Vulcan and LarKC

LarKC Kickoff Meeting

Dr. Mark Greaves
Vulcan Inc.
markg@vulcan.com +1 (206) 342-2276

Talk Outline

- Why I'm So Excited about LarKC
- Systems AI – Vulcan's Halo Program
  - Deep Reasoning over the AP problem
  - Systems AI: Unified Knowledge Acquisition, Representation/Reasoning, and Question Answering
- Reasoning in a System – Lessons From Halo for Dynamic Reasoning on the Semantic Web

Talk Outline

- Why I'm So Excited about LarKC
- Systems AI – Vulcan's Halo Program
  - Deep Reasoning over the AP problem
  - Systems AI: Unified Knowledge Acquisition, Representation/Reasoning, and Question Answering
- Reasoning in a System – Lessons From Halo for Dynamic Reasoning on the Semantic Web

The L209 LARK Rocket Launcher

From “Resistance: Fall of Man” for the Sony PS/3

Technology Focus: Agile Computing meets KR&R

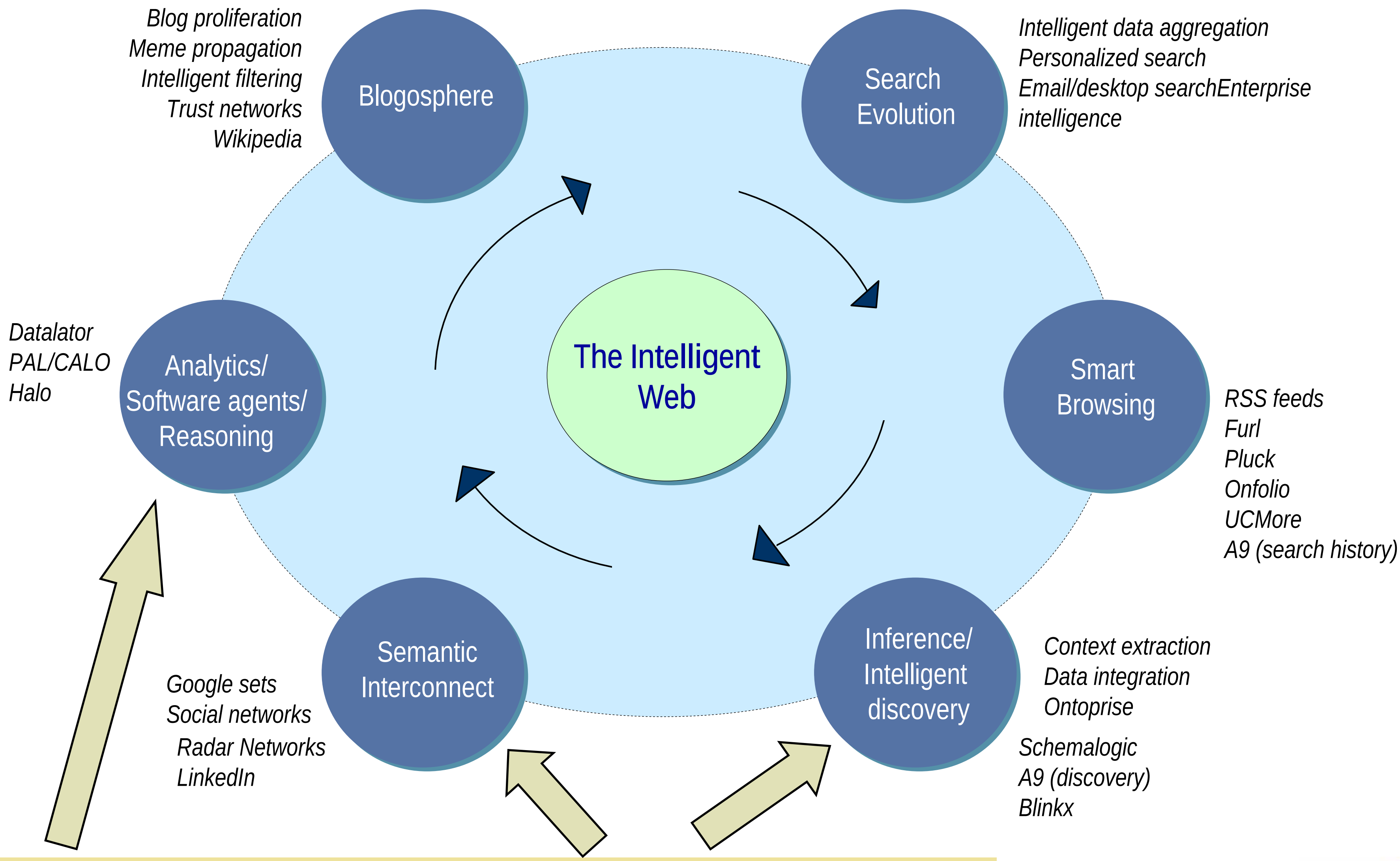
- My time at DARPA: 5/2001 to 5/2004
  - A vision of *agile computing* – a robust distributed infrastructure for dynamic reliable computing
    - Oriented towards responsiveness rather than prespecified optimality
    - Provides "illity" and QoS arguments
    - Supports adaptive, survivable workflows
    - Leverages local rules over global ones
    - Is a step beyond interoperability
  - Agile computing requires objects, relations, workflows, and other semantic/intentional notions
- Currently with Vulcan (Seattle, WA)
  - Vulcan (www.vulcan.com) is the corporate vehicle through which Paul Allen manages his assets
  - Areas include music, movies, sports teams, aerospace, philanthropy, personal tech, new energy, cable TV, venture capital, AIBS, AI…
  - I am responsible for the AI/KR&R research portfolio, including Project Halo and Semantic Wikis / Web 2.0
    - NeOn, STI2, and AIBS responsibilities

DARPA Activities

	Data	Process	Represen-tation
Programs and Seedlings	▪ DARPA Agent Markup Language (DAML)	▪ UltraLog ▪ Semantic Enabling and Exploitation (SEE) ▪ Network-Centric Logistics (NCL)	▪ Fog Light ▪ OMNI
Core Technologies	▪ Description Logics ▪ KR&R Languages ▪ Rule Systems ▪ Web Languages ▪ Ontologies, Taxonomies	▪ Software Agents ▪ Semantic Web Services ▪ Peer-to-Peer Algorithms ▪ Distributed Planning ▪ Complex Adaptive Systems and Control	▪ Situation Theory ▪ Task Planning Systems ▪ Problem Isomorph-ism ▪ Semantic Web Services

VULCAN

The Larger KR Environment: The Evolving Web



Web-Scale Reasoning: Scalable, Tolerant, Dynamic!

VULCAN

The Larger KR Environment: Web Science

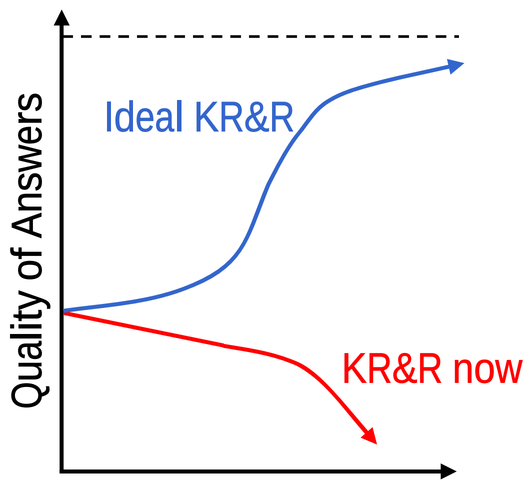
- The evolving web is driving a new computing paradigm
  - The Web has given us more than HTML/HTTP
  - Computation is situated, 24x7, and embedded in the environment
    - Distributed processes are linked through web services, ebXML, J2EE, application servers, Microsoft .NET…
    - Everything is best-effort
  - Turing machine theoretical models are yielding to stream and data-driven computing
    - *Algorithm* becomes *control surface*
    - *Data-centric* becomes *process-centric*
    - *Halting conditions* become *evolutionary stability* and *lifecycle properties*
    - *Total functions* become *partial functions*
    - *Determinism* becomes *stochastic optimization*
- We have a vast amount of instance (A-box) data
  - Web 2.0 authoring systems, information harvesting technologies, and the sensor revolution are supplying massive quantities of data
- We have a small-but-growing amount of ontological data, mostly fairly weak
- Effective reasoning/analytics is the new bottleneck
- We see this in microcosm in Halo; we look for solutions from LarKC

VULCAN

Traditional KR and the Google Property

- We seek KR systems that have the "Google Property:" *they get (much) better as they get bigger*
  - Google PageRank™ yields better relevance judgments as it indexes more pages
  - Current KR&R systems have the antithesis of this property
- So what are the components of a scalable KR&R system?
  - Distributed, robust, reliable infrastructure
  - Multiple linked ontologies and points of view
    - Single ontologies are feasible only at the program/agency level
  - Mixture of deep and shallow knowledge repositories
  - Simulations and procedural knowledge components
    - "Knowing how" and "knowing that"
  - Embrace uncertainty, defaults, context, and nonmonotonicity in all components
  - Facts of life for large-scale KBs – you don't know what you know, "facts" go away, contradiction is rampant, computing must be resource-aware, surveying the KB is not possible

KR&R Goals



KR&R System Scale
(Number of Assertions
Number of Ontologies/POVs
Number of Rules
Linkages to other KBs
Reasoning Engine Types ...)

Scalable KR&R Systems should look just like the Web!!

(coupled with great question-answering technology)

VULCAN

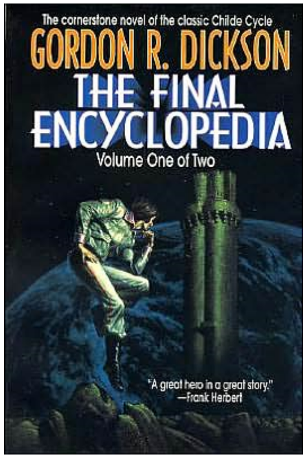
Talk Outline

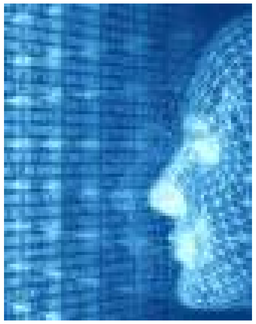
- Why I'm So Excited about LarKC
- Systems AI – Vulcan's Halo Program
  - Deep Reasoning over the AP problem
  - Systems AI: Unified Knowledge Acquisition, Representation/Reasoning, and Question Answering
- Reasoning in a System – Lessons From Halo for Dynamic Reasoning on the Semantic Web

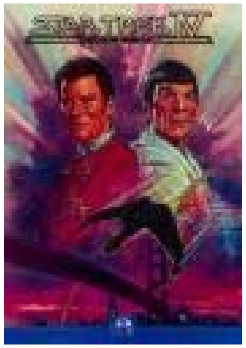
VULCAN

Envisioning the Digital Aristotle for Scientific Knowledge

- Inspired by Dickson's Final Encyclopedia, the HAL-9000, and the broad SF vision of computing
  - The "Big AI" Vision
- The volume of scientific knowledge has outpaced our ability to manage it
  - This volume is too great for researchers in a given domain to keep abreast of all the developments
  - Research results may have cross-domain implications that are not apparent due to terminology and knowledge volume
- "Shallow" information retrieval and keyword indexing systems are not well suited to scientific knowledge management because they cannot reason about the subject matter
  - Example: "What are the reaction products if metallic copper is heated strongly with concentrated sulfuric acid?" (Answer: Cu^{2+}, SO_2(g), and H_2O)
- Response to a query should *supply the answer* (possibly coupled with conceptual navigation) rather than simply list 1000s of possibly relevant documents





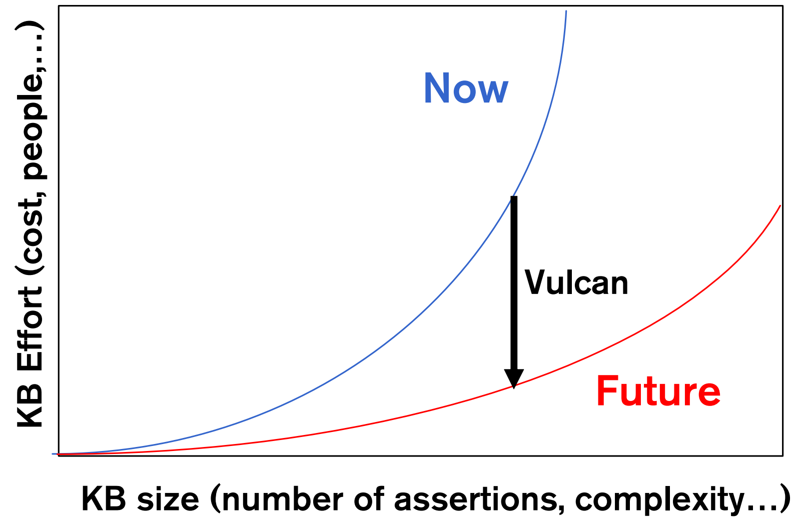


DIGITAL ARISTOTLE

VULCAN

How do we get to the Digital Aristotle?

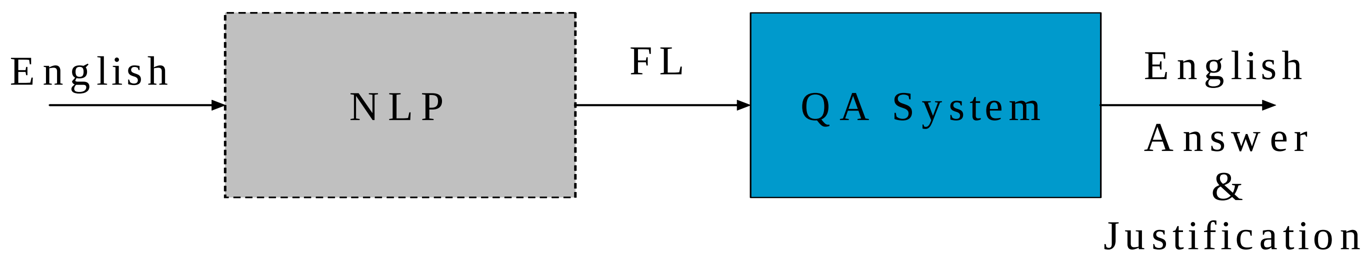
- Vulcan's Goals for its KR&R Research Program
  - Address the problem of scale in Knowledge Based Systems
    - *KBS size scaling*: Focus on the ontological information, not instance information
      - Inspired by DAML, Cyc, and commercial rule-based systems
    - *KBS cost scaling:* Use large numbers of SMEs in KB construction and maintenance
      - Inspired by DARPA RKF and Web 2.0
  - Have high impact
    - Show the Digital Aristotle is possible
    - Change our experience of the Web
    - Have quantifiable, explainable metrics
  - Have commercial offramps



Halo is one concrete program that addresses these goals

The Halo Pilot

- In 2004, Vulcan funded a six-month effort to determine the state-of-the-art in fielded "deep reasoning" systems
  - Can these systems support reasoning in scientific domains?
  - Can they answer novel questions?
  - Can they produce domain appropriate answer justifications?
- Three teams were selected, and used their best available tech
  - SRI, with Boeing Phantom Works and UT-Austin
  - Cycorp
  - Ontoprise GmbH
- No NLP in the Pilot



The Halo Pilot Domain

- 70 pages from the United States AP-chemistry syllabus (Stoichiometry, Reactions in aqueous solutions, Acid-Base equilibria)
  - Small and self contained enough to be do-able in a short period of time, but large enough to create many novel questions
  - Complex “deep” combinations of rules
  - Standardized exam with well understood scores (AP1-AP5)
  - Availability of experts for exam generation and grading
  - Chemistry is an exact science, quite “monotonic”, and (at this level) needs no novel KR
  - No undo reliance on graphics (e.g., free-body diagrams)

- Example: Balance the following reactions, and indicate whether they are examples of combustion, decomposition, or combination

  (a) $C_4H_{10} + O_2 \rightarrow CO_2 + H_2O$

  (b) $KClO_3 \rightarrow KCl + O_2$

  (c) $CH_3CH_2OH + O_2 \rightarrow CO_2 + H_2O$

  (d) $P_4 + O_2 \rightarrow P_2O_5$

  (e) $N_2O_5 + H_2O \rightarrow HNO_3$

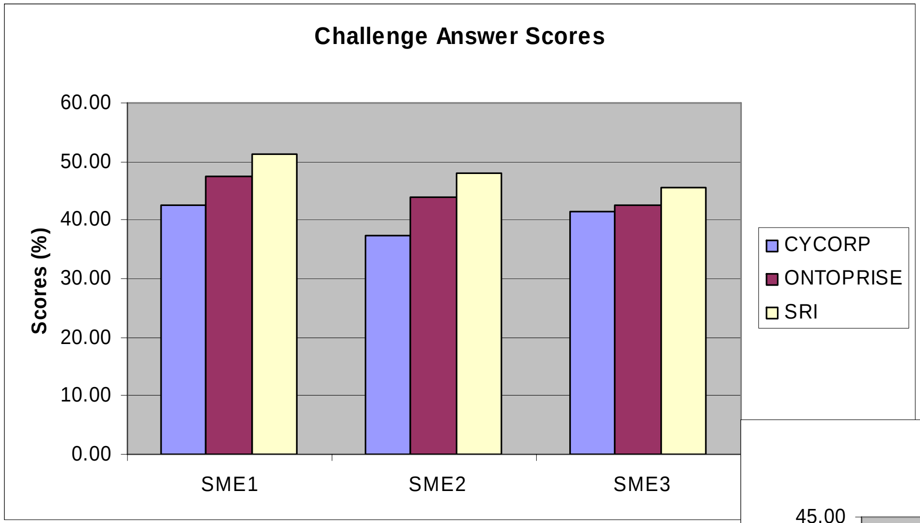
Halo Pilot Evaluation Process

- Evaluation
  - Teams were given 4 months to formulate the knowledge in 70 pages from the AP Chemistry syllabus
  - Systems were sequestered and run by Vulcan against 100 novel AP-style questions (hand coded queries)
  - Exams were graded by chemistry professors using AP methodology

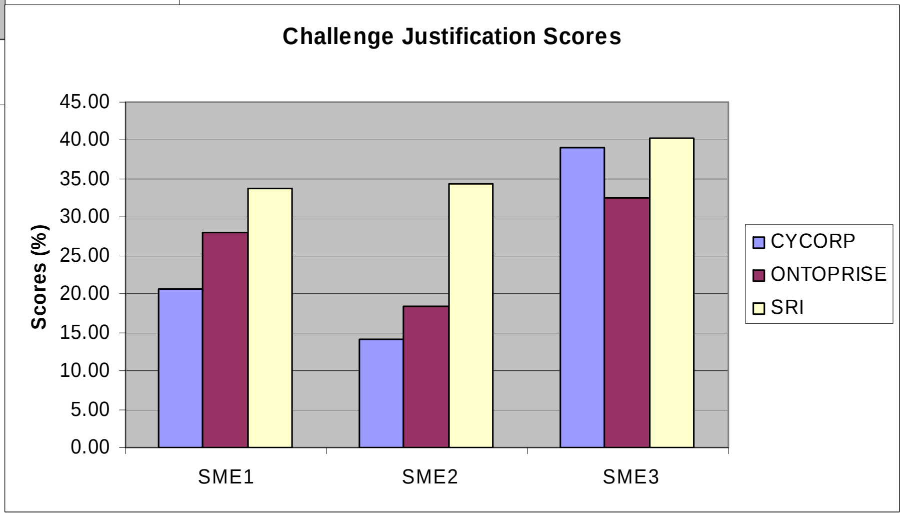
- Metrics
  - **Coverage:** The ability of the system to answer novel questions from the syllabus
    - What percentage of the questions was the system capable of answering?
  - **Justification:** The ability to provide concise, domain appropriate explanations
    - What percentage of the answer justifications were acceptable to domain evaluators?
  - **Query encoding:** The ability to faithfully represent queries
  - **Brittleness:** What were the major causes of failure? How can these be remedied?

VULCAN

Halo Pilot Results



Best scoring system scored the equivalent of an AP2-3 (on our restricted syllabus)



Full Details in *AI Magazine* 25:4, “Project Halo: Towards a Digital Aristotle”

VULCAN

From the Halo Pilot to the Halo Project

- Halo Pilot Results
  - Much better than expected results on a very tough evaluation
  - Most failures attributed to modeling errors due to contractors' lack of domain knowledge
  - Expensive: O($10,000) per page, per team

- Project Halo Goal: To determine whether tools can be built to facilitate robust knowledge formulation, query and evaluation by domain experts, with ever-decreasing reliance on knowledge engineers
  - Halo Guiding Hypotheses:
    - Textbook-based Knowledge Acquisition: SMEs can use textbooks to build robust question-answering systems that demonstrate excellent coverage of a given syllabus, and the ability to answer novel questions and produce readable domain appropriate justifications
    - Limited NLP for Question Formulation: SMEs can effectively use controlled languages to pose AP questions to these systems.
  - Do these systems address key failure and cost issues encountered in the Pilot?
    - Metrics are correctness/coverage (AP scores/subscores) and cost ($/textbook page)

SRI International

THE UNIVERSITY OF TEXAS AT AUSTIN

BOEING

ontoprise
know how to use Know-how

AIFB

Carnegie Mellon

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

VULCAN

Project Halo Intermediate Evaluation: Sept 2006

- Science grad student KBs
- Extensive natural lang
- ~$100 per syllabus page

Domain	Number of questions	Percentage correct			
		SME1	SME2	**Avg**	**KE**
Biol	146	52%	24%	**38%**	**51%**
Chem	86	42%	33%	**37.5%**	**40%**
Phy	131	16%	22%	**19%**	**21%**

VS.

Halo Pilot System	Percent correct
Cycorp	37%
SRI	44%
Ontoprise	47%

- Professional KE KBs
- No natural language
- ~$10K per syllabus page

Knowledge Formulation

- Time for KF
  - Concept: ~20 mins for all SMEs
  - Equation: ~70 s (Chem) to ~120 sec (Physics)
  - Table: ~10 mins (Chem)
  - Reaction: ~3.5 mins (Chem)
  - Constraint: 14s Bio; 88s (Chem)
- SME need for help
  - 68 requests over 480 person hours (33%/55%/12%) = 1/day

Question Formulation

- Avg time for SME to formulate a question
  - 2.5 min (Bio)
  - 4 min (Chem)
  - 6 min (Physics)
  - Avg 6 reformulation attempts
- Usability
  - SMEs requested no significant help
  - Pipelined errors dominated failure analysis

System Responsiveness

- Biology: 90% answer < 10 sec
- Chem: 60% answer < 10 sec
- Physics: 45% answer < 10 sec

	Interpretation (Median/Max)	Answer (Median/Max)
Bio	3s / 601s	1s / 569s
Chem	7s / 493s	7s / 485s
Phy	34s / 429s	14s / 252s

VULCAN

Halo Question-Answering Goals for December 2008

- Demonstrate a 75% score for correctness and explanation on the intermediate evaluation questions, using SME authored KBs
  - Current scores range from 16% to 52%
- Median number of SME question reformulation attempts will be 5 or less (end-to-end)
  - Current numbers are 5 (Chem); 7 (Physics); and 1 (Bio, constrained by limited possible question types)
- Performance
  - Complete 75% of the knowledge formulation operations in 5 sec or less
  - For 75% of the final evaluation questions, the mean response time for interpreting a question and answering a question will be less than 10 sec.
  - For 90% of the questions, the mean system response time for answering the question will be less than 1 minute

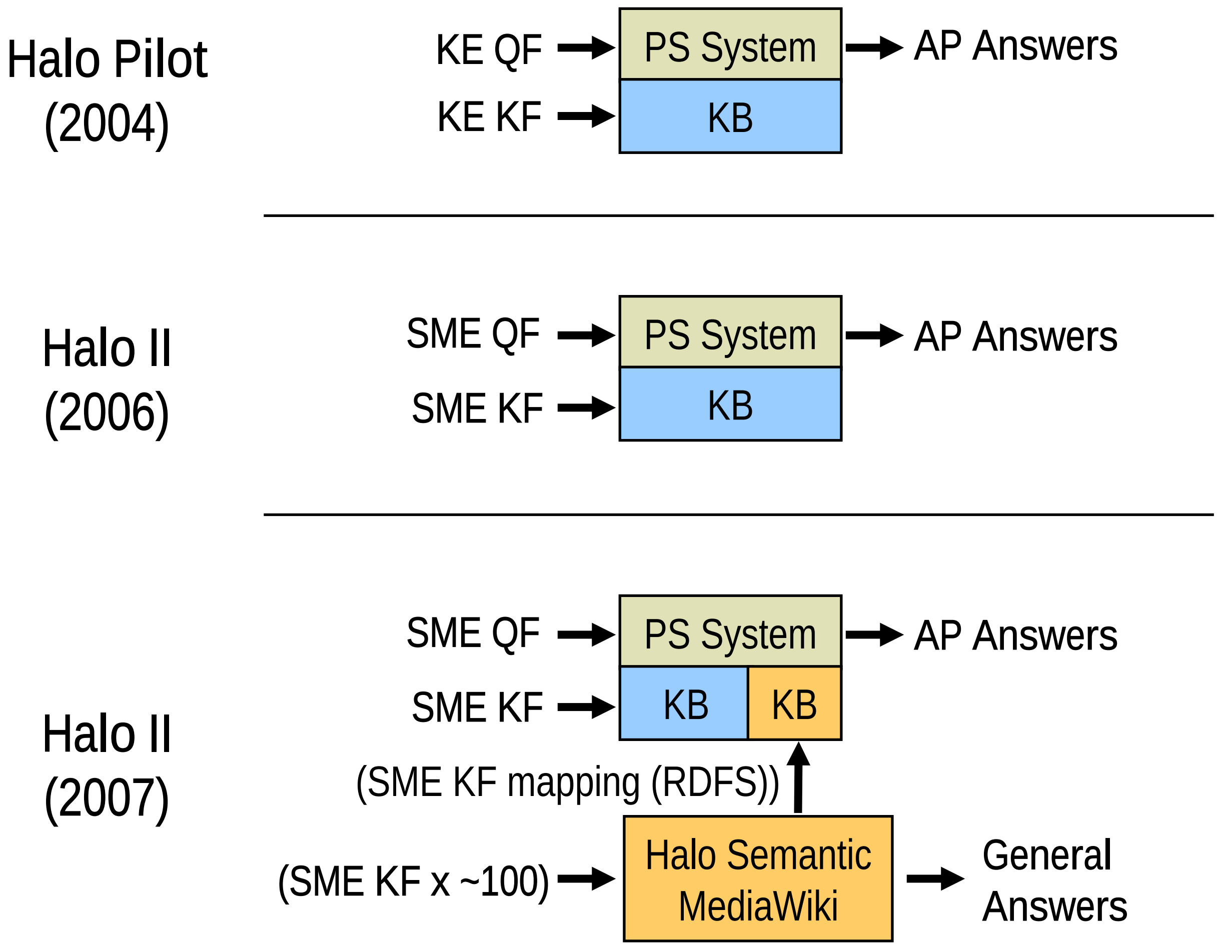
The Halo Project Today

PROJECT HALO

- Low-cost SME Knowledge Entry and Question Answering (Aura)
- Scaling up the KB (Offshore Knowledge Entry)
- Scaling up Participation (Halo Extensions to SMW)
- SME formulation and use of defaults and rule knowledge (Halo Advanced Research)

VULCAN

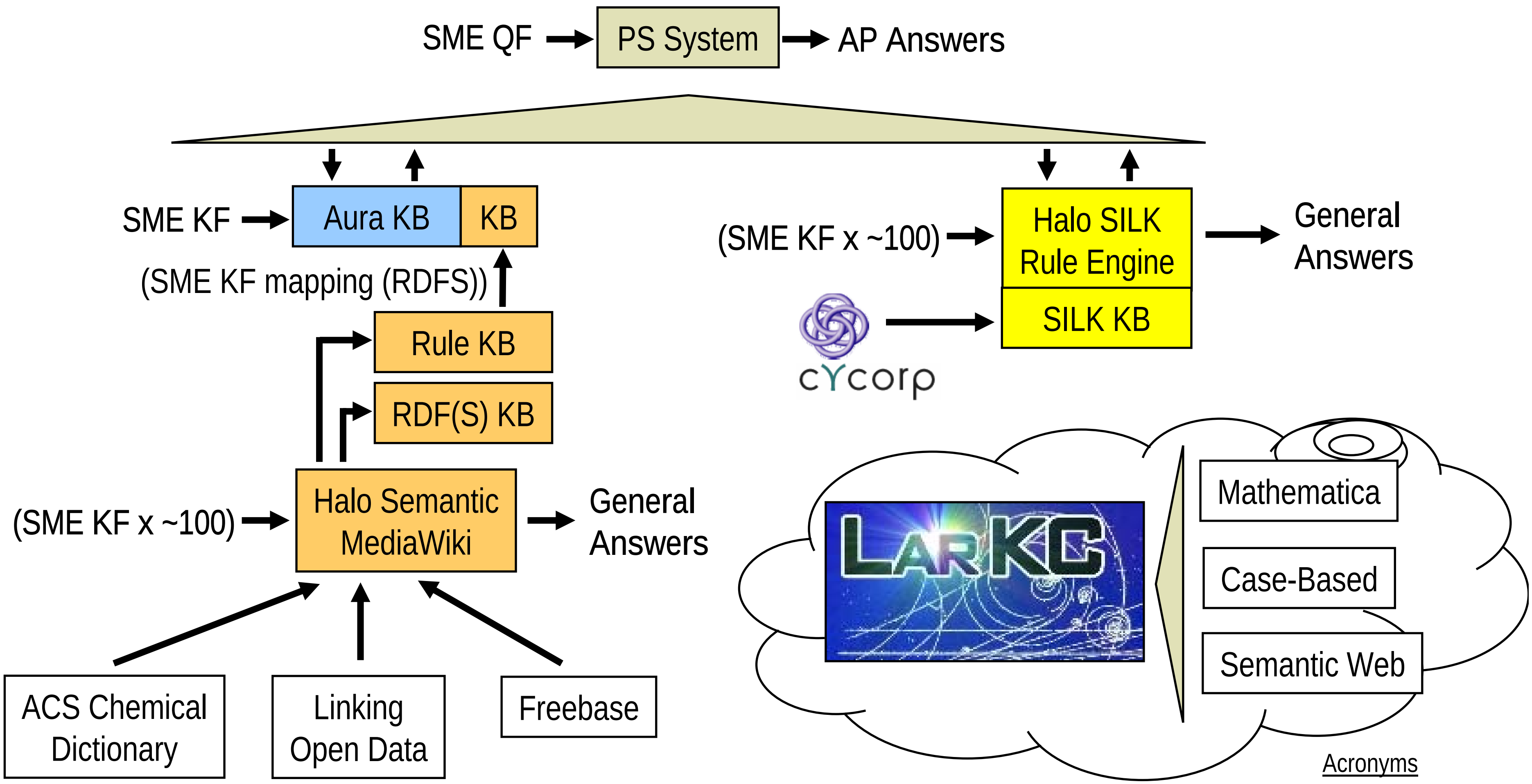
Evolving Halo Architectures



<u>Acronyms</u>

KB = Knowledge Base
KE = Knowledge Engineer
SME = Subject Matter Expert
KF = Knowledge Formulation
QF = Question Formulation
AP = Advanced Placement
PS = Problem Solving

Halo III Architecture (2009-10)

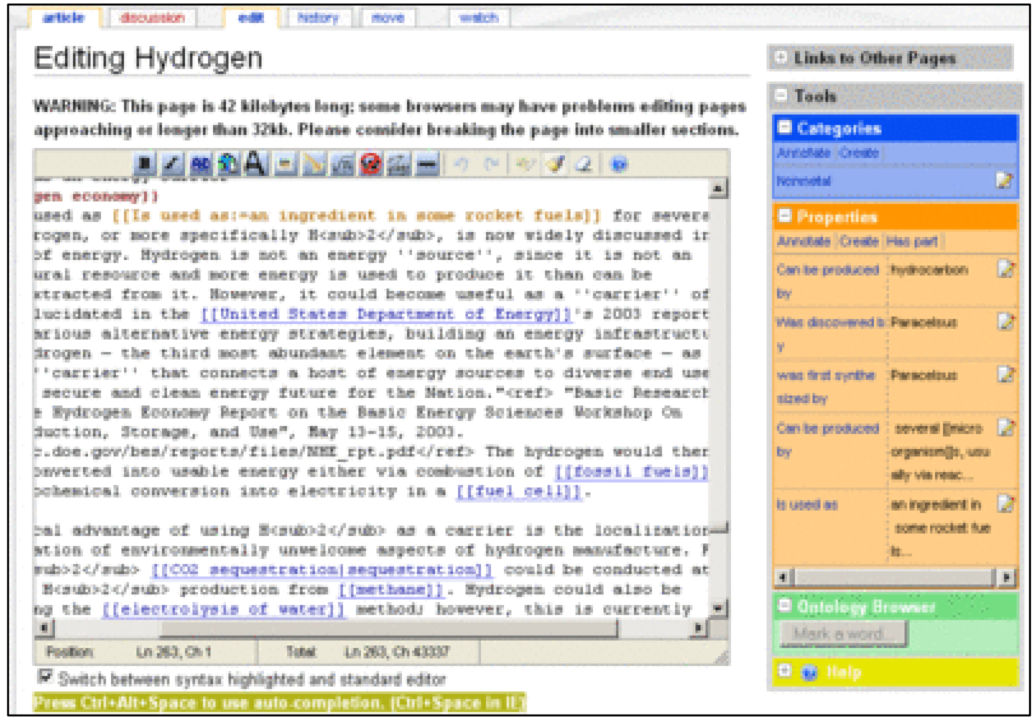


- Decoupled the problem-solving system from Aura
- Overlapping KRs and meta-level problem solving architectures
- Adds SILK for explicit rule formulation and reasoning

Acronyms
KB = Knowledge Base
KE = Knowledge Engineer
SME = Subject Matter Expert
KF = Knowledge Formulation
QF = Question Formulation
AP = Advanced Placement
PS = Problem Solving

Halo’s Extensions of Semantic MediaWiki

- Major new Semantic MediaWiki capabilities
  - Syntax-highlighting editor
  - Autocompletion
  - Ontology browser
  - Automatic table generation via ASK (like RDQL or MQL)
  - Context-sensitive help
  - Export and integration
- Rules prototype is starting
  - Can we use wiki consensus mechanisms for rule consensus?
- Community Support
  - All work is GPL open source
  - http://www.semanticweb.org/wiki/Halo_Extension
  - Ontoprise provides commercial-grade support



- Hypotheses and Metrics
  - Better: the wiki-entered knowledge will be of higher quality
  - Faster: It is quicker to map than to enter
  - Cheaper: Reusable wiki knowledge will meet 30% of Halo’s knowledge requirements

VULCAN

Halo and Rules (Halo Advanced Research)

- SILK: Suite of core knowledge representation and reasoning (KR) modules
  - Provide defaults, hypotheticals, actions, and processes capabilities
    - First Focus: Combine defaults with as much as possible of other established features for monotonic (DB, classical, ontology). Default flavor pervades the KR
    - Key ideas: Courteous extension of Logic Programs, distributed, event-driven
    - Second Focus: Hypotheticals/Actions/Processes. Key ideas: advanced defaults and rules
  - Employ distributed algorithms and platform for high scalability
    - Focus: Incremental update/merge, with distributed dynamic import
    - Key ideas: dependency analysis, precomputation
  - Progressively/iteratively extend with new expressive features and algorithms
  - Early iterates, e.g., initial defaults, have substantial value for science and business/govt.
  - Interoperate via KR and SOA standards with other systems/sources, including web sources
- Knowledge acquisition (KA) and UI modules, building on SILK KR
  - Provide assert, query, answer, browse, edit, test, explain, analyze, debug capabilities
- Integration of the above
  - Into Aura, to significantly boost AP performance
  - Into Semantic MediaWiki (SMW) or other wiki/Web2.0 environment, for knowledge acquisition
  - As a stand-alone KR technology

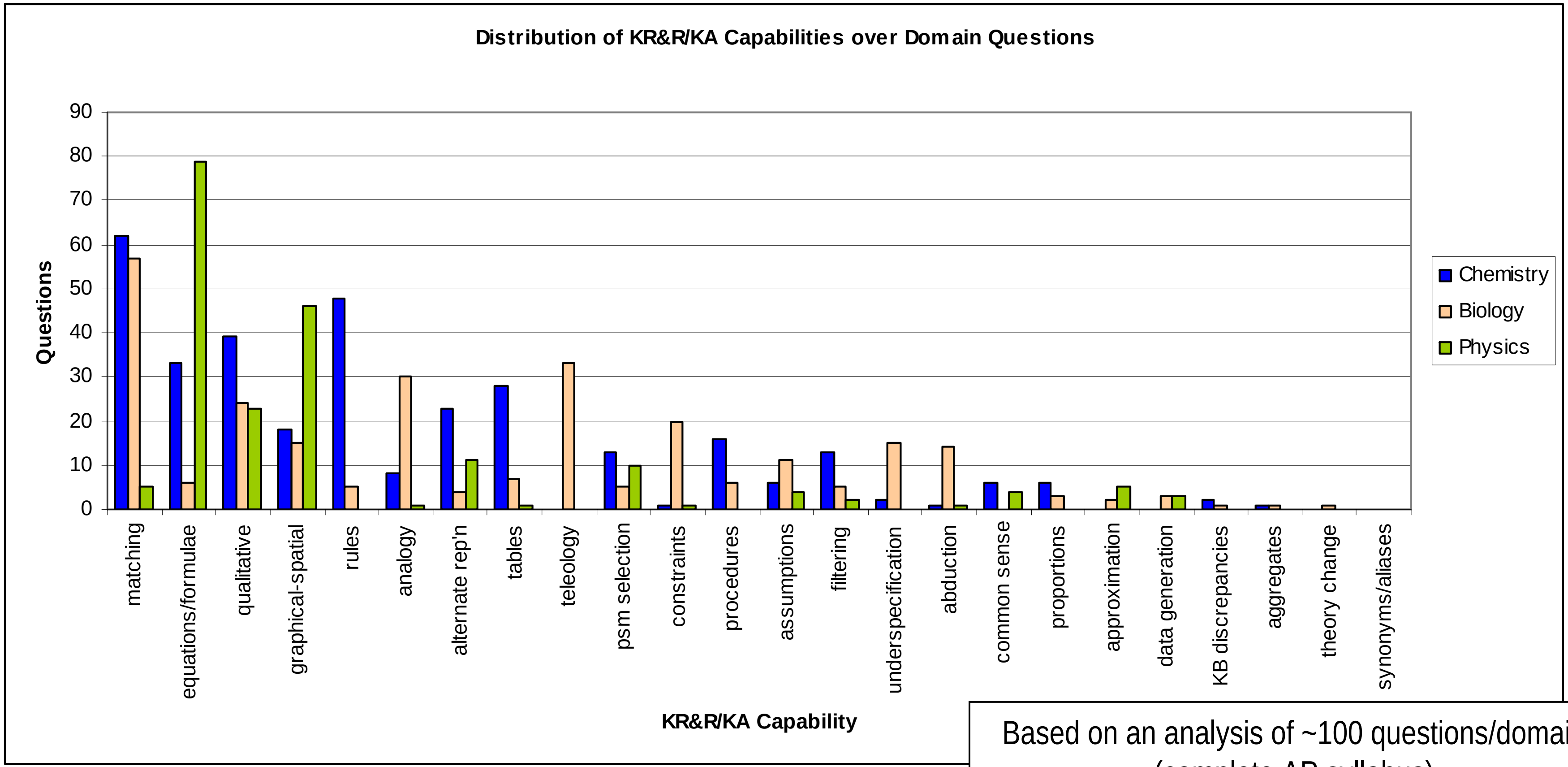
VULCAN

Talk Outline

- Why I'm So Excited about LarKC
- Systems AI – Vulcan's Halo Program
  - Deep Reasoning over the AP problem
  - Systems AI: Unified Knowledge Acquisition, Representation/Reasoning, and Question Answering
- Reasoning in a System – Lessons From Halo for Dynamic Reasoning on the Semantic Web

VULCAN

Lesson 1: Classical Reasoning is Overrated



Based on an analysis of ~100 questions/domain (complete AP syllabus)

- Distribution of reasoning types is highly uneven over domains
- Much reasoning is quite domain-specific (c.f. rules for geographic closeness for school attendance vs. geographic closeness for a mailbox)
- General reasoning is not the dominant tactic for most questions

Lesson 2: Fully-Automated Reasoning is Overrated

- Fully-automatic reasoning is a relic of the algorithmic orientation of computer science
  - Users introduced all kinds of difficulties
- Making users an essential part of the Halo pipeline made Halo's problems easier, not harder
  - Tighter specification of the reasoning problem
    - Mapping of problem onto the knowledge elements
    - Range/precision/type of possible answers
  - Selection of problem-solving framework and ontology
    - No single correct ontology
  - "Black box" question answering does not reflect the real experience of people's more sophisticated use of the web
    - Embrace query refinement
    - Embrace partial answers and "answering questions with questions"
    - Embrace user experiments and hypotheticals with the LarKC KB
- User trust requires system transparency
  - Usability of applications will be tied to explanation/drill-down capability
  - LarKC will require traceability, truth maintenance, provenance, security

VULCAN

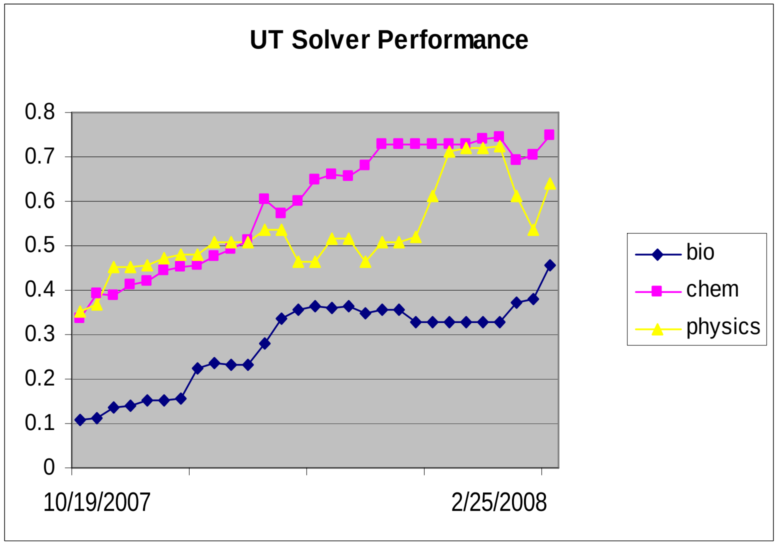
Lesson 3: Context and Nonmonotonicity are Everywhere

- Even in AP-level science, assertions are relative to the pedagogical context
  - Question context must be specified
- Knowledge has a lifecycle
  - Mistakes are part of the system
  - (Again) The LarKC platform needs an explanation module, with citations back for provenance of all the reasoning outcomes
- Many applications will require personal and context-dependent rule-like components for use by LarKC
  - Customization / personalization of LarKC results requires differential treatment of classes of data
    - A "personal LarKC" part of the platform that allows users/groups to contextualize the LarKC results (and guide the LarKC considerations)
    - E.g., sampling from data sources of a particular quality or point of view
  - The variety of needed customized reasoning requires (at least) defaults, exceptions, and processes
- (Some) commonsense is needed
  - Users are needed to specify this and guide the system

VULCAN

Lesson 4: What is Measured is What Matters

- LarKC needs quantifiable benchmarks!!
  - These are critical to sell LarKC past the Early Adopters Group



- Good benchmarks are good for a program
  - Halo's focus on SME-driven AP scores is our Occam's Razor
  - Ex: "LarKC will find 80% of the most network-connected individuals in 250M FOAF triples using a 10% sample, and be 6x faster than the same algorithms applied to a 60% sample"

- The design of benchmarks, development of a robust testing facility, and repeated running of the benchmarks is contentious and expensive, but worth it
  - Exposes the gap between great theory and great software engineering
  - Shows the parts of a system architecture that are actually critical
  - Provides a real-time control signal to the program development/prioritization activities
  - Allows potential users to "try before they buy"

VULCAN

Lesson 5: Customers have Needs

- Quality software engineering for a hybrid parallel architecture
  - Bug tracking, performance profiling, continuous regression testing, use of industry standard dev tools (Eclipse?) and languages (no LISP or Prolog)
  - Deadlock, livelock, race conditions, and other concurrency mistakes will happen
  - The only way LarKC can achieve reliable concurrent code is to start with extremely good and rigidly enforced architectural and development principles, and follow up with lots of testing
- A user community that is easy for industrial users to join and contribute to
  - Frequent releases of the platform and current tools, with demos
  - Clear LarKC metrics and a published road map
  - Simple GUI apps against LOD (or other) datasets that show the technology
  - A published road map and decent documentation
  - LarKC "developer days" in parallel with the main program meeting tracks
  - Supportive engagement in terms of web sites, mailing lists, and publications
- A company-friendly IP policy
  - Commercially reasonable licenses (no GPL!) for the framework and sample reasoners
  - The ability to use proprietary technology in LarKC plug-ins

Lesson 6: The world will change, but you can guide it

- LarKC is looking 4 years out (to 2012)
  - More services on the web (data supply, data reduction, data presentation)
  - Databases will continue their trend to better performance
  - More distributed computation via mobile devices
  - Sensors will be far more prevalent
    - Evanescent facts
    - Sensor error and context-dependence
  - Loads of intentionally false semantic material
    - Lesson of the HTML <META> tag
- What are the competitive/complimentary technologies?
  - Hardware scaling with multicore processors
  - More search-ready data and format convergence
  - NLP-based dialog systems and query refinement
  - Linking Open Data movement
- Keep your competitors close
  - Continuous, frequent releases of code will help ensure LarKC is part of the future

VULCAN

Weight and Balance Considerations for LarKC Liftoff

- R&D Goals
  - Algorithmic correctness and academic purity
  - Publications and community respect
- Transition/Revolution Goals
  - Robustness
    - Code dependability in a parallel environment can be unbelievably hard
    - The UltraLog cluster experience
  - Scalability
    - "Scalability by muscle power" is not a bad thing; centralized caching architectures are standard in Web search
    - Parallel access using industry standard server-farm tools (e.g., SQUID for server web caching, Apache load balancing and replication)
  - Transparency
  - Solving a particular problem better / faster / cheaper than anyone else
- Build LarKC as an enabler for problems that require broad, shallow, non-precise semantic answers from the web
  - Data fusion and question-answering companies like ZoomInfo
  - Information vendors like Reuters or Lexis/Nexis or search engines
  - Advertising companies who need methods for more precise placement

VULCAN

Thank You