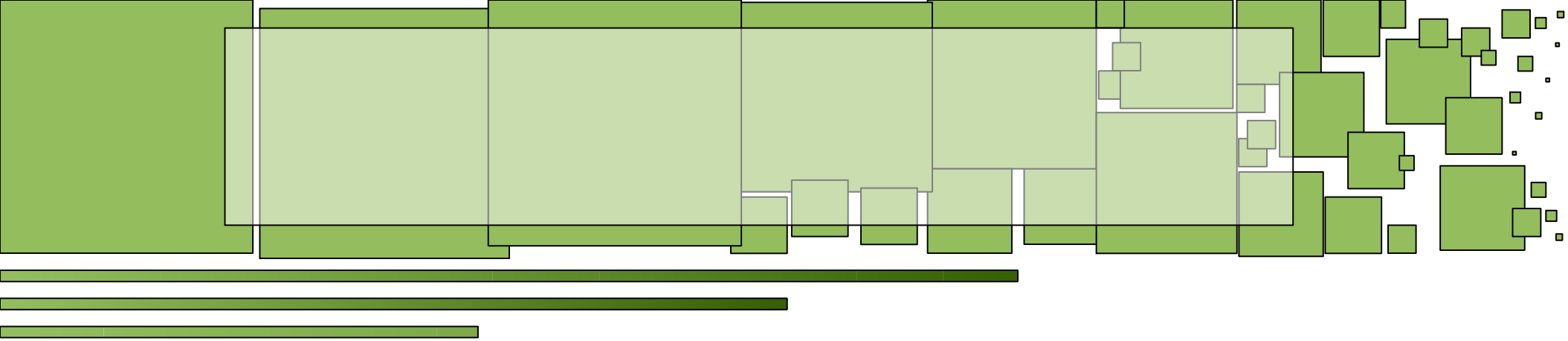
Multi-Env-Sim coordination: sim/CLIPS/PVM (distributed obj/services)

Anything else from my CV, that is of interest &/or General Questions

Challenges to building a KB sys

Getting time with motivated sets of Subject-Matter-Experts, and having as few distractions in tasks that do not pertain to finding, capturing/routing the data/knowledge, to create the core KB/reasoning.

The best projects are when the current team: end-users, mgt, & IT are all on board&ready to help



Long-term goal

- An easy to build&maintain system w/high ROI
- Take the largest chunk of the problem that can be solved with the least amount of effort
- Find/organize/extend Kn-Representation just enough so can represent matching for state transitions towards finding the best final goal

The Present Situation

- Any aid centered around what is presently captured in DB; usually just display/manipulation
- Rest done with text based manuals
- Usually little culling of useless options and no direction toward most likely next steps
- Have to look up in manuals, vs having the more tedious rules applied automatically

Development up to present

- Closed/inflexible *implied* use of knowledge
- Data might be viewed&used but not Kn
- This comes from old problem outlooks
- This turns out not to fit well w/new Kn economy
- Still works on old(er) problems though

Potential Alternatives

- Can keep all useful-kn/description closed
- Or open it up in an *explicit*/modular/usable way
- Adding stove-piped procedures is familiar
- Explicit KnMgt&use/reasoning reusably scales
- Old way slows dev; change raise costs quickly
- New way: better return, without cost going up

Recommendation

- Start incrementally augmenting old systems to be able to solve new problems
- Know that you have captured only little bit of your Kn; much of what you have is outside a DB&even w/in it relationships not made explicit in any machine actionable way
- It is time to get as much of what you already have(in DBs)&can easily move into computable forms, over&start exercising it
- And prep for further efficient capture/extraction/learning
- Having knowledge in computable forms can allow for a new generation of problem solving & speed ups, that scale well
- Start picking the low hanging fruit; get people to task/focused
- SubjectMatterExperts, KnEng/Coordinators, &technologists

Real World Examples

- Usually add as little representation as possible and do no learning; but still get a big 1st jump
- If your problem is a little more open ended &/or you want to automate more for an end user without having much expert aid; then you should plan to organize/capture/use a bit more
- Many examples, but will use 2 that show some of the types of matching that is often useful
- I have many others that can help explain other facets