11th Protégé Conference

2009

Amsterdam
Netherlands

A great year for Protégé

- 11th great Protégé Conference
- 21st anniversary of PROTÉGÉ I
- 123,612 Protégé registrations
- Major development activities shifting from Protégé 3 to Protégé 4

Lots of new stuff happening to Protégé

- Even more performance enhancements
- New features that facilitate collaboration
- New Web-based version for Protege
- Amazing new plug-ins for
  - Rules
  - Spreadsheets
  - Cognitive support
- More intergration with technology from the National Center for Biomedical Ontology
- All the work that we will hear about for the first time at this conference!

Protégé at 21

Protégé no longer gets carded

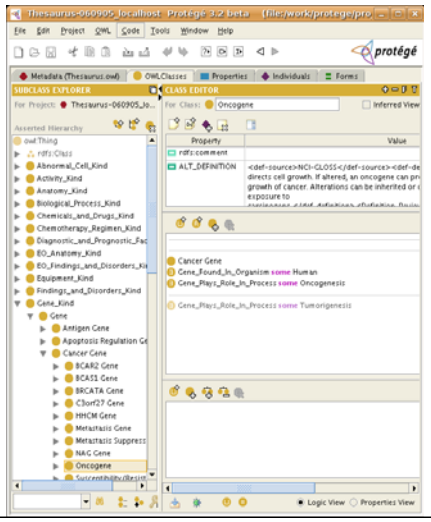
Mark A. Musen

Stanford Center for Biomedical Informatics Research

The Protégé ontology editor

- Free, open source ontology editor and knowledge-base framework
- Support for different:
  - ontology languages (OWL, RDF(S), Frames)
  - backends: Database, XML, CLIPS, etc.
- Strong user community: more than 123K downloads
- Used widely in academic, government, and industry

http://protege.stanford.edu



PROTÉGÉ-I was build for a different world

- No Web
- No “agents”
- No notion of ontologies as engineering artifacts
- No standard languages for knowledge representation
- No significant interest in description logic
- Just tons of people trying to build rule-based expert systems—that were failing

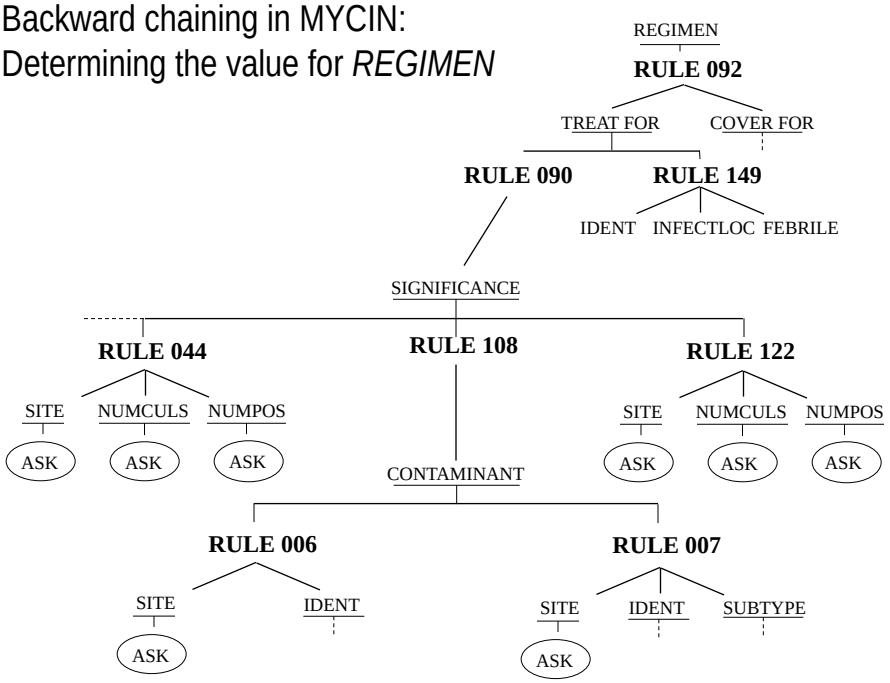
Sample MYCIN Rule

```
PREMISE:  ($AND
  (SAME CNTXT GRAM GRAMPOS)
  (SAME CNTXT MORPH COCCUS)
  (SAME CNTXT CONFORM CLUMPS))
ACTION: (CONCLUDE CNTXT IDENT STAPHYLOCOCCUS
       TALLY 700)
```

IF: 1) The gram stain of the organism is grampos
2) The morphology of the organism is coccus
3) The conformation of the organism is clumps

THEN:
There is suggestive evidence (.7) that
the identity of the organism is staphylococcus

Backward chaining in MYCIN:
Determining the value for *REGIMEN*



Consider this rule …

IF: (1) A "Complete Blood Count" test is available
(2) The White Blood Cell Count is
less than 2500

THEN:

The following bacteria might be causing infection:
E. coli,
Pseudomonas aerugenosa
Klebsiella-pneumonia

What is implicit in this rule?

- "White Blood Cell count less than 2500" *is-a-subclass-of* "immunosuppressed patient," which *is-a-subclass-of* "compromised host"
- E. coli, Pseudomonas, and Klebsiella are *instances* of "gram negative rod," which *is-a subclass-of* "bacterium normally found in the gut"
- Unless a Complete Blood Count test has been ordered, it's pointless to ask the value of the White Blood Cell Count (White Blood Count *is-a-part-of* a Complete Blood Count)

Some other screening clauses in MYCIN

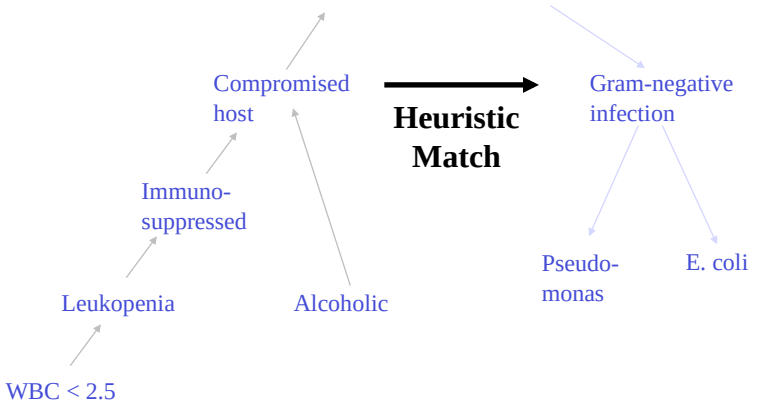
- If the patient has undergone surgery and the patient has undergone neurosurgery, then …
- If the patient is older than 17 and the patient is an alcoholic, then …

Screening clauses coerce the system to ask questions in a certain way, while obscuring the knowledge that caused the clauses to be created in the first place.

Why rule-based systems failed

- A few hundred rules were barely manageable; a few thousand rules were impossible to keep straight.
- Developers "programmed" the systems in nonobvous ways, by tinkering with the order of rules and of clauses
- Developers could rarely tell by inspection how any element of the system contributed to problem solving

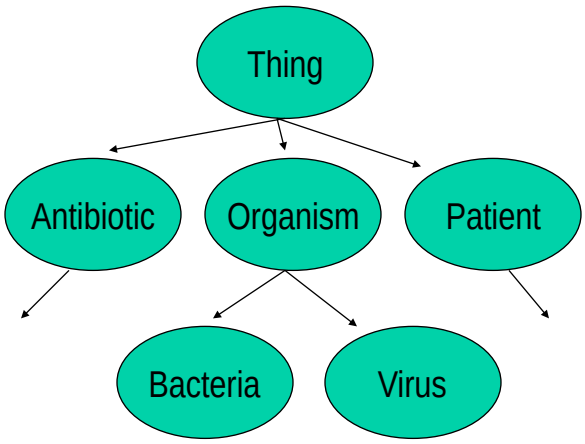
Heuristic classification in MYCIN (after Clancey)



Conceptual building blocks for designing intelligent systems

- Domain ontologies
  - Characterization of concepts and relationships in an application area, providing a domain of discourse
- Problem-solving methods (PSMs)
  - Abstract algorithms for achieving solutions to stereotypical tasks (e.g., constraint satisfaction, classification, planning, Bayesian inference)

For MYCIN, those building blocks would be …



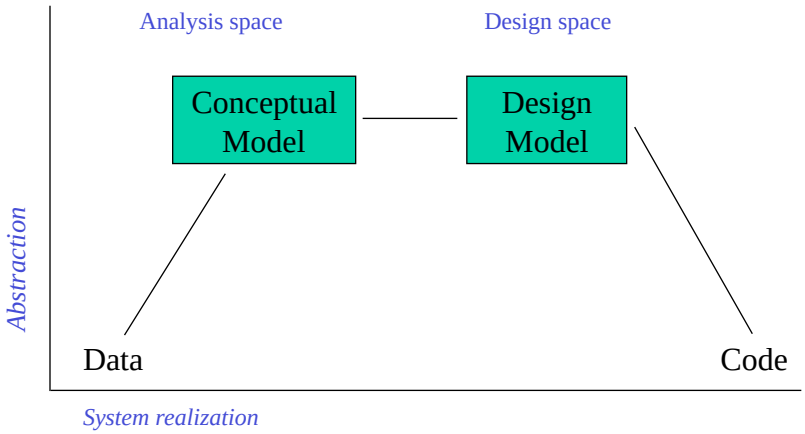
1. An ontology of infectious diseases

2. A problem-solving method that can use the ontology to identify likely pathogens and to recommend appropriate treatment

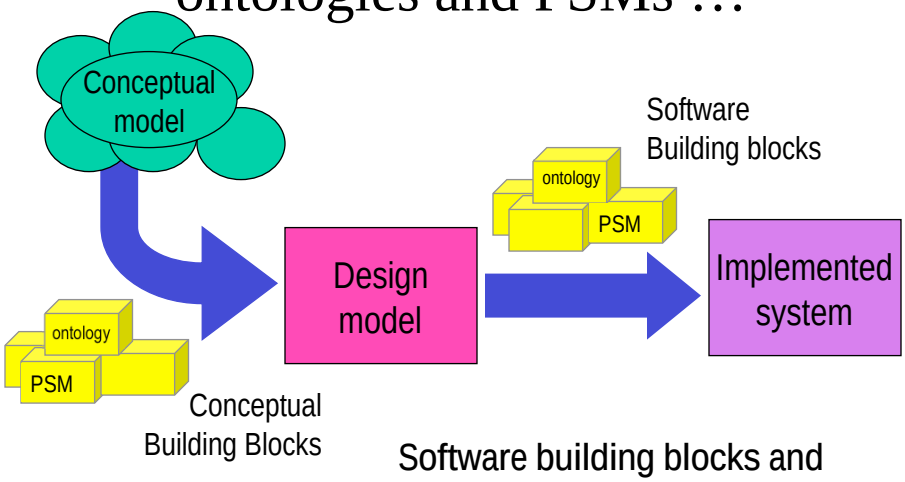
Common KADS

- Result of nearly 20 years of collaborative research in the European Union
- Centered at University of Amsterdam, with dozens of other partners
- Applies principled, software-engineering approach to development of intelligent systems
- De facto software-engineering standard for building intelligent systems

Conceptual models and design models in CommonKADS

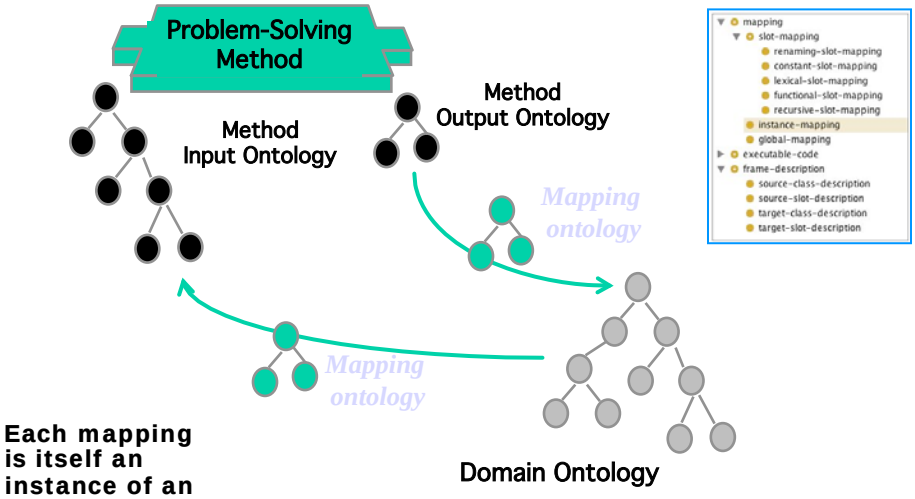


When building systems from ontologies and PSMs …



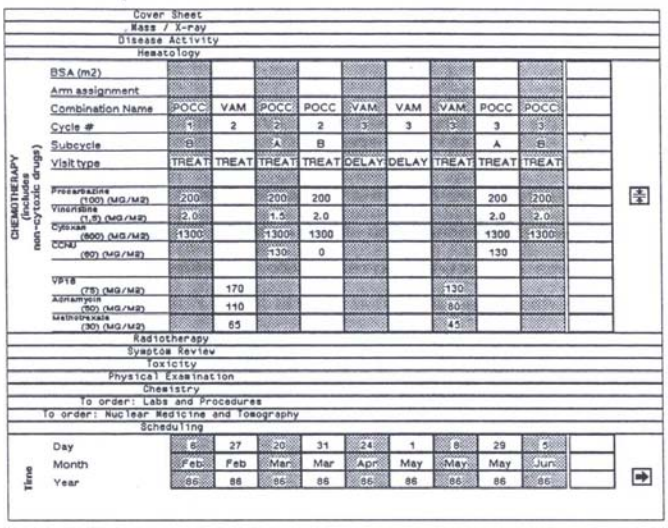
Software building blocks and conceptual building blocks can be identical!

Mapping domain to PSM explicitly



Each mapping is itself an instance of an ontology of possible mapping types

User interface from the workstation version of ONCOCIN (ca. 1986)



A rule from an early version of ONCOCIN (ca. 1980)

RULE075

To determine the attenuated dose for drugs in MOPP chemotherapy or for all drugs in PAVe chemotherapy

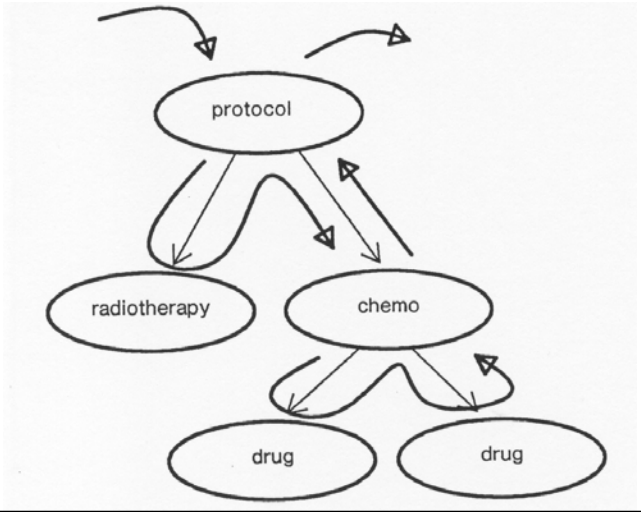
IF:

1) This is the start of the first cycle after a cycle as aborted, and
2) The blood counts do not warrant dose attenuation

THEN:

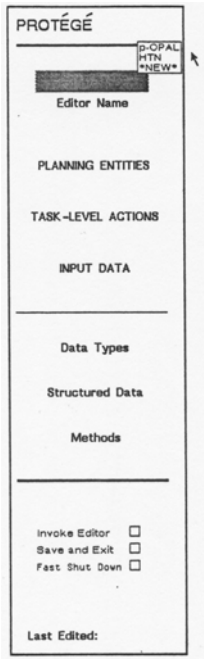
Conclude that the current attenuated dose is 75% of the previous dose

Episodic Skeletal Plan Refinement was the Problem Solver used with PROTÉGÉ I



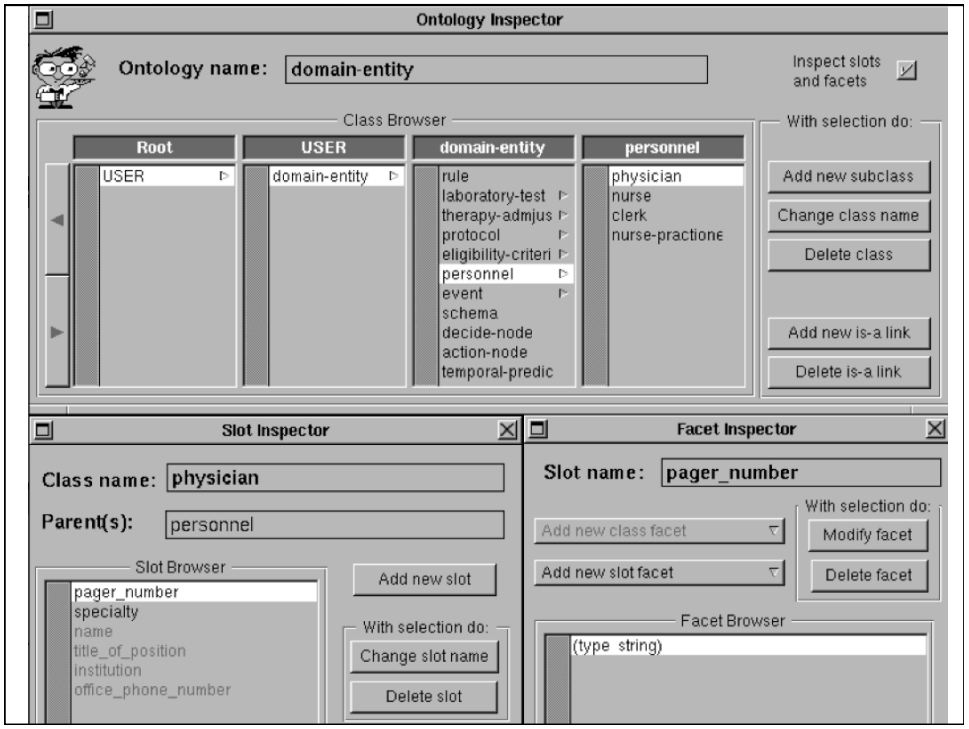
PROTÉGÉ I construed problem solving as the interplay of

- A hierarchy of **plans** that might be invoked
- **Actions** that could affect the way in which the planning would take place
- **Data** input from the enviroment that might directly or indirectly predicate the plans to be involved or the actions to take

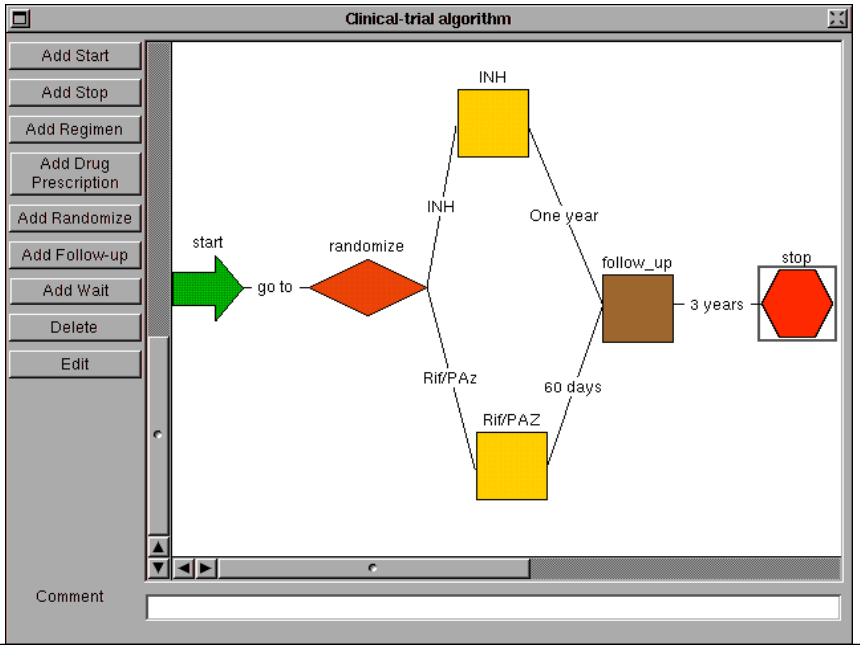


The Next Step: PROTÉGÉ-II

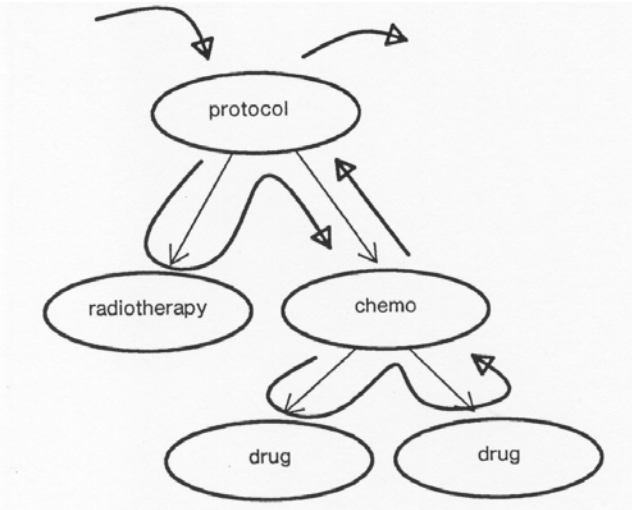
- Made ontologies explicit with a separate ontology editor
- Supported arbitrary problem-solving methods—dropped the dependence on ESPR
- Allowed sophisticated facilities for generating knowledge-acquisition interfaces based on the domain ontology
- Took advantage of sophisticated NeXTSTEP object-oriented UI system
- First tool to use the Protégé nerd icon!



A clinical algorithm in PROTÉGÉ-II



Episodic Skeletal Plan Refinement was the Problem Solver used with PROTÉGÉ I

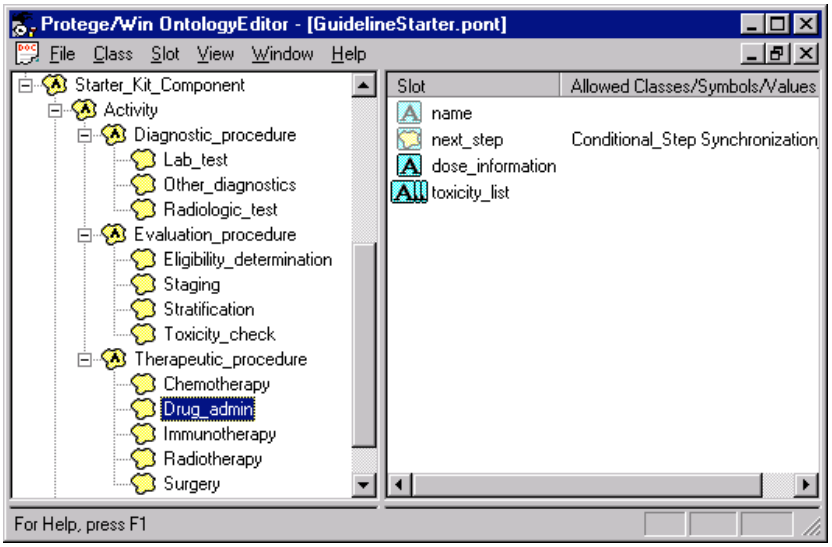


Protégé/Win

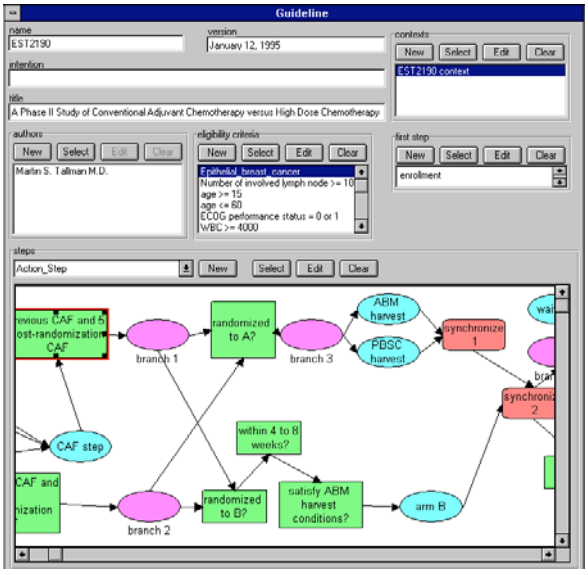
Built for the Masses!

- Moved Protégé to a widely available platform—just in time!
- Enabled integrated ontology editing and forms layout —eliminating the need for batch forms generation
- Marked the start of a growing Protégé user community

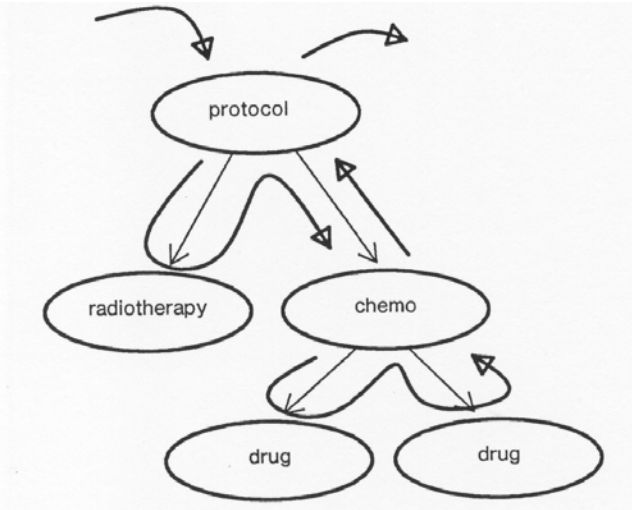
A Protocol Ontology in Protégé/ Win



Protégé/Win KA tool

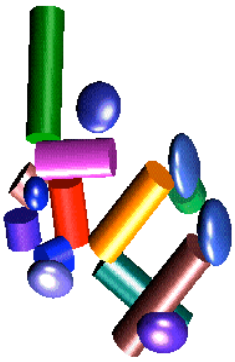


Episodic Skeletal Plan Refinement was the Problem Solver used with PROTÉGÉ I

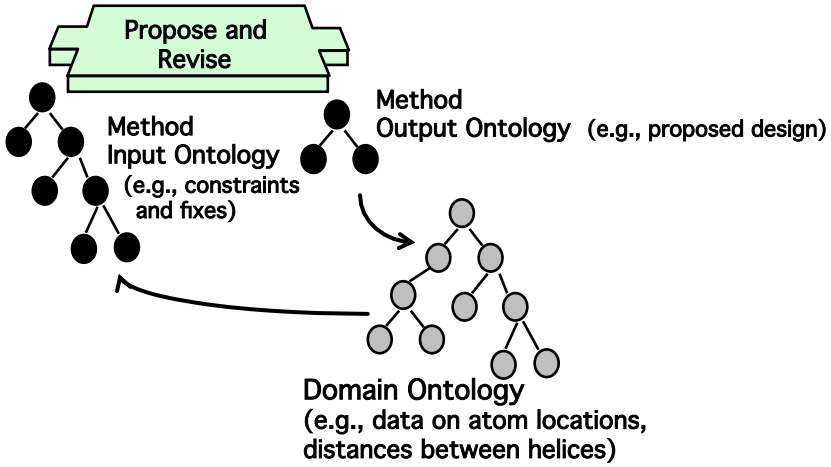


Reuse of the *propose-and-revise* method

- Determination of ribosome structure from NMR data can be construed as constraint satisfaction
- Mapping propose-and-revise to a new domain ontology automates the structure-determination task

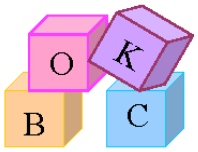


Use of *propose-and-revise* to solve the ribosome problem

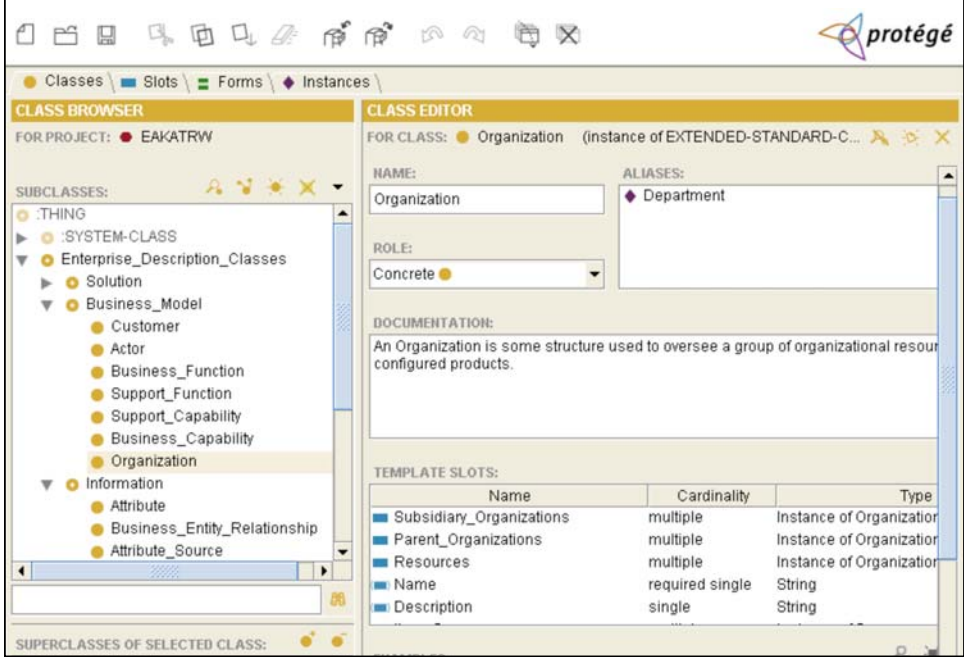


The Next Step: Protégé-2000

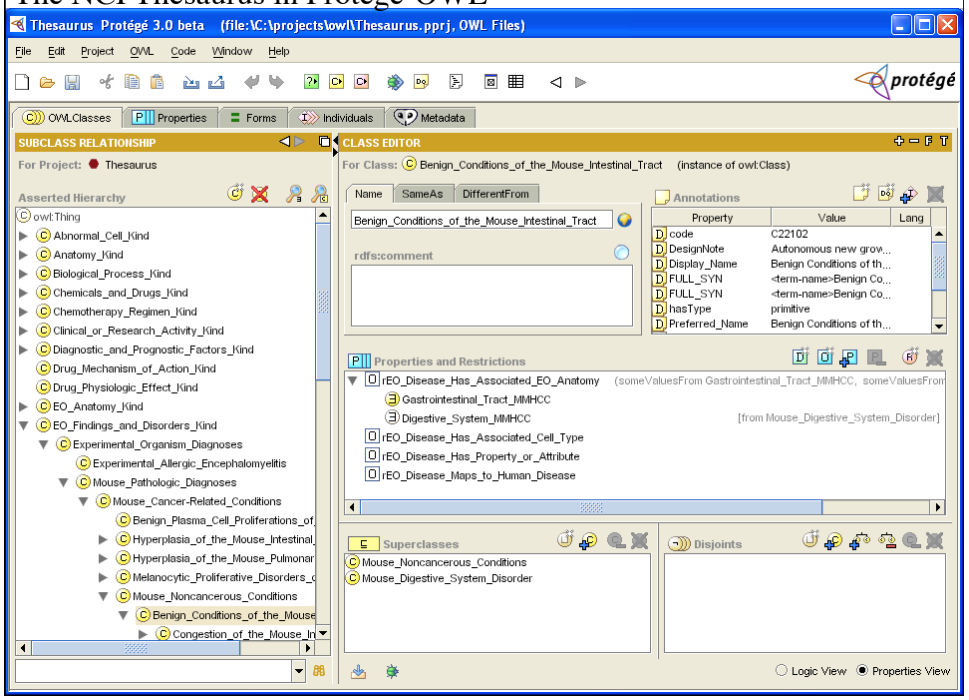
- Ray Fergerson rewrote the whole thing in **Java**
- We provided support for the (then) OKBC frame standard
  - Metaclasses
  - Slots as first-class entities
  - Axioms
- We created an open, **plug-in** architecture



Perot Systems Organizational Model in Protégé-Frames



The NCI Thesaurus in Protégé-OWL



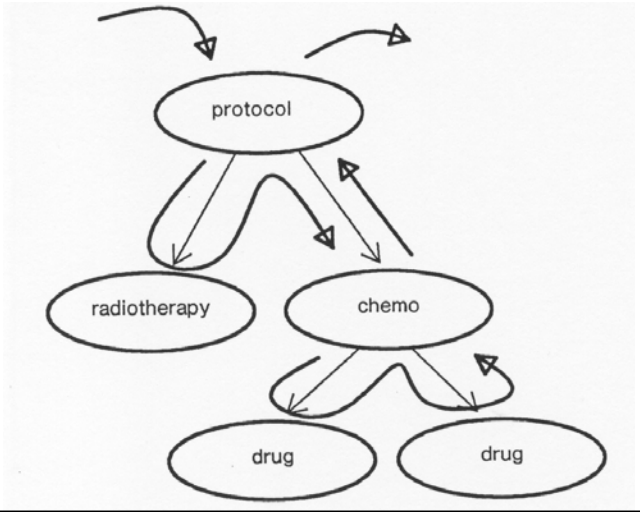
By now, everyone was concentrating on ontologies

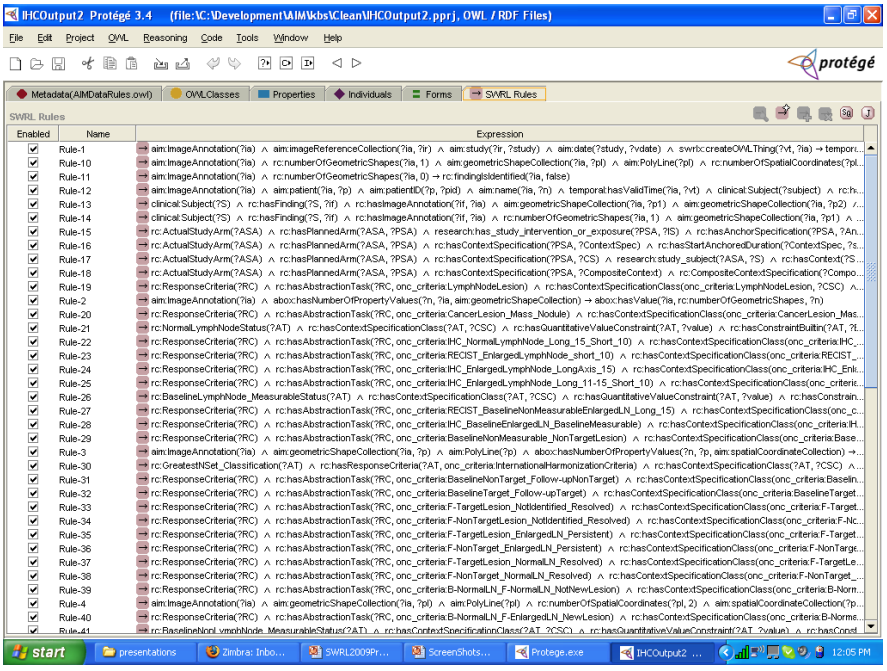
- The world rediscovered description logic
- The emphasis became building better and better knowledge representations
- Ontologies alone were great for question-anwering tasks
- Tools for building ontologies (including Protégé) flourished
- And people became less focused on problem solving

The Era of Big Ontologies was Upon Us

- Foundational Model of Anatomy
- NCI Thesaurus
- Gene Ontology
- Word Net
- SNOMED-CT
- OBI

Episodic Skeletal Plan Refinement was the Problem Solver used with PROTÉGÉ I





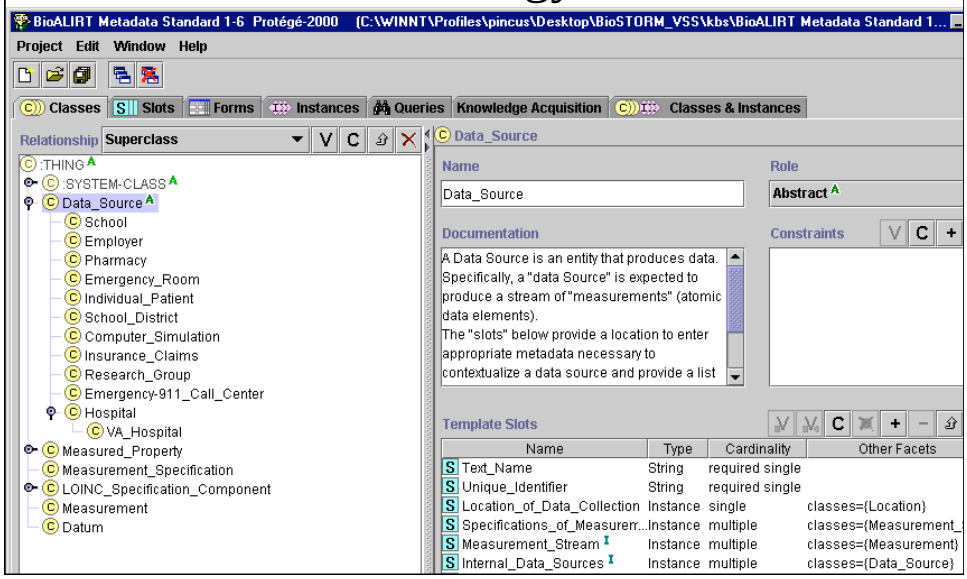
How can we evaluate ontologies independent of problem solvers?

- How do we know whether they make the "right" distinctions?
- How do we know where the gaps are?
- How do we find inconsistent granularity?
- How do we know what our ontologies are actually competent at describing?

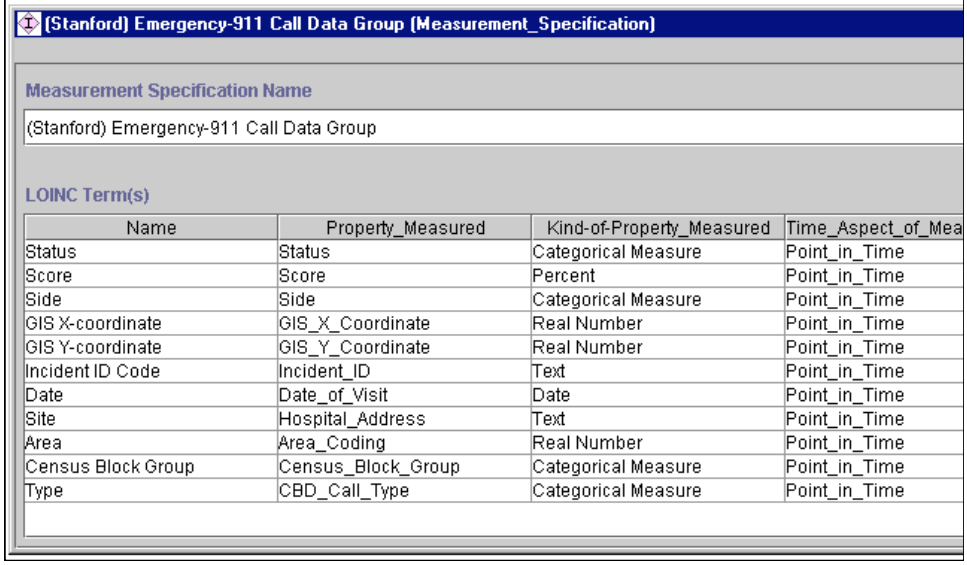
BioSTORM: A Prototype Next-Generation Surveillance Sytem

- Developed at Stanford, initially with funding from DARPA, now from CDC
- Provides a test bed for evaluating alternative data sources and alternative problem solvers
- Demonstrates
  - Use of ontologies for data acquisition and data integration
  - Use of a high-performance computing system for scalable data analysis

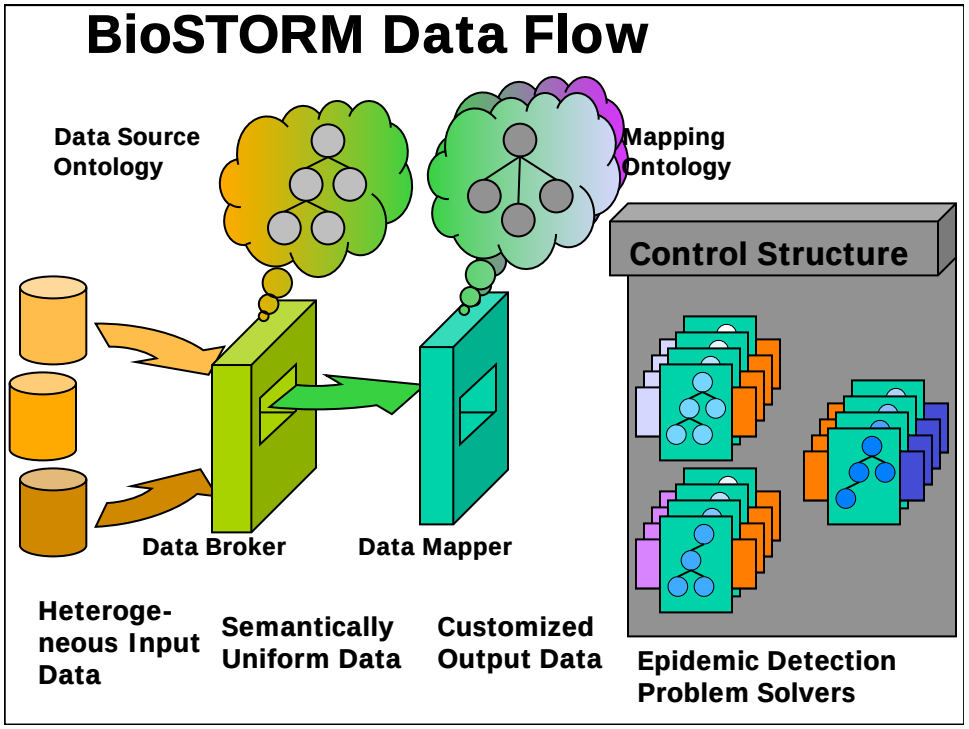
Biosurveillance Data Sources Ontology

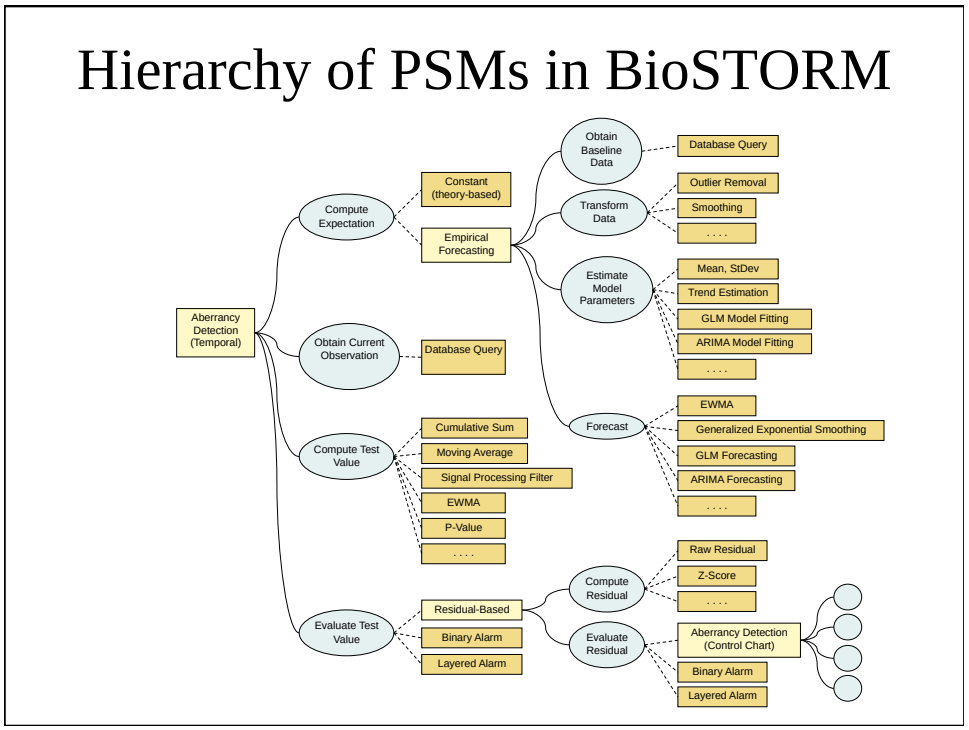


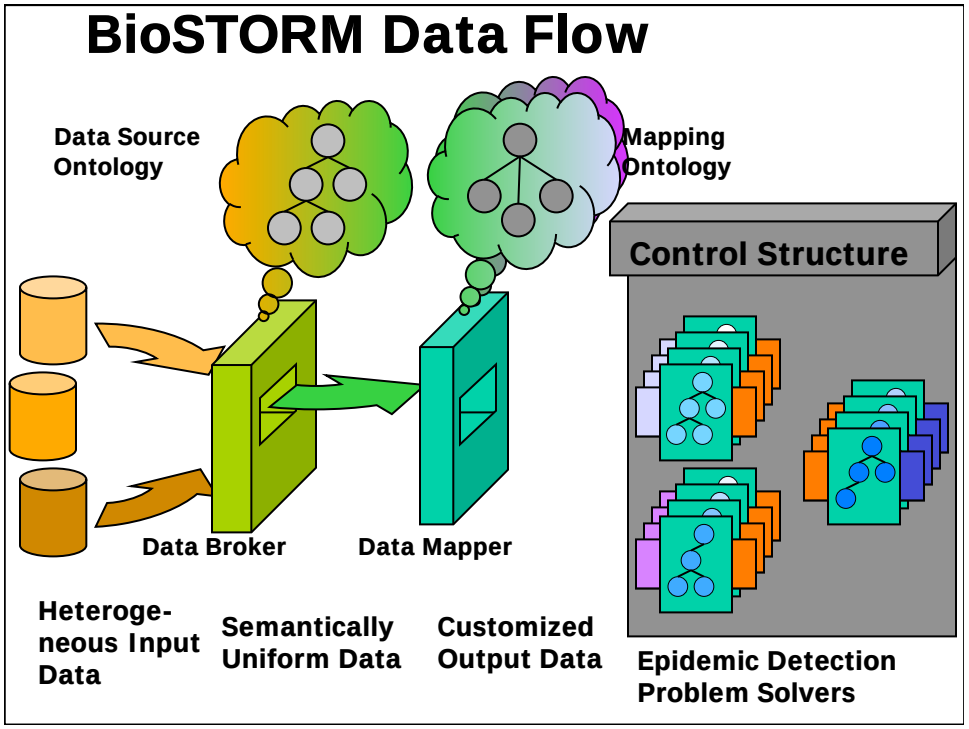
Ontology defines how data should be accessed from the database



Name	Property_Measured	Kind-of-Property_Measured	Time_Aspect_of_Mea
Status	Status	Categorical Measure	Point_in_Time
Score	Score	Percent	Point_in_Time
Side	Side	Categorical Measure	Point_in_Time
GIS X-coordinate	GIS_X_Coordinate	Real Number	Point_in_Time
GIS Y-coordinate	GIS_Y_Coordinate	Real Number	Point_in_Time
Incident ID Code	Incident_ID	Text	Point_in_Time
Date	Date_of_Visit	Date	Point_in_Time
Site	Hospital_Address	Text	Point_in_Time
Area	Area_Coding	Real Number	Point_in_Time
Census Block Group	Census_Block_Group	Categorical Measure	Point_in_Time
Type	CBD_Call_Type	Categorical Measure	Point_in_Time


BioSTORM Data Flow
Data Source Ontology
Mapping Ontology
Control Structure
Data Broker
Data Mapper
Heteroge-neous Input Data
Semantically Uniform Data
Customized Output Data
Epidemic Detection Problem Solvers

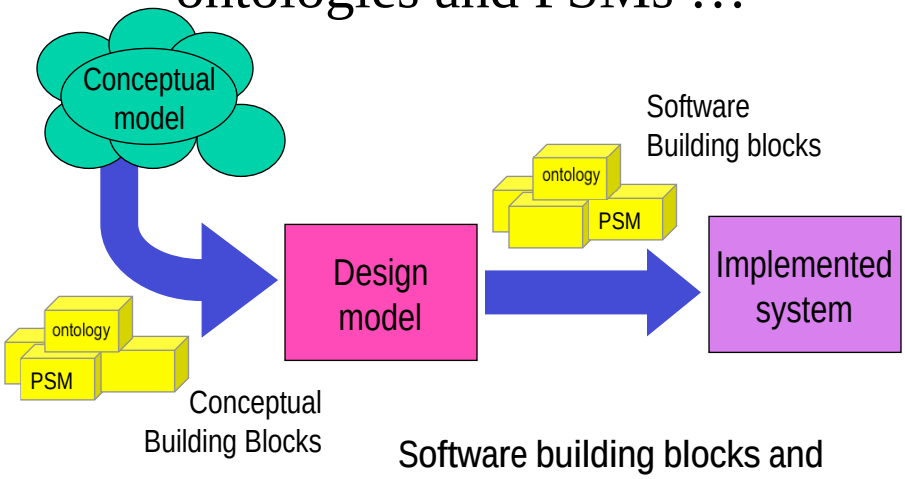

Hierarchy of PSMs in BioSTORM
Aberrancy Detection (Temporal)
Compute Expectation
Constant (theory-based)
Empirical Forecasting
Obtain Baseline Data
Database Query
Transform Data
Outlier Removal
Smoothing
. . . .
Estimate Model Parameters
Mean, StDev
Trend Estimation
GLM Model Fitting
ARIMA Model Fitting
. . . .
Forecast
EWMA
Generalized Exponential Smoothing
GLM Forecasting
ARIMA Forecasting
. . . .
Obtain Current Observation
Database Query
Compute Test Value
Cumulative Sum
Moving Average
Signal Processing Filter
EWMA
P-Value
. . . .
Evaluate Test Value
Residual-Based
Binary Alarm
Layered Alarm
Compute Residual
Raw Residual
Z-Score
. . . .
Evaluate Residual
Aberrancy Detection (Control Chart)
Binary Alarm
Layered Alarm



There is a need for balance

- Better languages and tools for building domain ontologies
- Better languages and for designing and implementing problem-solving methods
- Better methods and tools for bringing these components together
- Building systems with *use cases*—not ontologies—as the driving component

When building systems from ontologies and PSMs …



Software building blocks and conceptual building blocks can be identical!