THE ARCHITECTURE OF WATSON HOW DOES IT WORK?

Grady Booch
IBM Fellow

(with deep appreciation to the entire Watson team for access to their inner sanctum)

What Is Watson Not?

Watson is not

the beginning of Skynet

nor our new computer overlord

nor an advanced search engine

nor a fancy database retrieval system.

What Is Watson?

Watson is a reasoning system

with a question and answer front end

that processes natural language

across structured and unstructured data

using deep analytic algorithms

that the system learns to combine

in optimal ways.

How Does Watson Work?

Watson operates by

analyzing a question

generating hypotheses (forward chaining)

collecting evidence (backward chaining)

then presenting its results with scored

levels of confidence.

What Is Unique About Watson?

Watson is unique in that it

attends to heterogeneous sources

postulates multiple possible answers

considers evidence across multiple dimensions and learns.

Who Is Watson?

Watson does not

understand

nor does it think.

What's So Hard About This Problem?

Real language is filled with nuance, slang, and metaphor*.

Reasoning about open-ended problems requires inferring

context

and meaning

and relevance

from evidence that is often incomplete,

ambiguous, and sometimes contradictory.

* David Ferrucci, Watson Principle Investigator

Opening the Curtain on Watson

Watson was

built by humans

and it’s still just ones and zeros

down at the bottom.

What Is Architecture?

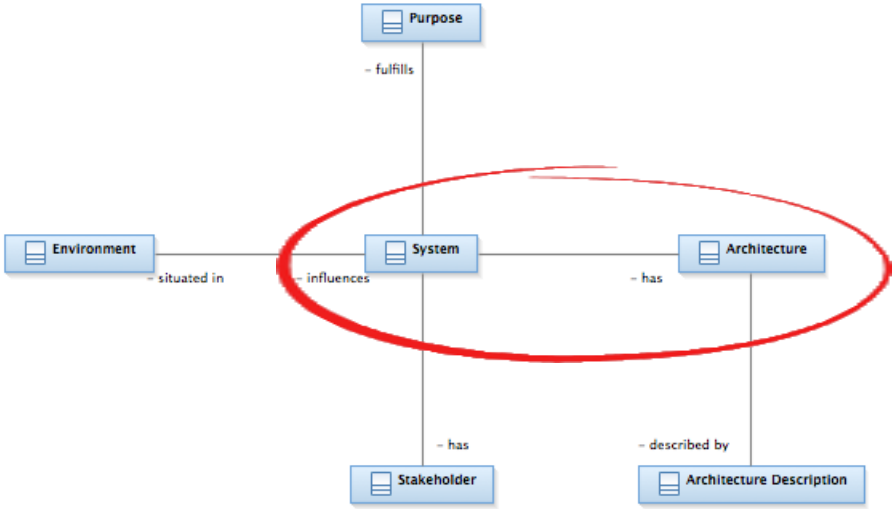
- Architecture as **essence**.

  Architecture is the fundamental conception of a system in its environment embodied in elements, their relationships to each other and to the environment, and principles guiding system design and evolution.

- Architecture as **blueprint**.
- Architecture as **literature**.
- Architecture as **language**.
- Architecture as **decision**.

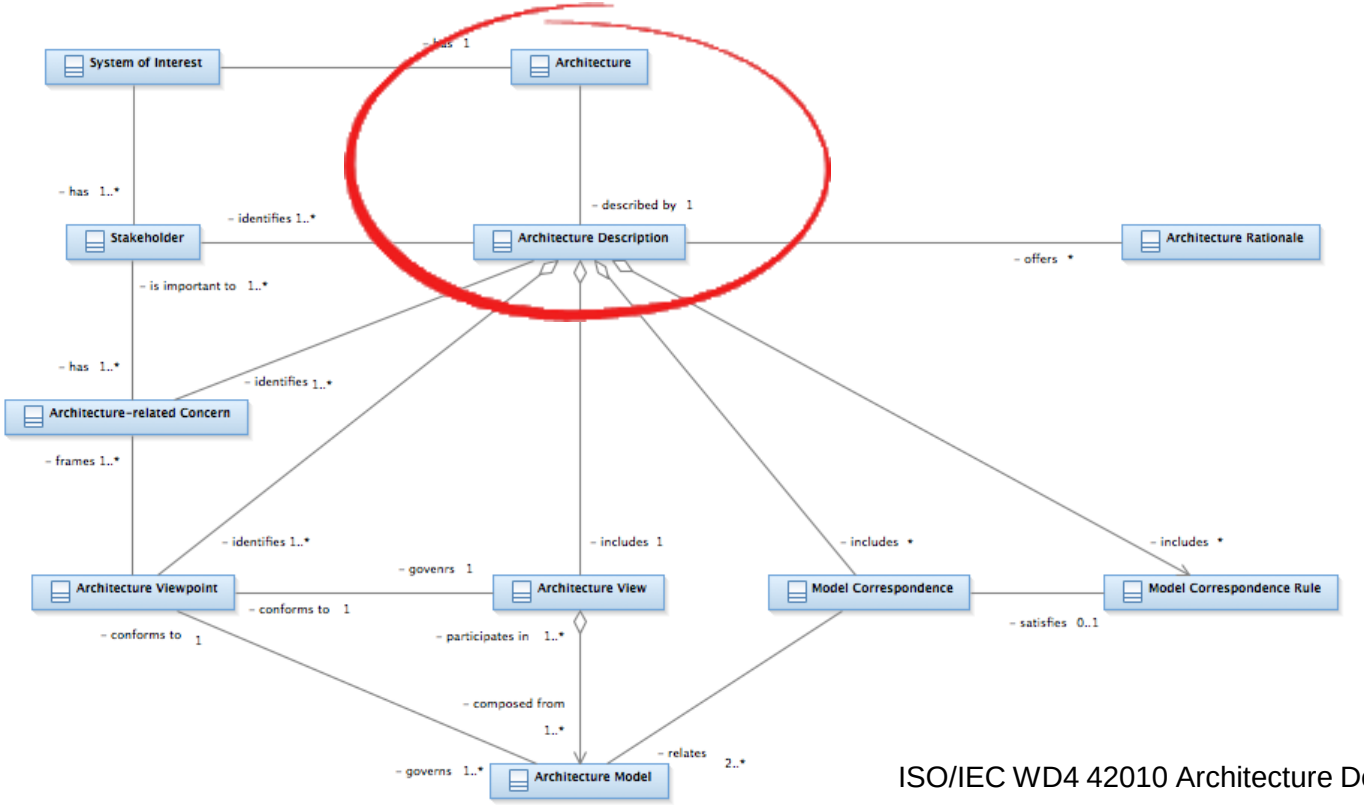
ISO/IEC WD4 42010 Architecture Description

Describing A System's Architecture



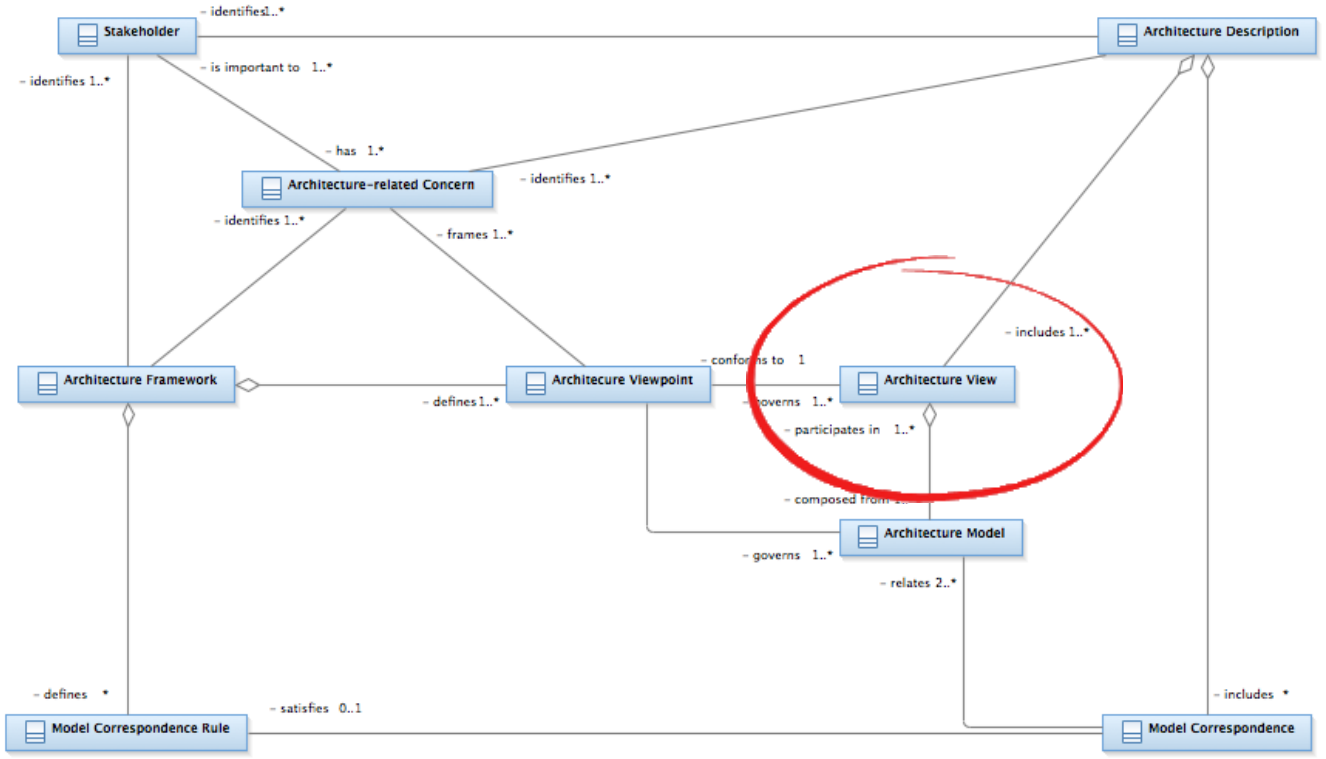
ISO/IEC WD4 42010 Architecture Description

Describing A System's Architecture



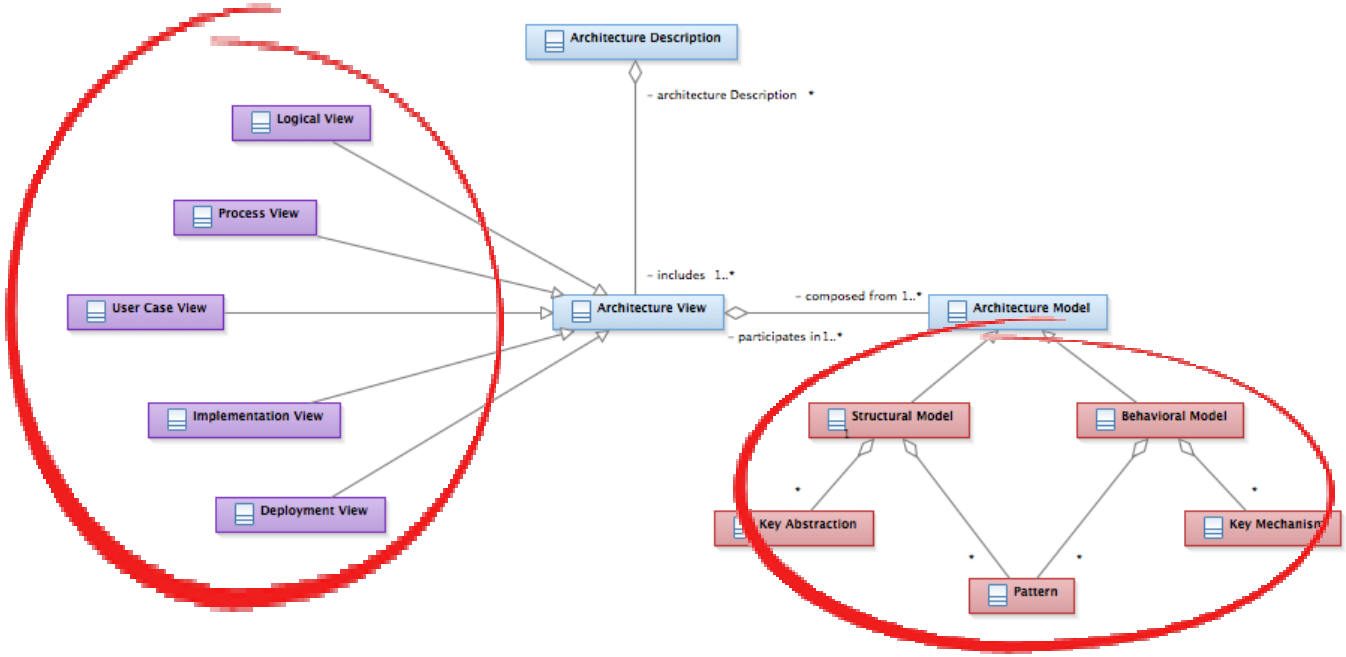
ISO/IEC WD4 42010 Architecture Description

Describing A System's Architecture



ISO/IEC WD4 42010 Architecture Description

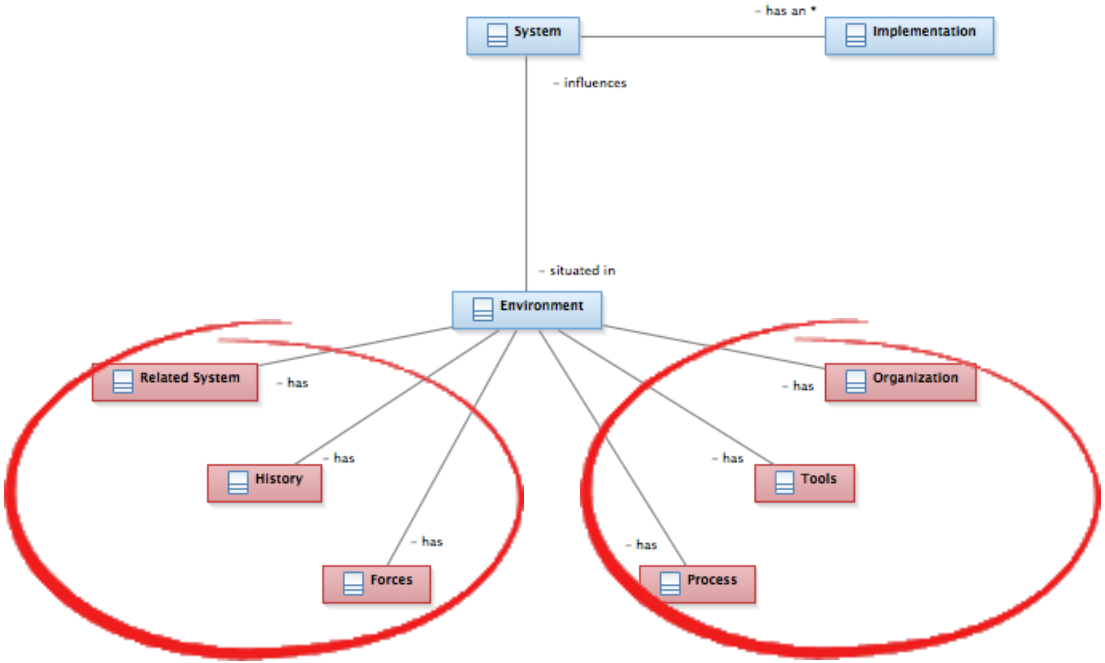
Describing A System's Architecture



Kruchten, "The 4+1 View Model of Software Architecture"

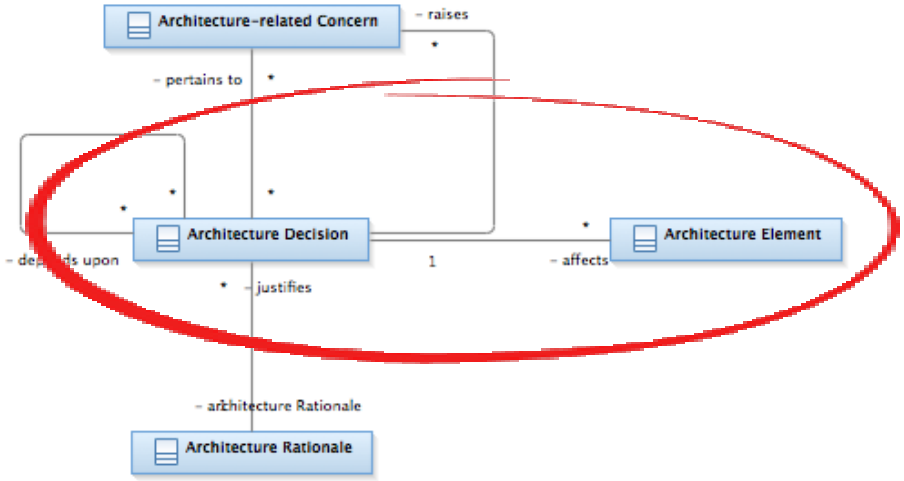
Booch, *The Handbook of Software Architecture*

Describing A System's Architecture



Booch, *The Handbook of Software Architecture*

Describing A System's Architecture



ISO/IEC WD4 42010 Architecture Description

Conducting An Archeological Dig

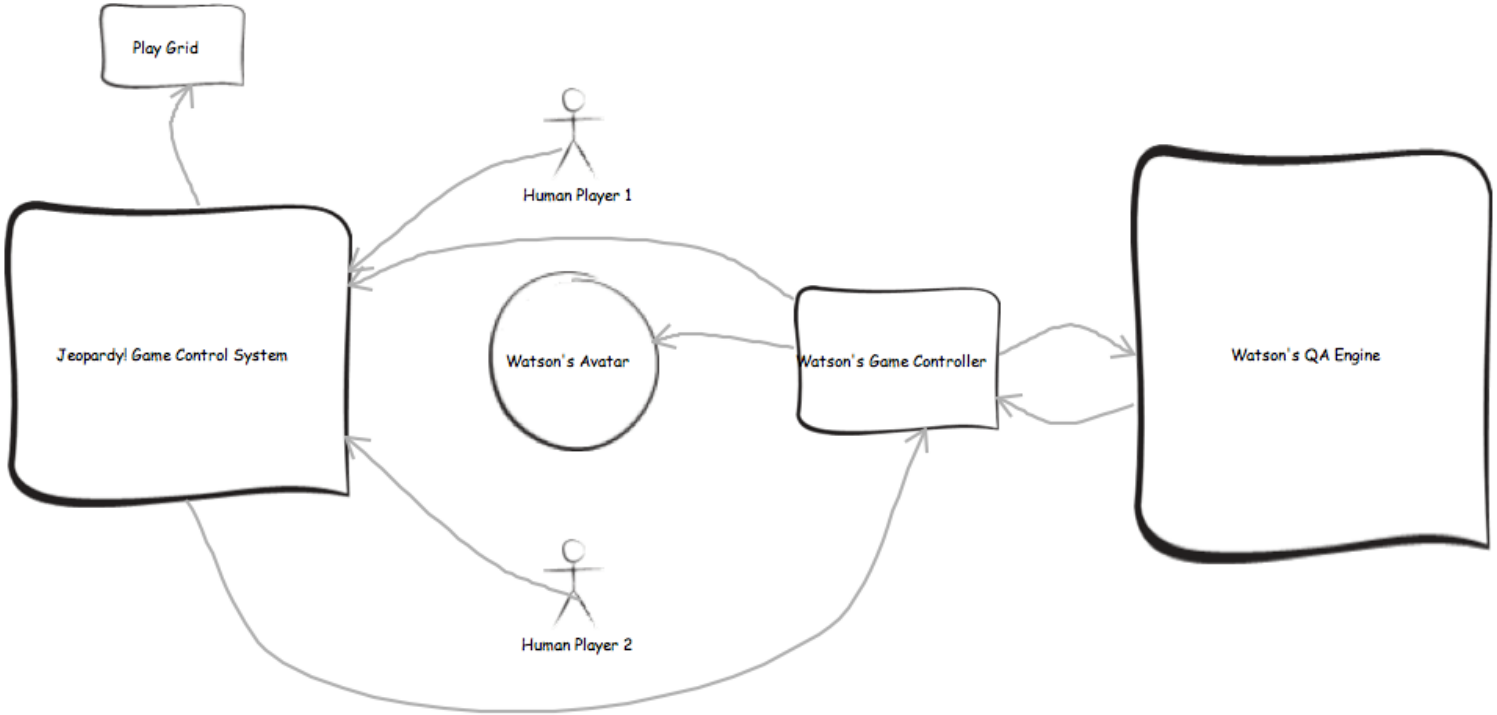
- Become immersed in the domain.
- Absorb all relevant development documents.
- Interview the development team.
- Study the source code.
- Interpret the architecture.
- Use the architecture description in anger.
- Repeat.

Booch, *The Handbook of Software Architecture*

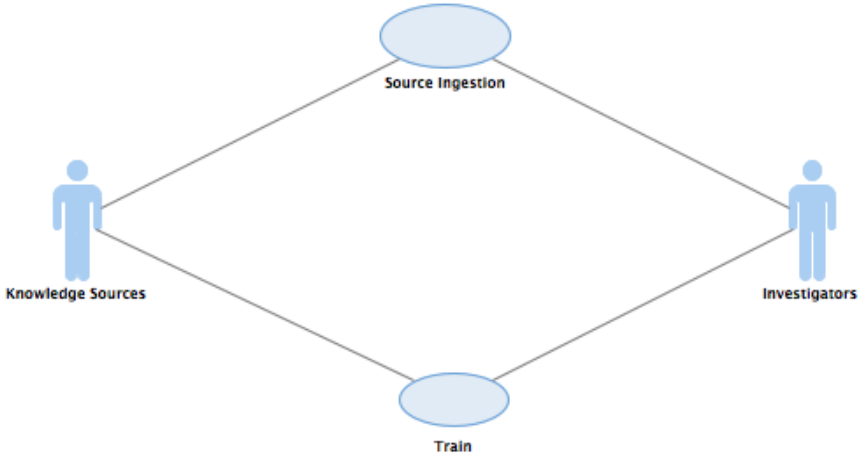
Statistics

Development Team	**25 people**
Project Duration	**4 years**
Software	**1,000,000+ SLOC**
	700K Java, 300K C++, plus other bits
	~ 130 components
Hardware	**90 IBM Power 750 servers**
	2880 Power7 cores @ 80+ TFLOPS
	20 TB memory
	10 Gbps network

Architecture: Context



Use Cases


Source Ingestion
Knowledge Sources
Investigators
Train


Questioner
AnswerQuestion

Key Design Decisions: Technical

- Use a pipe and filter architectural style.
- Acquire and apply heterogeneous data sources.
- Consider many possible candidate answers.
- Retrieve and evaluate multiple pieces of evidence in support of each candidate answer.
- Evaluate evidence along multiple dimensions.
- Combine evidence using machine learning.
- Build on UIMA-AS.

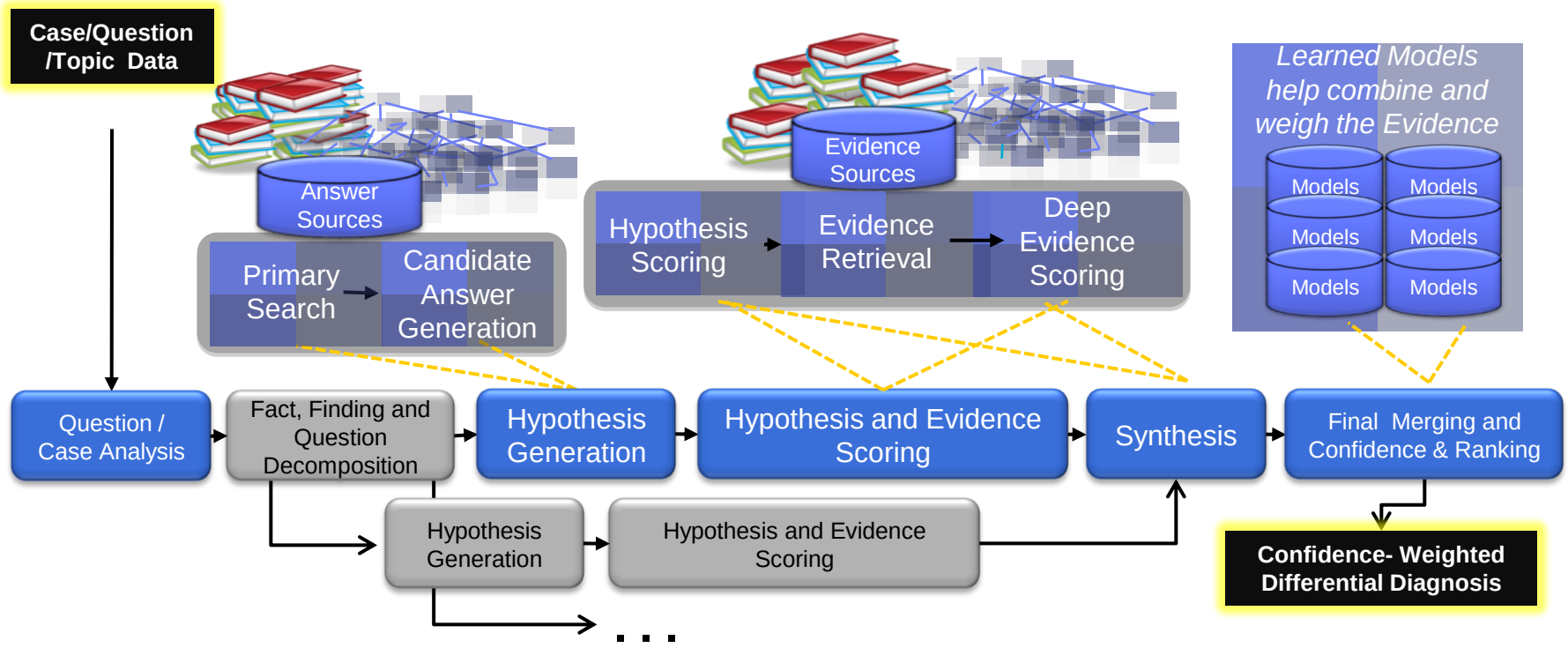
Key Design Decisions: Operational

- Permit the location of data to be configurable according to the needs of a specific deployment scenario.
- Directly map massive logical parallelism to massive deployment parallelism (but with mechanisms for easy reconfiguration).

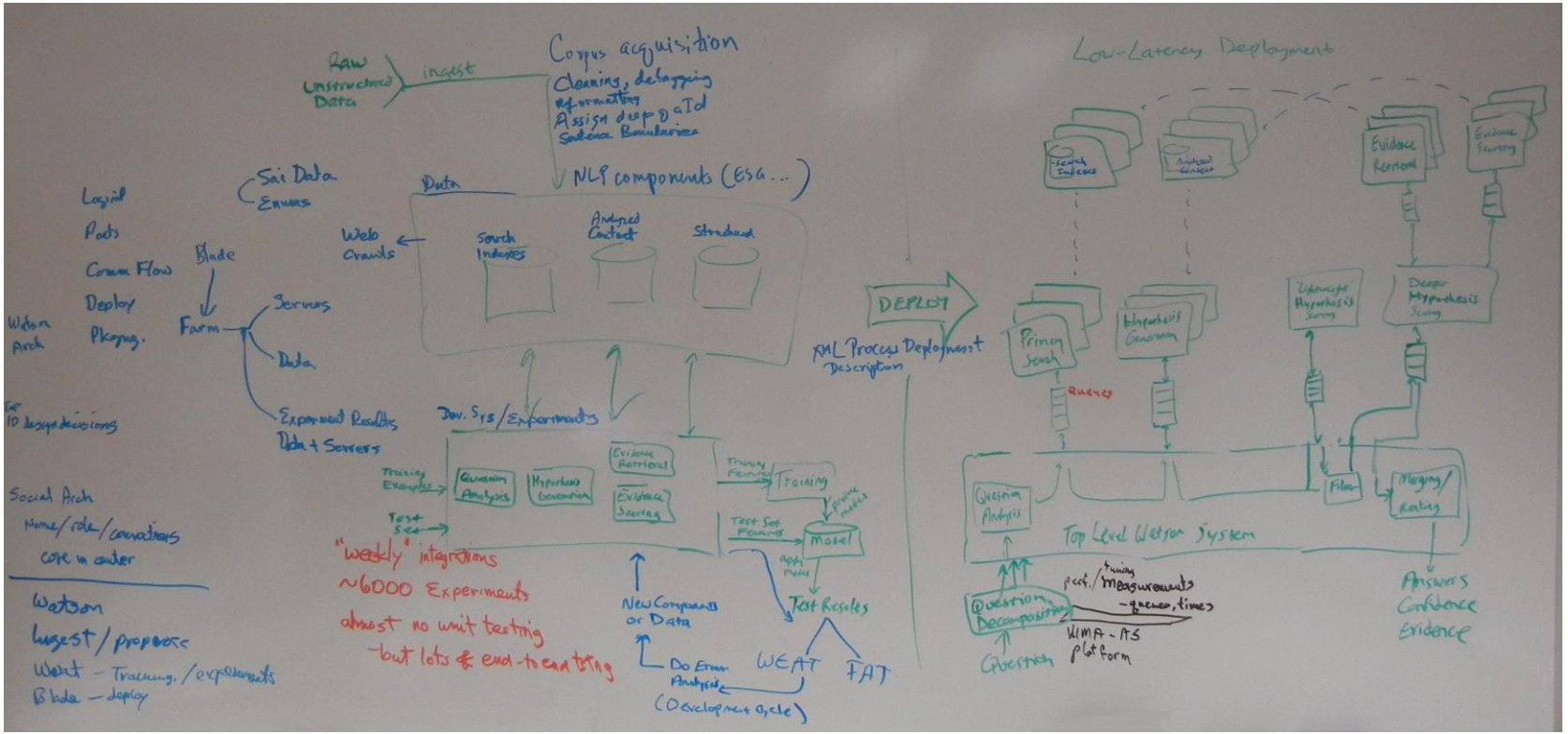
Key Design Decisions: Methodological

- Establish strong end-to-end metrics.
- Preserve considerable meta information for data, algorithms, and processes.
- Invest in tools to analyze Watson’s operation.

Architecture: In Their Own Words



Architecture: In Their Own Words



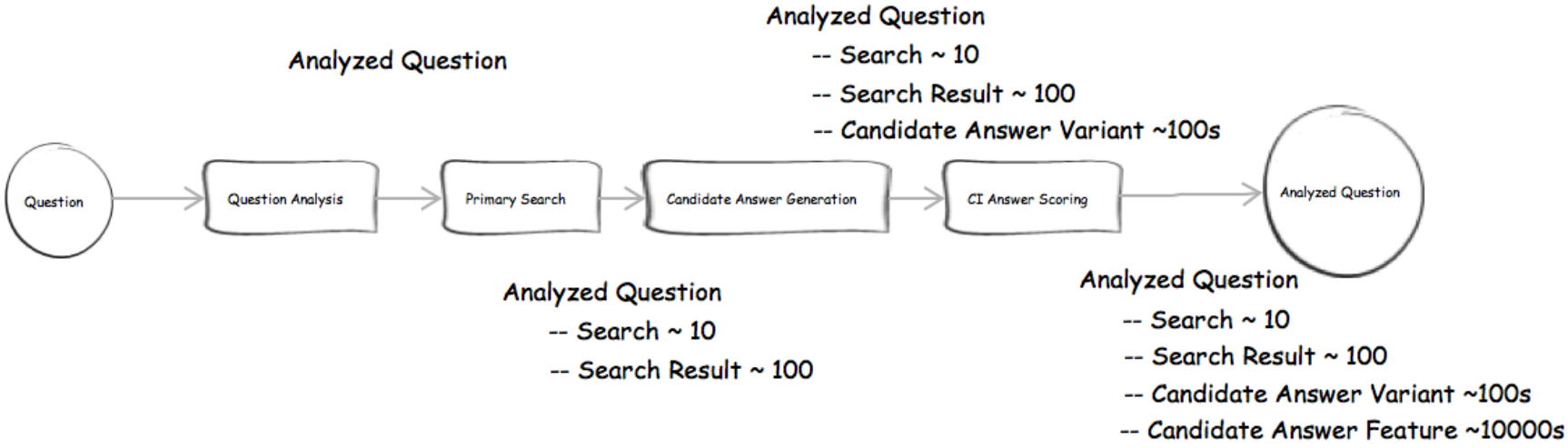
Sources

- Wikipedia/Wikiquote/Wiktionary/Wikibooks (The Free Encyclopedia) @ http://wikipedia.org
- YAGO2 (A Spatially and Temporally Enhanced Knowledge Base from Wikipedia) @ http://www.mpi-inf.mpg.de/yago-naga
- Dbpedia (Extracting Structured Information from Wikipedia) @ http://dbpedia.org
- WordNet (A Lexical Database for English) @ http://wordnet.princeton.edu
- Web expansion of many primary sources.
- Various licensed encyclopedias, dictionaries, books of quotations, and wire news.

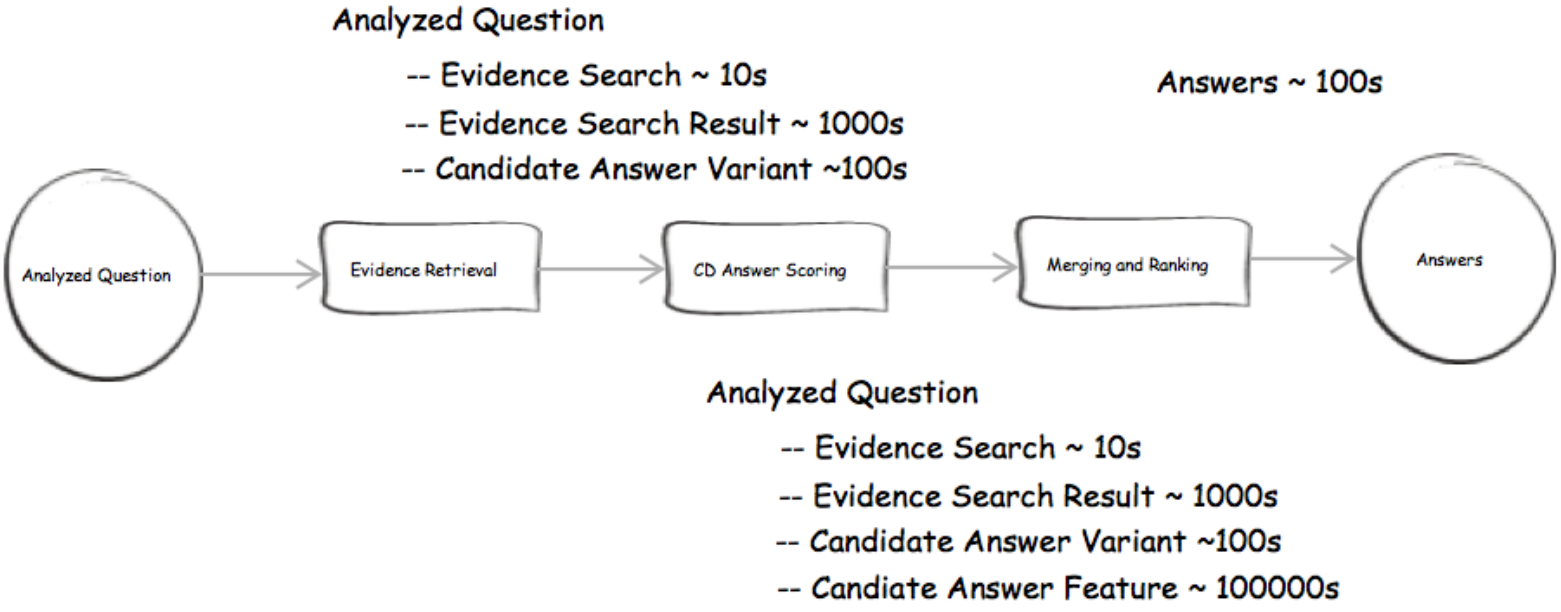
Test And Training Data

- J! Archive (A Fan-created Archive of Jeopardy!) @ http://www.j-archive.com

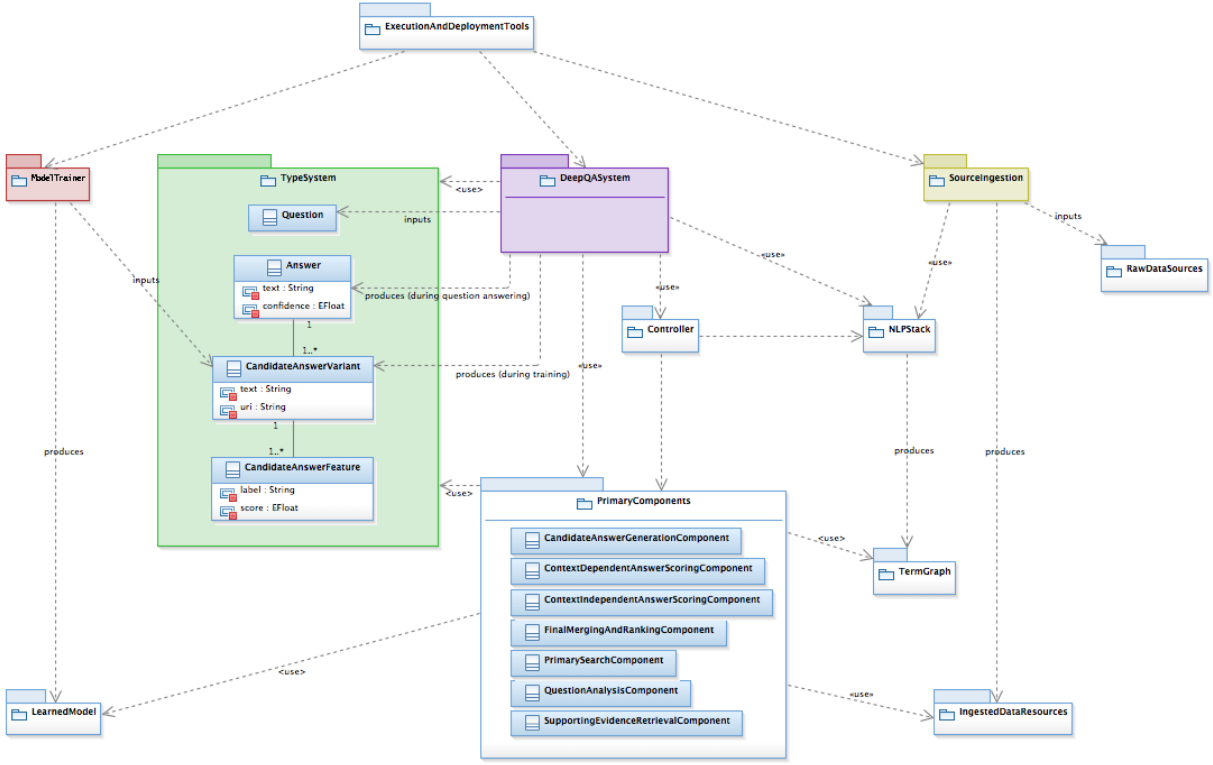
Logical View: Play Part 1



Logical View: Play Part 2



Logical View: Subsystems



Logical View: Key Abstractions

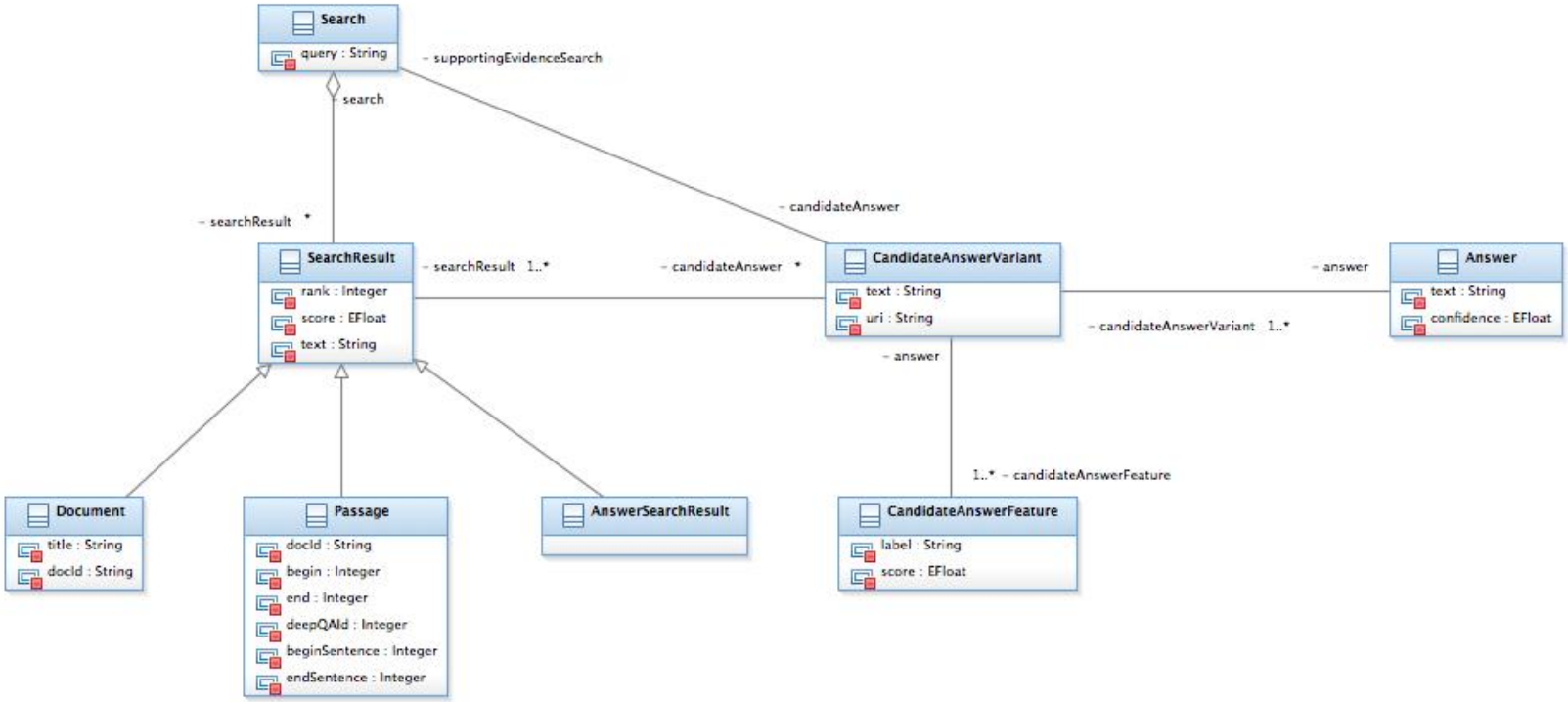
- UIMA Common Analysis Structure (CAS)
- CoreTypeSystem (aka Data Model).
- QuestionAnalysisTypeSystem.
- Terms.
- IngestedDataResources.
- PrimaryComponents.
- ModelTrainer.

Logical View: CAS

- The common data structure shared by all UIMA analytics to represent unstructured information being analyzed (artifact) as well as metadata produced by the analysis workflow (artifact metadata), encompassing:
  - The artifact (the object being analyzed).
  - The subject of the analysis (one or more views of the artifact).
  - A type system description (including types, subtypes, and features).
  - Metadata (describing the artifact or a region of the artifact).
  - Index repository (supporting efficient access to and iteration over the results of analysis).

http://uima.apache.org/

Logical View: Core Type System/Data Model

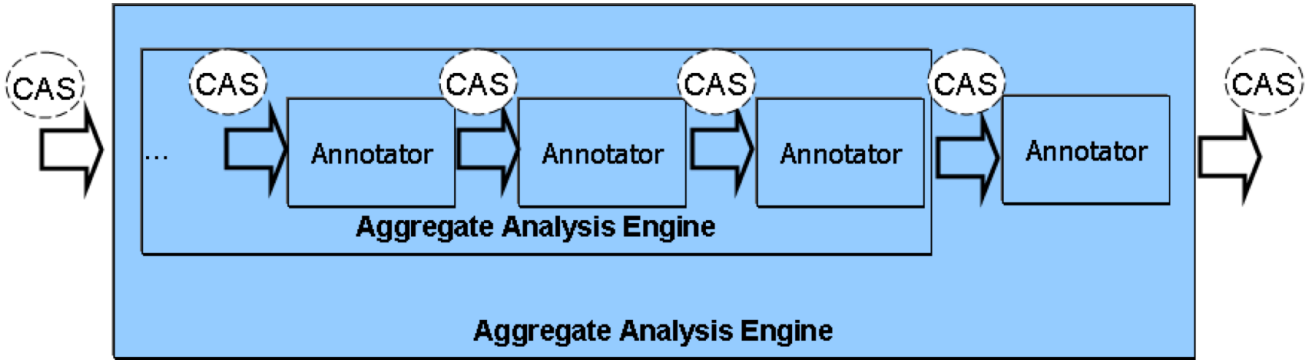


Logical View: Key Mechanisms

- UIMA.
- Question analysis.
- Primary search.
- Candidate answer generation.
- Shallow and deep scoring.
- Merging and ranking.

Logical View: UIMA

- UIMA is a software architecture which specifies component interfaces, data representation, design patterns, and development roles for creating, describing, discovering, composing, and deploying multi-modal analysis capabilities. The principal objective of the UIMA specification is to support interoperability among analytics.



http://uima.apache.org/

Logical View: UIMA

- Watson uses over 100 annotators:
  - Basic (parsers, co-reference, dictionary lookup, named entity detectors, true casers).
  - LAT (2500 different lexical answer types).
  - Decomposition
  - Special question form identification.
  - Language translation.
  - Temporal logic.
  - Geospatial.

Logical View: UIMA

- Watson's annotators reveal different kinds of features:
  - Temporal (events and people happen during particular times and have likely life extents).
  - Location (events happen in particular places; places are located in or near other places).
  - Passage support (passages relate key entities to a candidate answer).
  - Shallow evidence (passage superficially aligns with question text).
  - Deep evidence (candidate answer is understood to be in the right logical relationship with key entities).

Logical View: UIMA

- Watson's annotators reveal different kinds of features:
  - Classification (answers should be the right type or class).
  - Popularity (answer is popularly associated with key elements in question)
  - Source reliability (sources supporting answer are learned to be reliable).
  - Predicate role (candidate answer plays the right role in key predicate).
  - Document support (document discusses fact in the context of the answer).
  - Hidden link (candidate answer and question entities share a common thread that is logically aligned).
  - Pun (candidate answer and question entities are associated with each other in one or more ways, such as sounds like, part of, synonym of, etc).

Logical View: Candidate Answer Generation

- Produces candidate answers without limitation on the type of answers produced.
  - Type coercion components attempt to relate the candidate answers to the lexical answer type.

http://www.lemurproject.org/
http://lucene.apache.org/

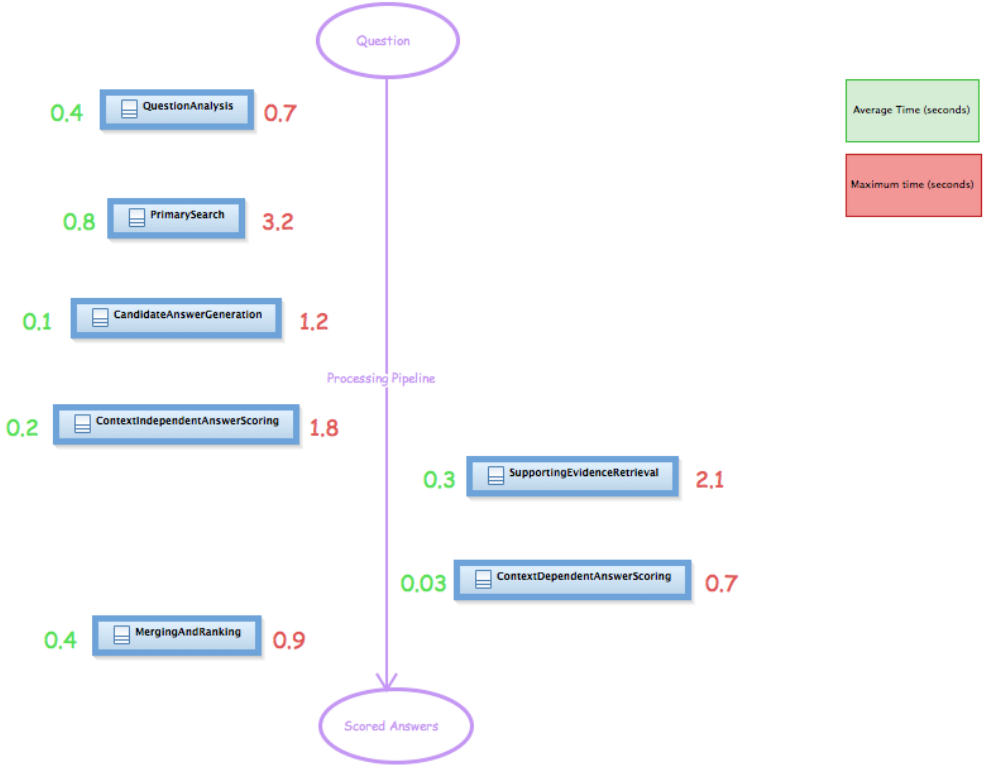
Logical View: Shallow And Deep Scoring

- Conducts both shallow (content independent and deep (context dependent) scoring.
  - Candidate answers that do not pass shallow scoring are passed directly to merging and ranking.
  - Candidate answers that pass shallow scoring continue to learned deep evidence retrieval and scoring, then passed on to merging and ranking.

http://www.lemurproject.org/
http://lucene.apache.org/

Process View: Processing Pipeline

Process View: Timing



Process View: Key Abstractions

- Concurrent-aware POJOs.
  - May be mixed and matched as to location and grouping.
  - ~70 Mb data flow per question within UIMA-AS transactions.
  - ~140 Mb data flow per question within Indri transactions.

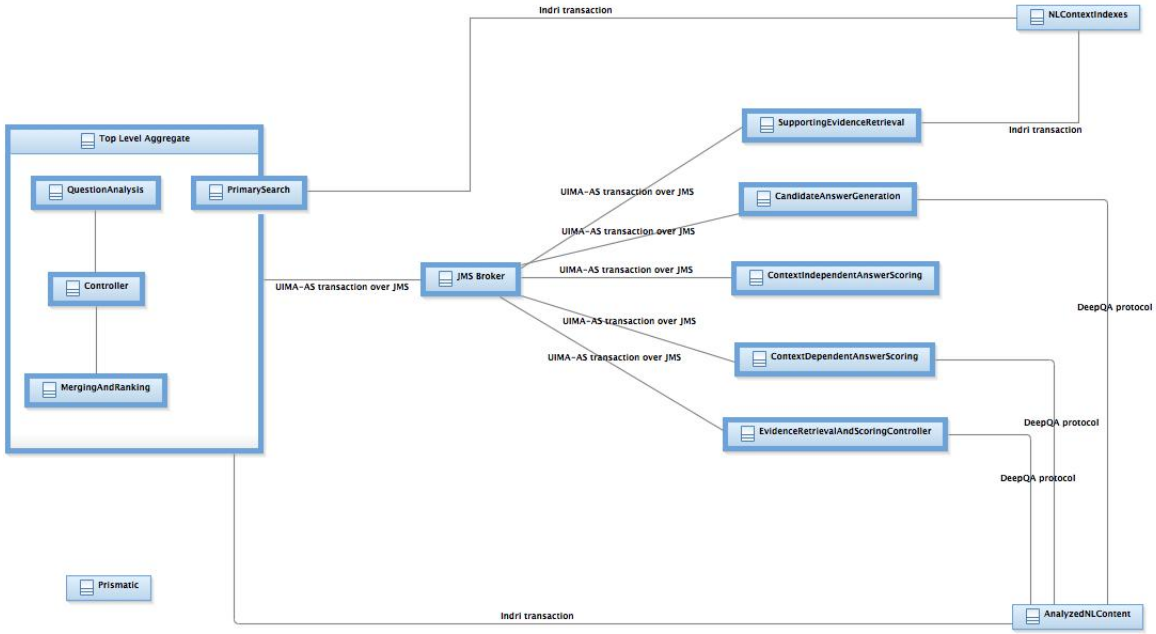
http://uima.apache.org/doc-uimaas-what.html

Process View: Key Mechanisms

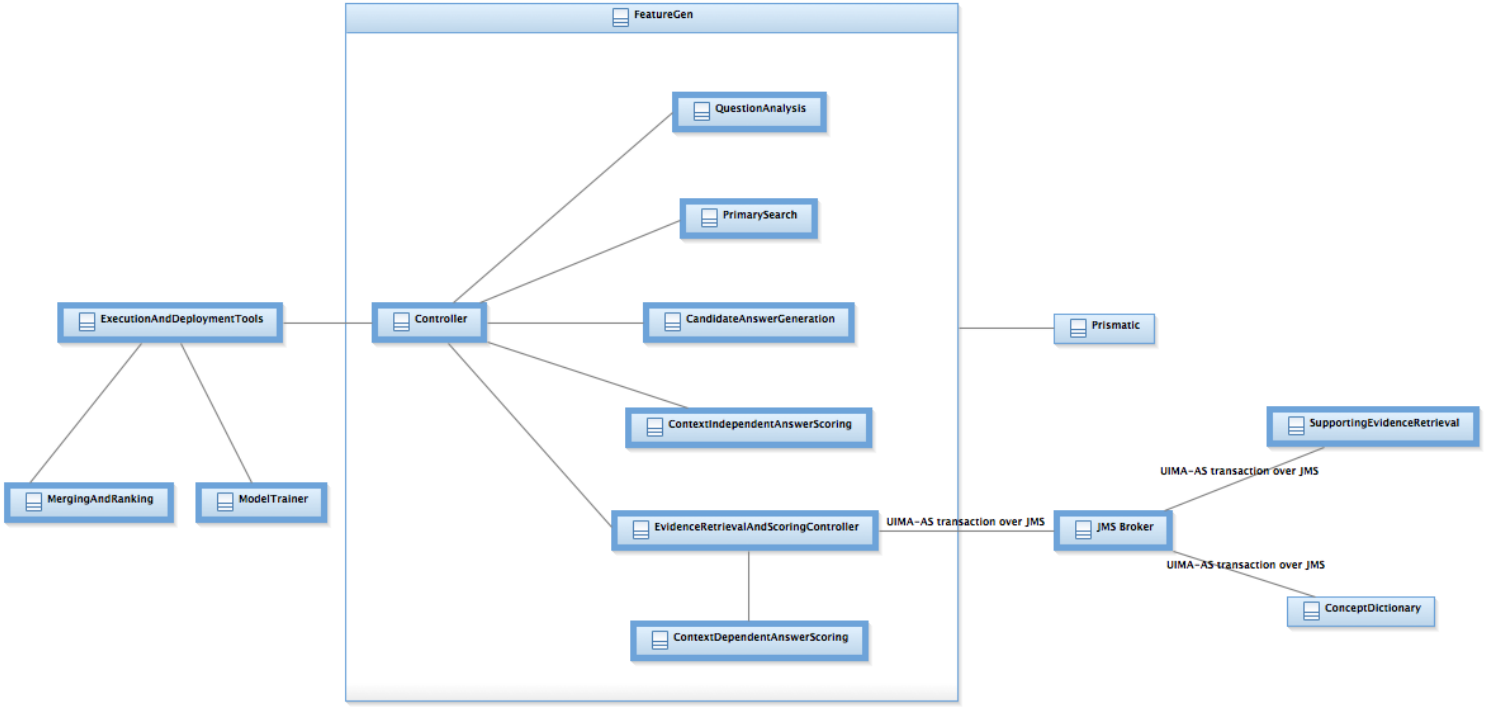
- UIMA-AS.
  - A set of general capabilities for achieving scale out, built upon UIMA.
- UIMA CAS Multiplier and CAS pools.
  - Expand/consolidate CAS envelopes efficiently across multiple configurable flows.
- Three communication protocols:
  - UIMA-AS transactions across JMS.
  - DeepQA protocol for accessing large in-memory datasets.
  - Indri distributed search protocol.

http://uima.apache.org/doc-uimaas-what.html

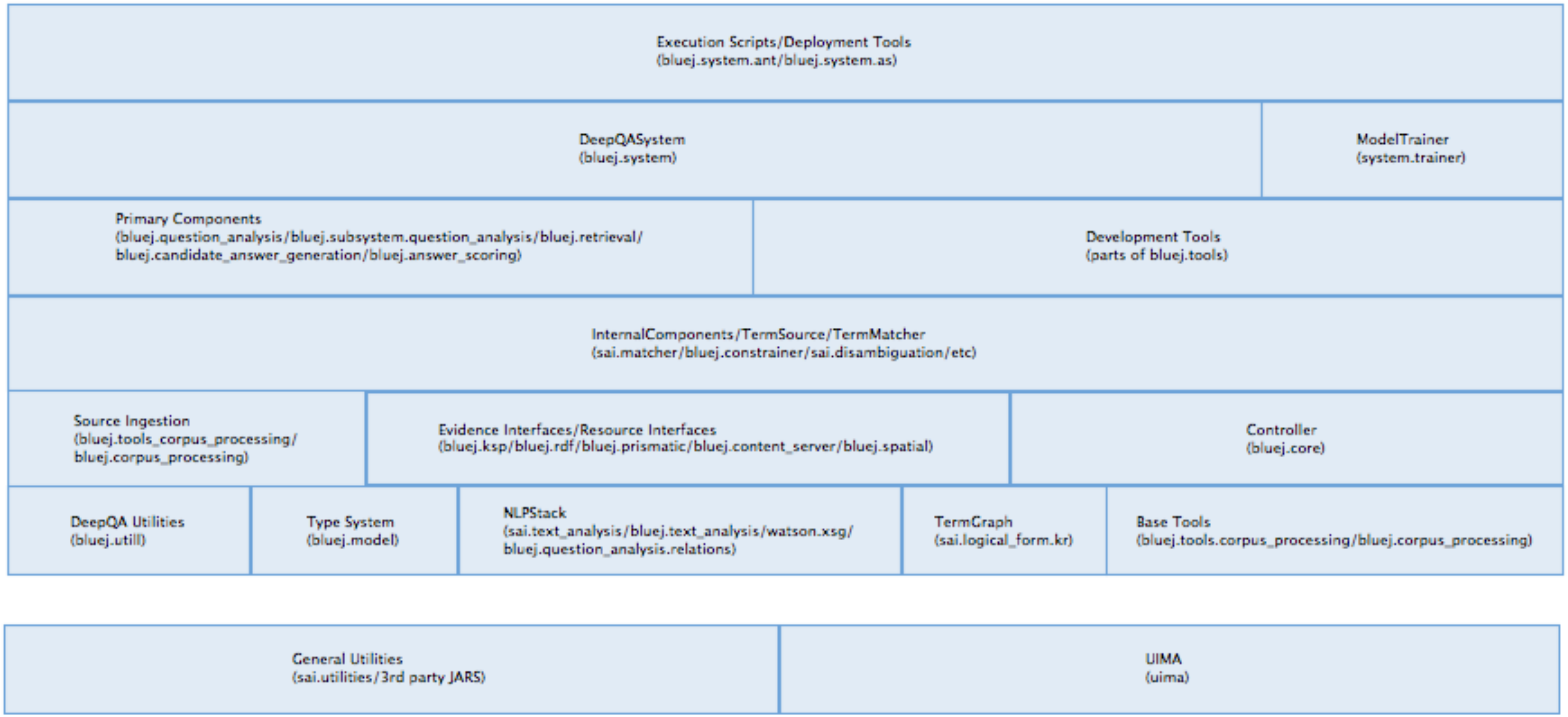
Process View: Low-latency Production



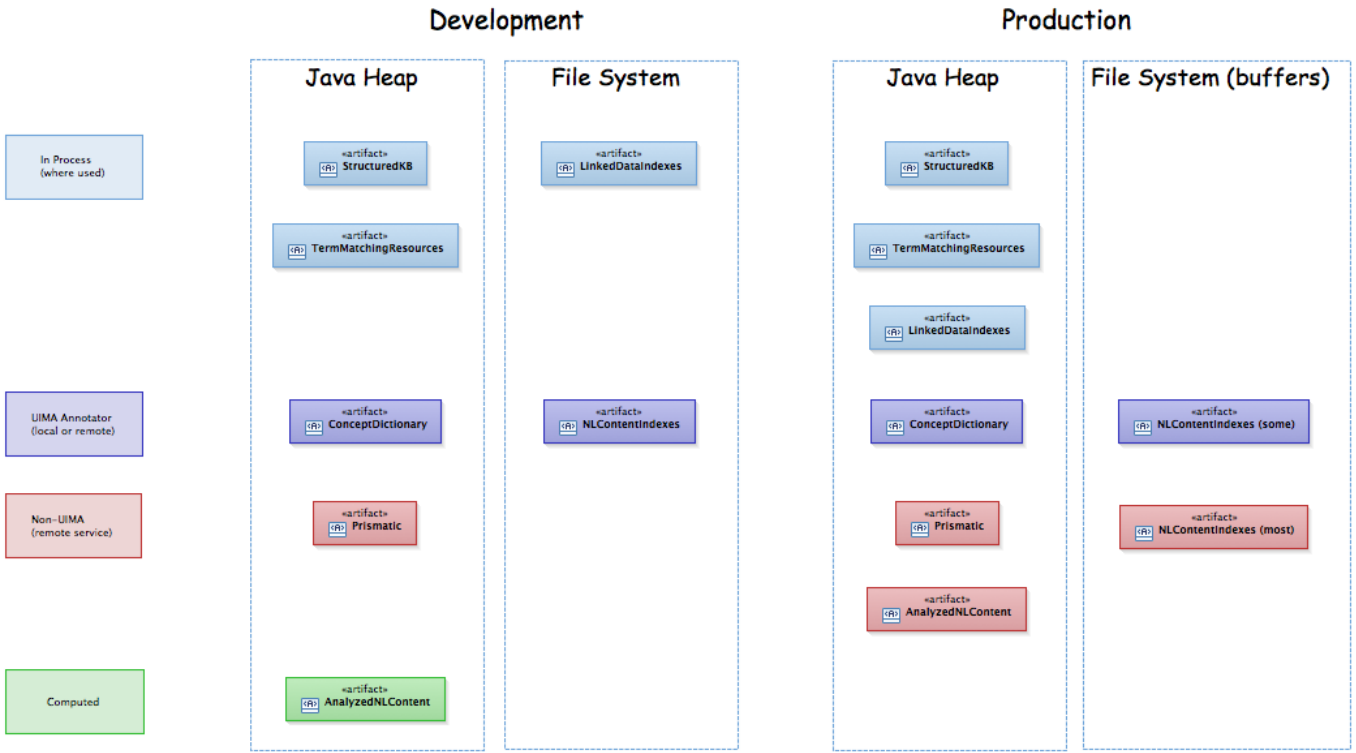
Process View: High-throughput Development



Component View: Code Layering



Component View: Data Storage

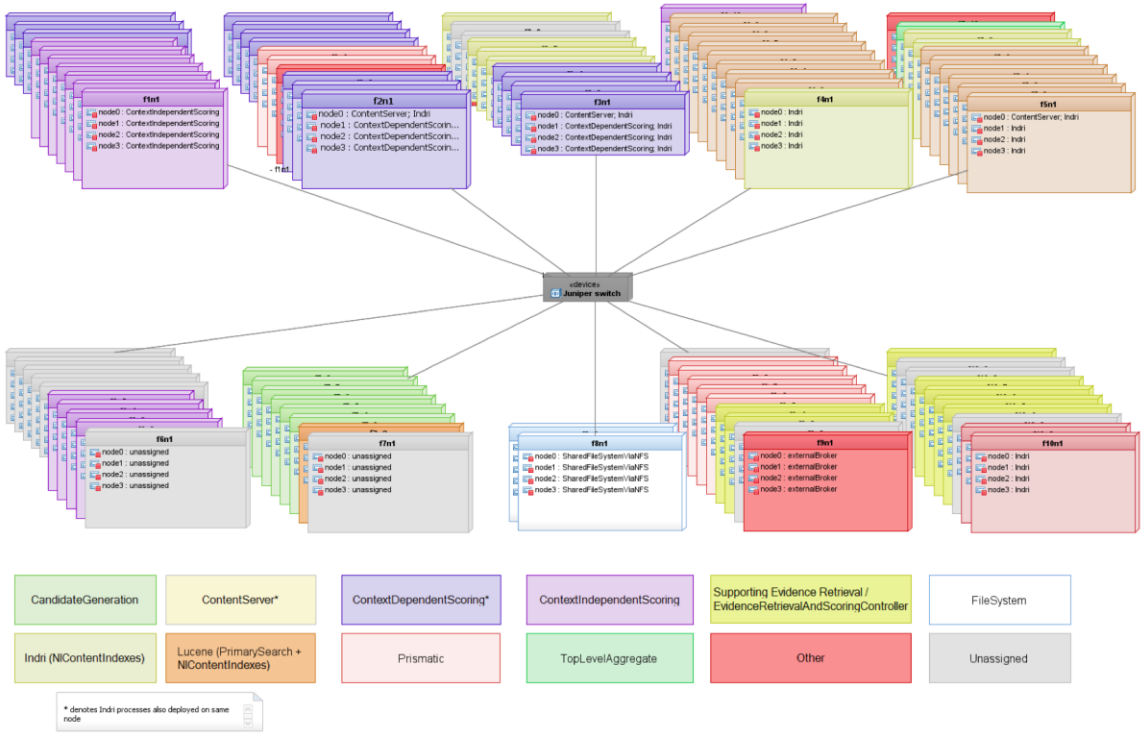


Deployment View: Watson

- 90 IBM Power 750 servers.
  - 4 Power7 processors/server.
  - 8 cores/processor.
  - 10 TB memory/server.
  - Linux.
- SONAS storage @ 20 TB.
- Juniper switch @ 10 Gbps.
- 2 20-air conditioning units.

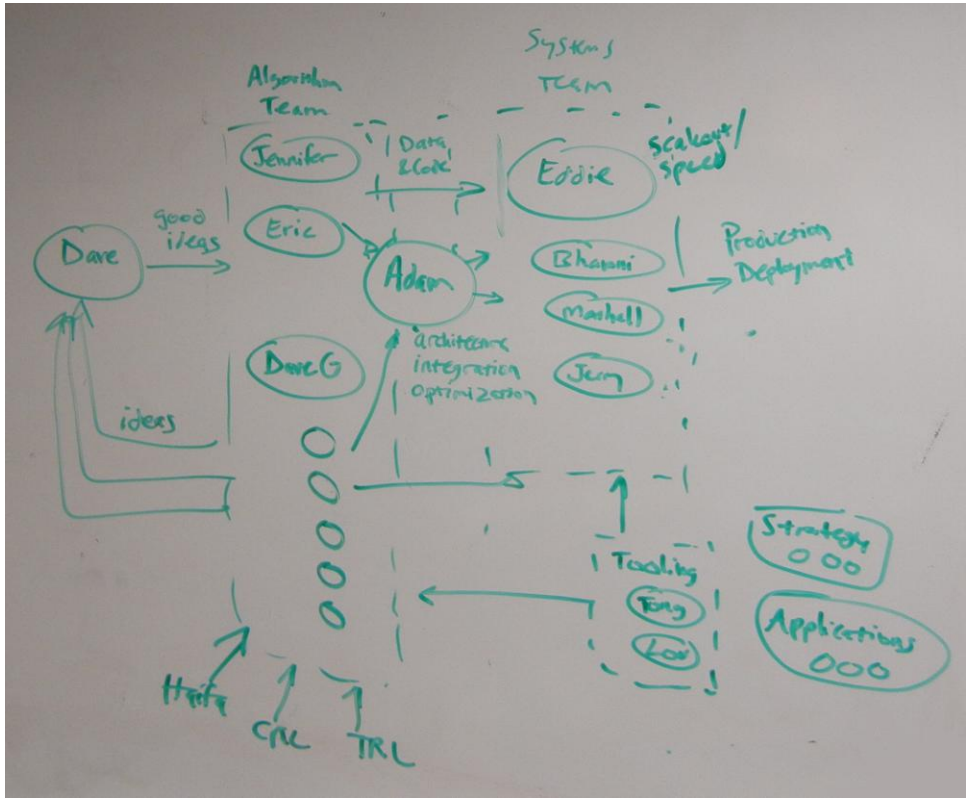
http://www.hypeframework.org/blog/content/ibm-watson-and-the-jeopardy-challenge/

Deployment View: Watson



Deployment View: Watson

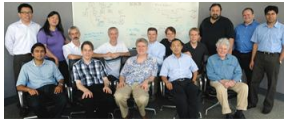
Organization: In Their Own Words



Organization

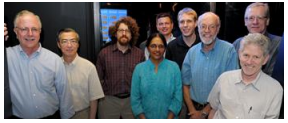
- David Ferrucci

- Algorithms Team

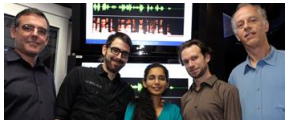


- Strategy Team

- Systems Team



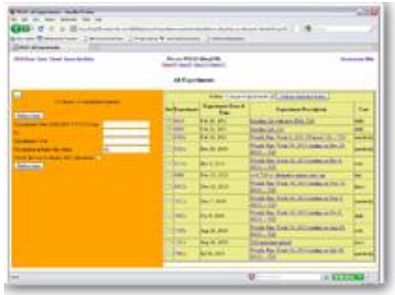
- Speech Team



- Annotations Team

- Project Management

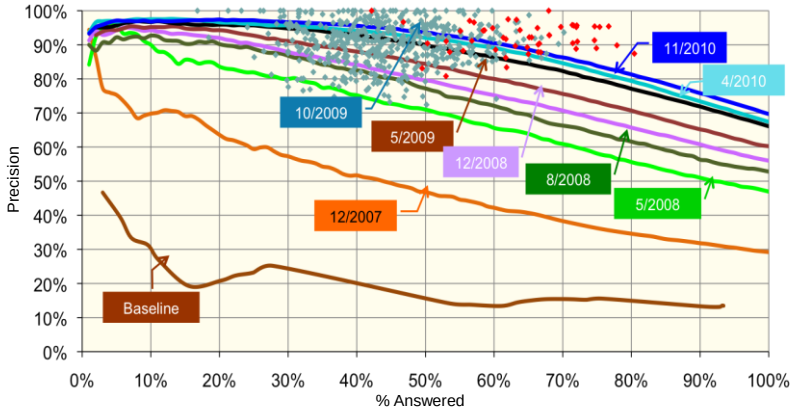
Tools



- Eclipse.
- Subversion -> RTC.
- Watson Error Analysis Tool (WEAT).
- Feature Analysis Tool (FAT).
- BlueJ Automatic Distributed Execution Environment tools (BAIDE)
- Data repository tools.

Process

- Agile development.
- War room setting with continuous collaboration.
- Weekly integration.
- Results driven with end to end regression testing.
- ~ 6,000 experiments
- 10 gigabits of test data/week.



Watson’s Future: New Development

- Hygienic.
  - Refactoring; elevating certain features to first-class architecture elements; performance and platform improvements; common configuration management.
- Research.
  - Greater introspection; dialoging; video & speech input; real time data input.
- Business.
  - Product line architecture.

Software. Everyware.

www.ibm.com/software/rational