AI in Software Engineering Research

David Garlan & Bradley Schmerl

Today’s Talk

- Building AI into modern system architectures
  - Some examples of the use of AI in emerging systems research,
  - In many cases this uses non-ML AI
- Three examples
  - Self-adaptive and self-healing systems: Rainbow
  - Task-oriented computing: Aura/RADAR
  - Hybrid planning

Modern software systems

- Must be highly resilient in presence of errors and changes in environment

  => Self-adaptive systems (a form of autonomy)

- Must support high levels of task assistance for their users

  ⇒ Flexible architecture for AI agents to help users

- Must be able to respond quickly when needed to solve a problem fast; but accurately when time permits

  ⇒ Supporting both fast and slow autonomy

AI can help

- Planners can reason about future courses of action in the presence of uncertainty
- Multiple AI agents that are coordinated to assist in complex tasks
- Timing-aware systems that can reason about how long it takes to perform AI tasks and choose the right AI mechanism

Project 1: Rainbow
Self-Healing Systems

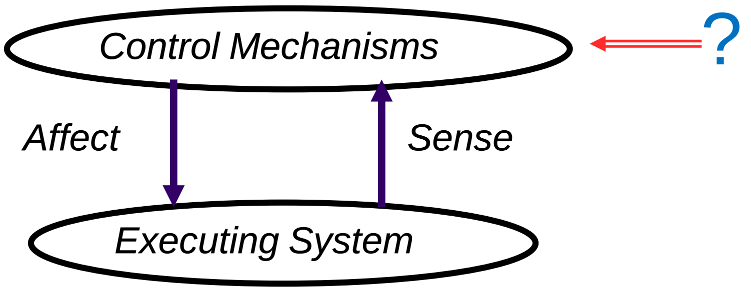
- Increasingly, systems
  - are composed of parts built by many organizations
  - must run continuously
  - operate in environments where resources change frequently
- For such systems, traditional software engineering methods break down
  - Exhaustive verification and testing not possible
  - Manual reconfiguration does not scale
  - Off-line repair and enhancement is not an option

A New Approach

- Goal: systems automatically and optimally adapt to handle
  - faults and attacks
  - variable resources
  - changes in user needs

But how?

Answer: Move from open-loop to closed-loop systems



Example: Google File System

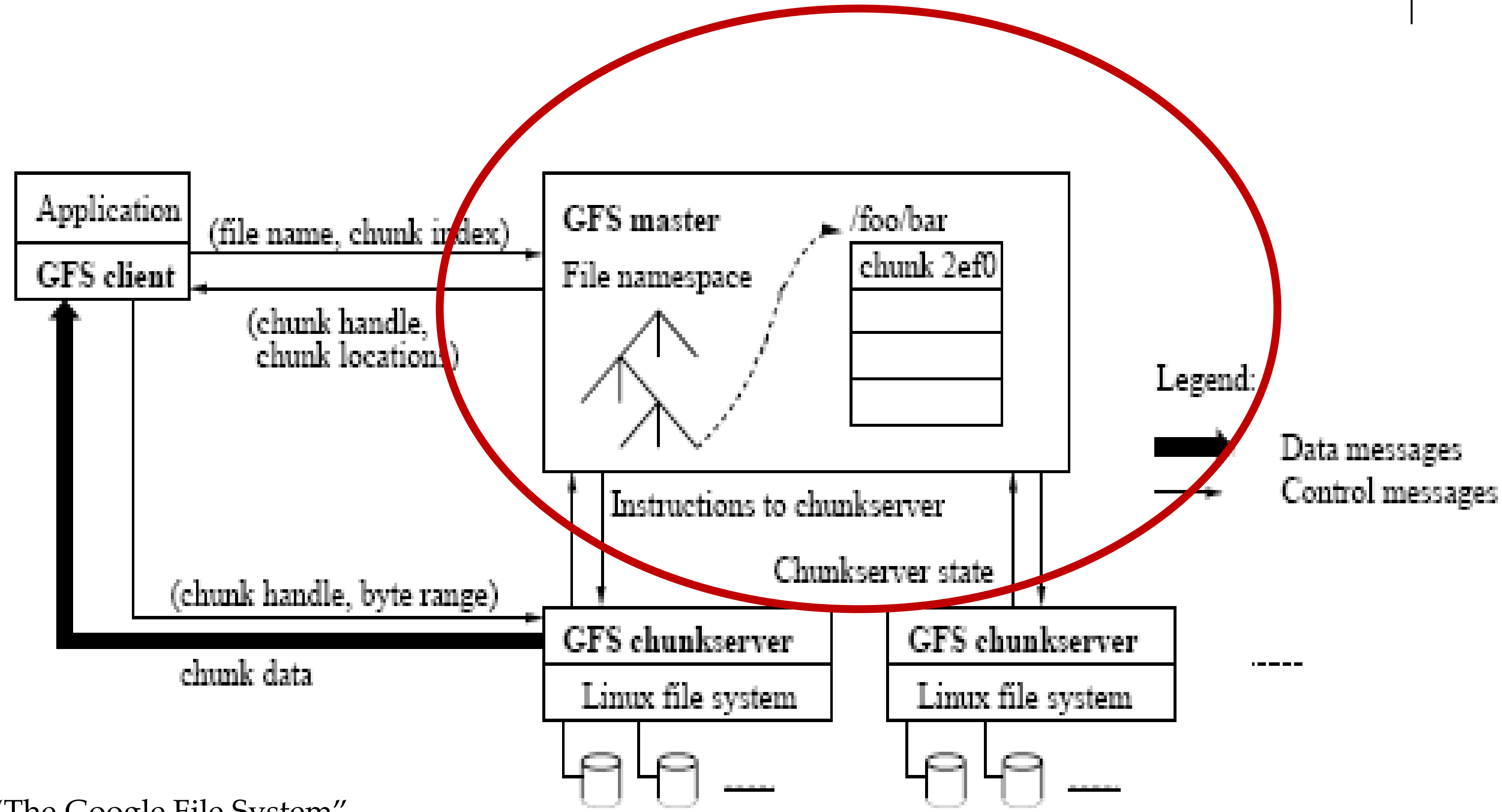


Figure 1: GFS Architecture

Source: "The Google File System" Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. SOSP 2003.

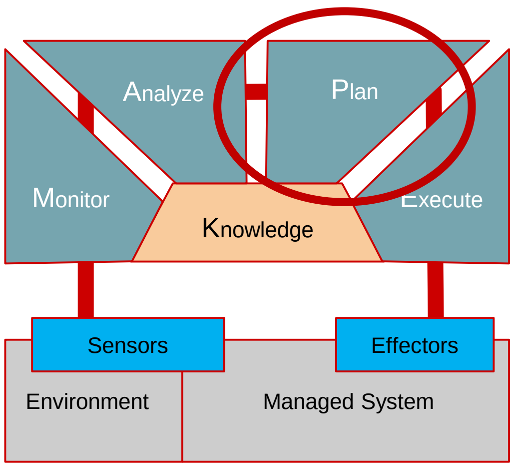
Research Approach

Three ideas:

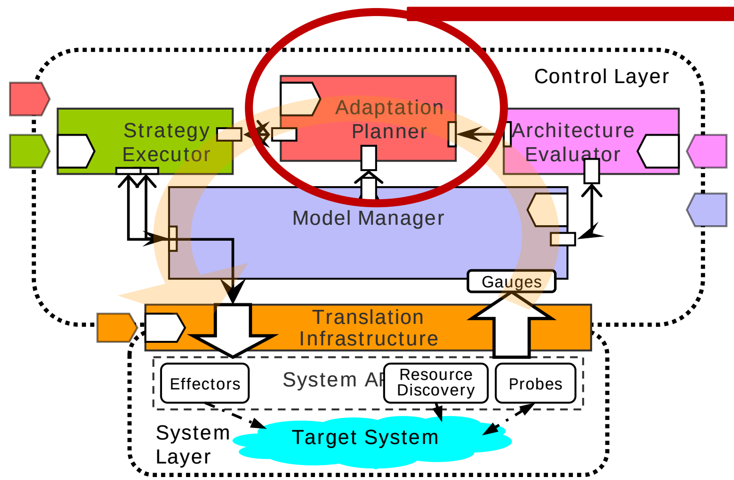
1. Maintain system models at run time as a basis for self-adaptation
   - monitoring
   - problem detection
   - repair – using planning technology
2. Explicitly model businesses
   - tailor problem detection to situation and user needs – dynamic tradeoffs
3. Cope with uncertainty by quantifying it
   - probabilistic modeling and analysis

AI Planning is a key component

- In our research we (and many others) have adopted a control systems view of system autonomy
- But what decides how to adapt?



MAPE-K



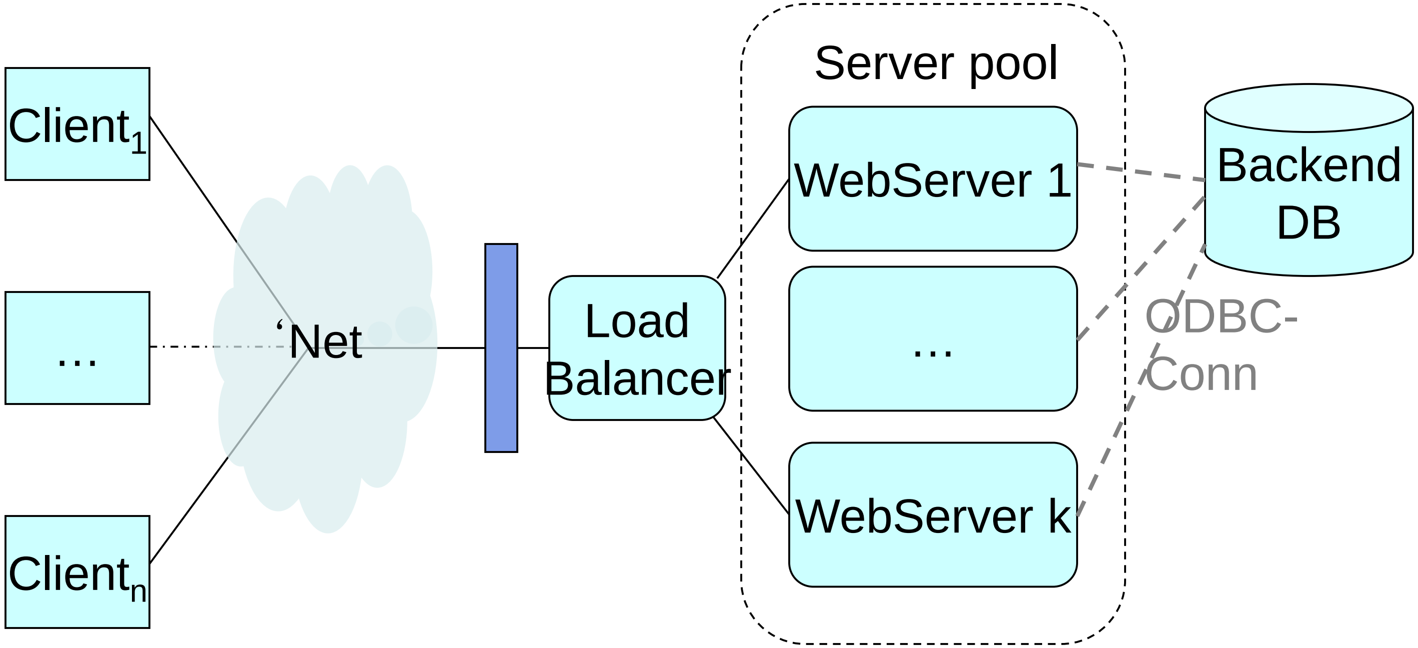
Rainbow Framework

Planning

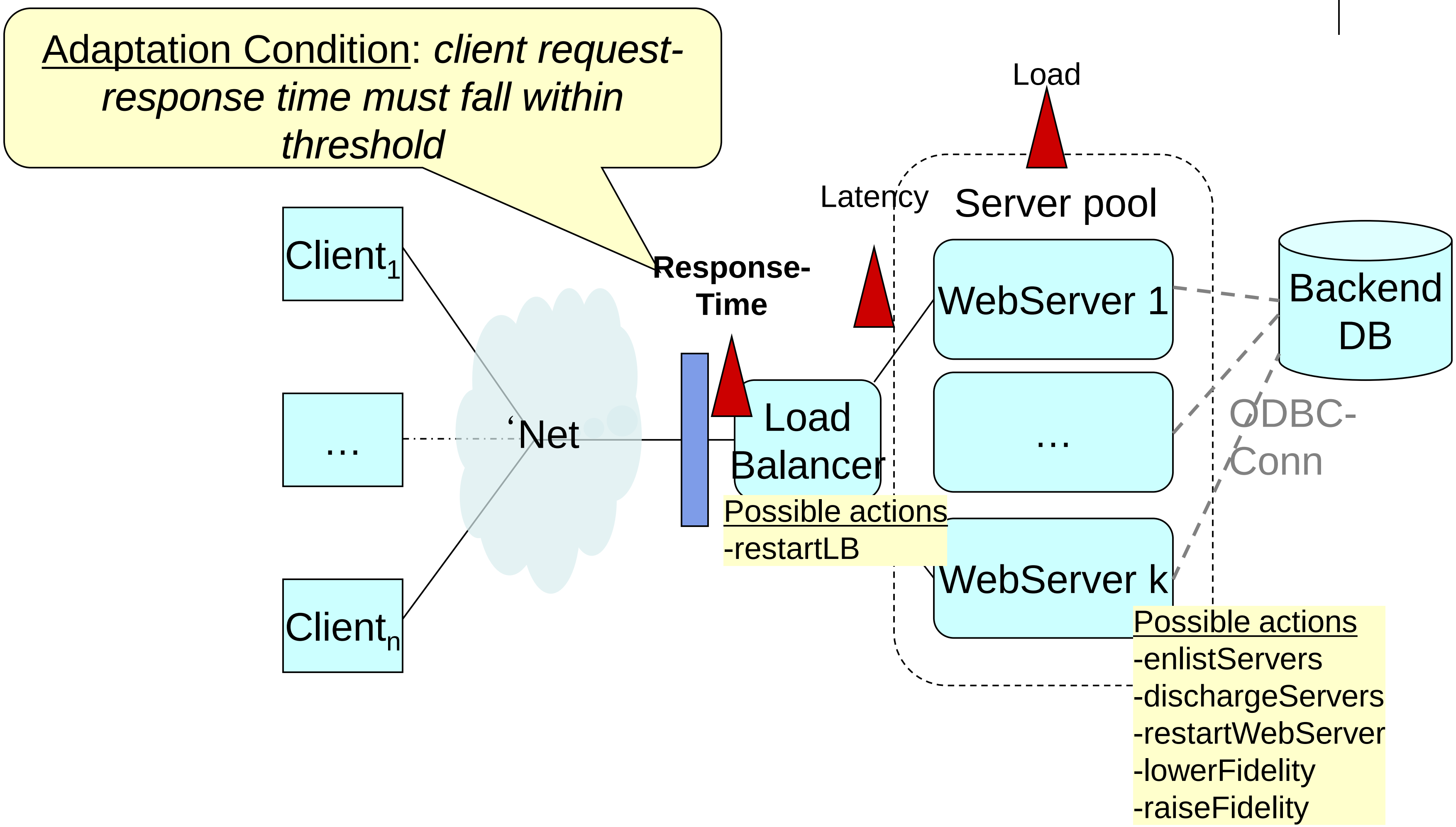
- Uncertainty
- Timing
- Proactivity
- Utility

Probabilistic Models
Stochastic Games
ML-based

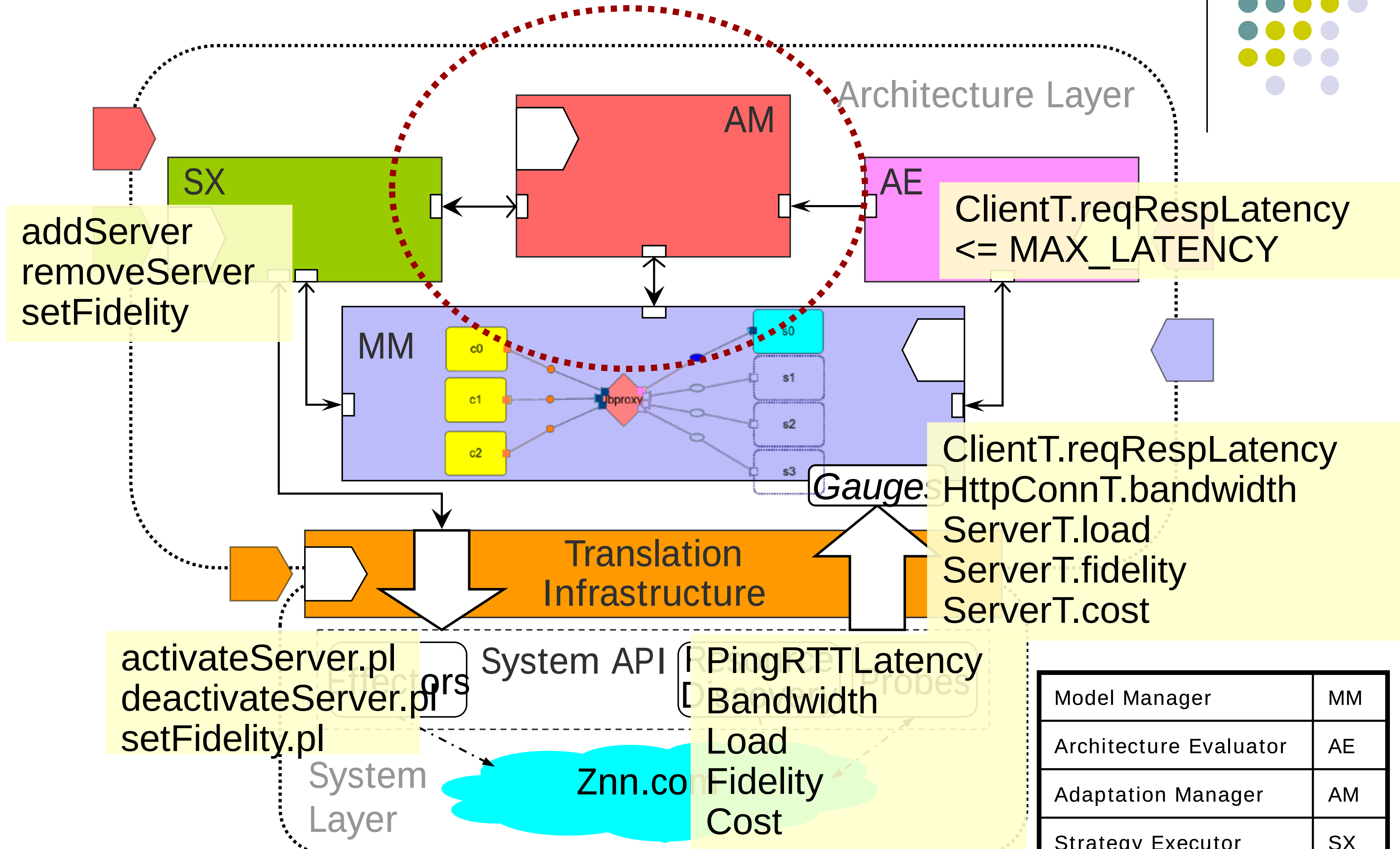
Self-Adaptation Example: *Znn.com*



Self-Adaptation Example: *Znn.com*

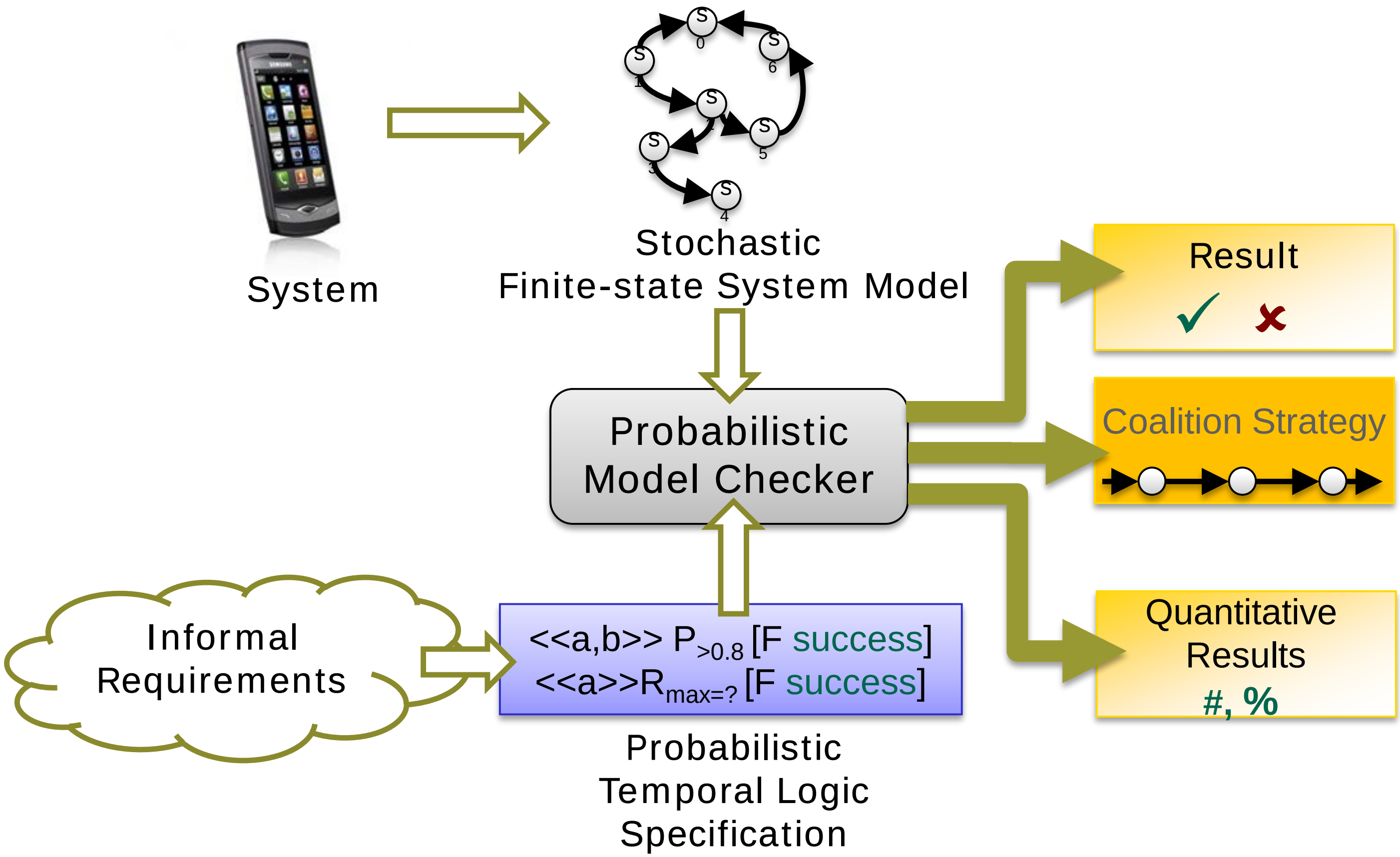


Znn.com: Rainbow Customizations



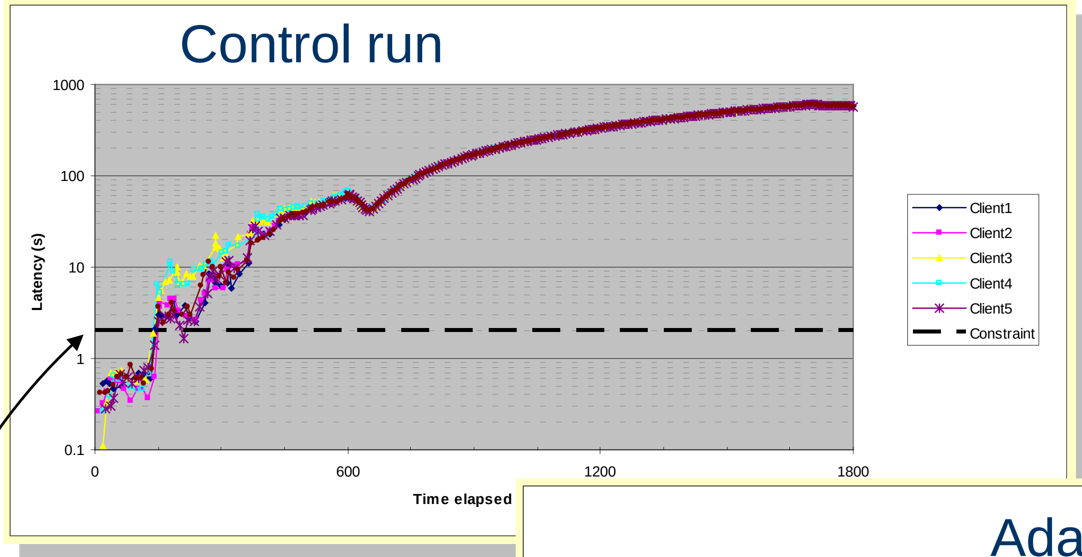
Model Manager	MM
Architecture Evaluator	AE
Adaptation Manager	AM
Strategy Executor	SX

Formal Verification and Strategy Synthesis

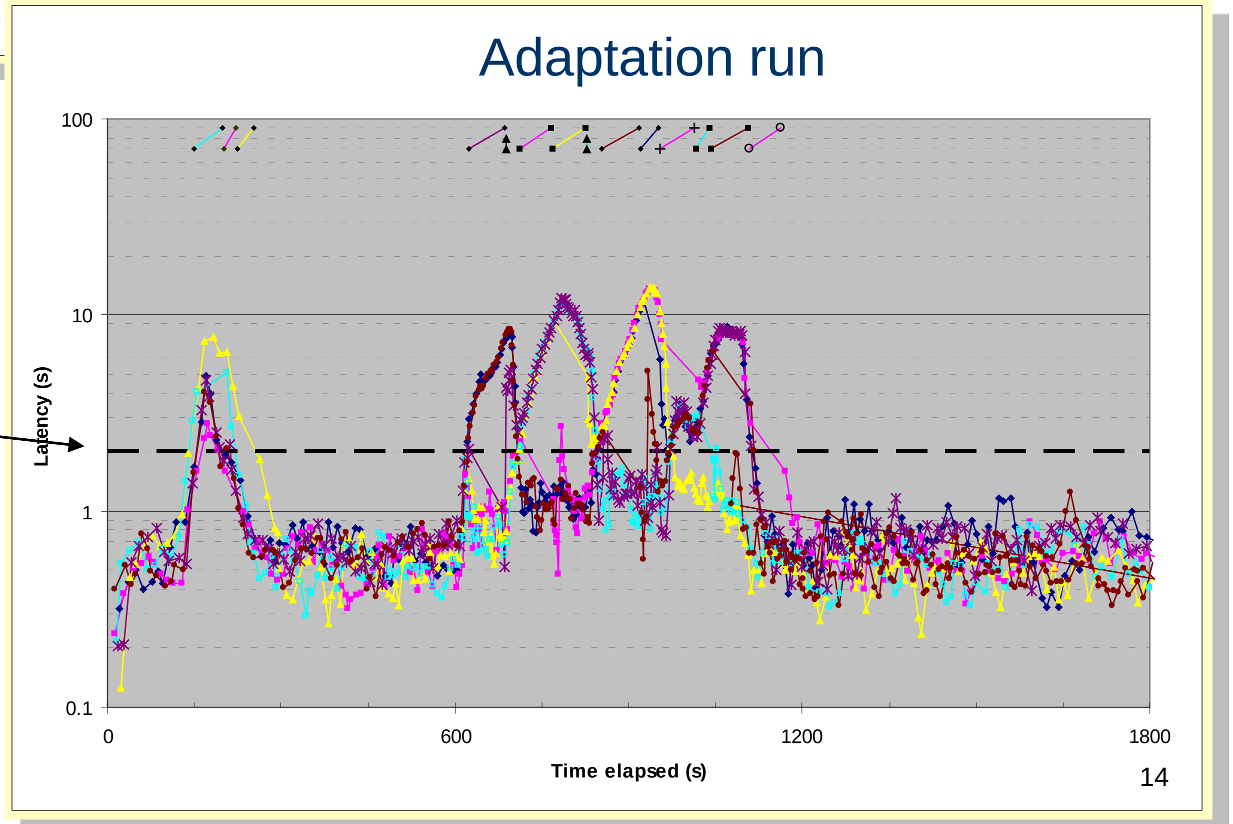


System Adapts

Data shows that our adaptation approach improves overall system performance



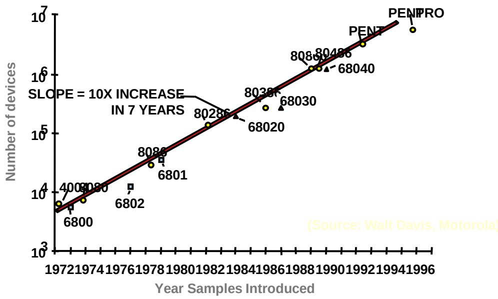
Latency = 2 secs



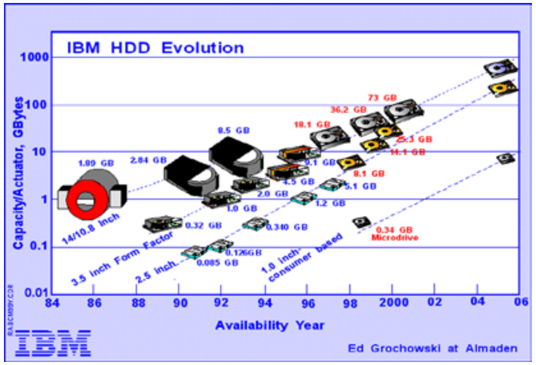
Project 2: RADAR Task-oriented Computing

- How can we get systems to do a better job helping us with our daily tasks?
- The problem: different tasks require different expertise; hence different AI approaches
- How can we come up with a flexible architecture that allows us to integrate all of these?

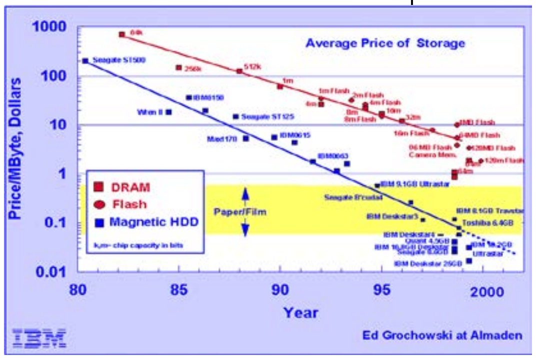
The Challenge for Computing



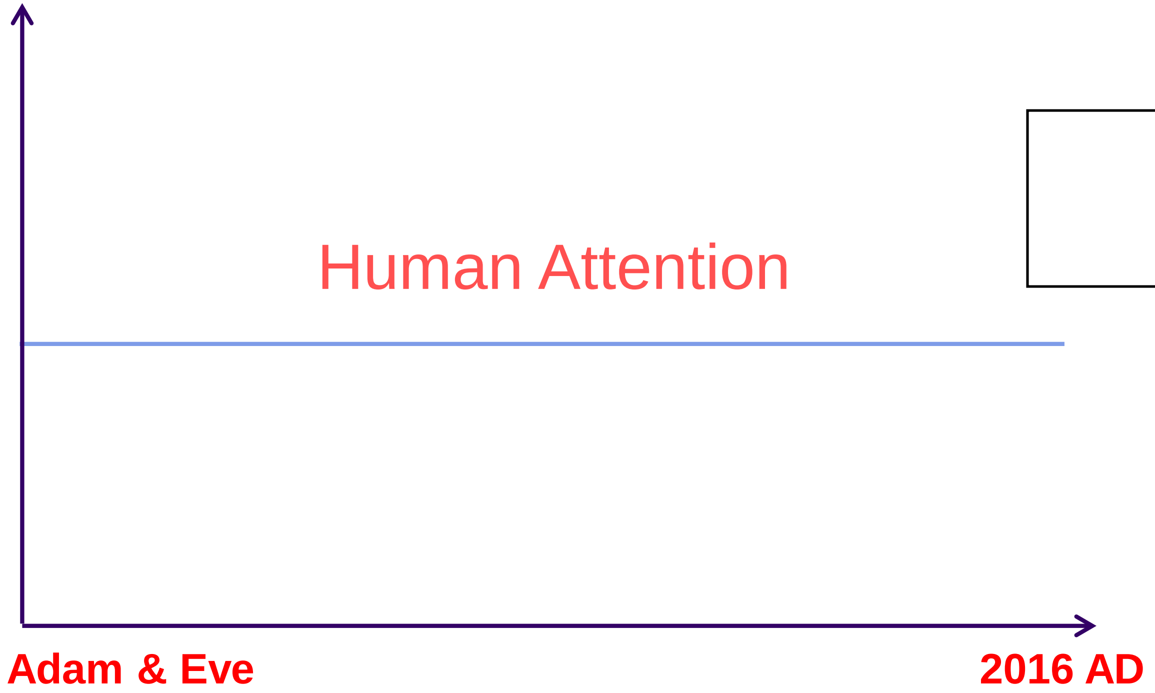
Transistors per Processor



Disk Capacity



Cost per Megabyte



Exception to Moore's Law

Research Addressing This Issue

- Task-Oriented Computing
- Self-managing Systems
- Energy-Aware Adaptation
- Cyber Foraging
- Smart Spaces
- Multi-fidelity Computation
- Predicting Resource Usage
- Intelligent Networking
- Context-aware computing
- User Interface Adaptability

Background Technologies

Speech Recognition

Language Translation

Multimodal User Interfaces

Software Composition

Proxies/Agents

Machine Learning

Rapid Failover

Security & Privacy

Robustness, Reliability

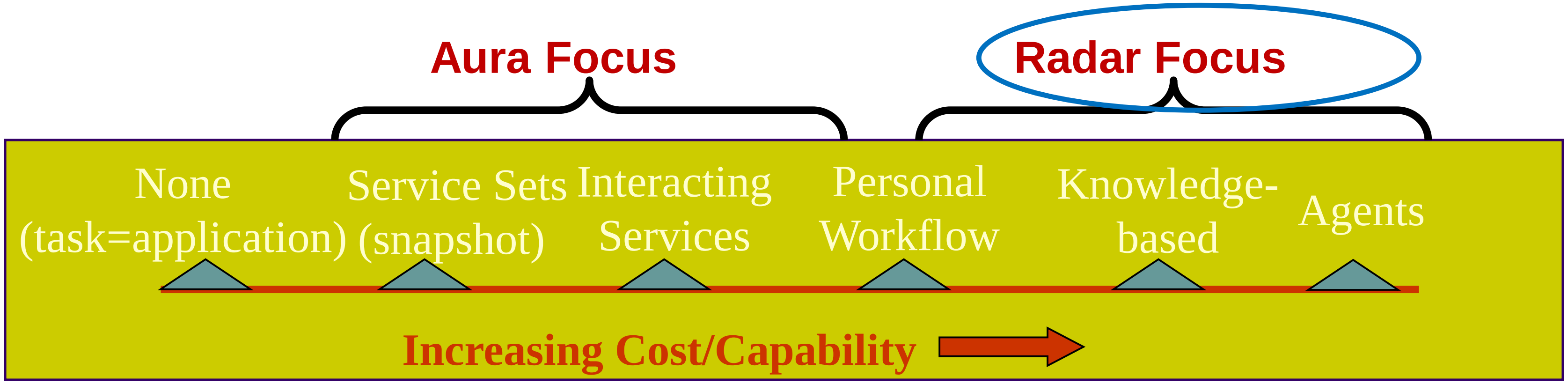
• • • • • • • • •

Tasks

- Tasks encode user goals
  - In office: preparing a lecture, writing a report
  - In home: cooking dinner, relaxing in evening
  - In car: planning a trip
- A system that “understands” user tasks can:
  - Support mobility
  - Automatically reconfigure an environment to take advantage of local resources
  - Adapt to faults, resource variations, changing user needs
  - Assist a user in doing useful things

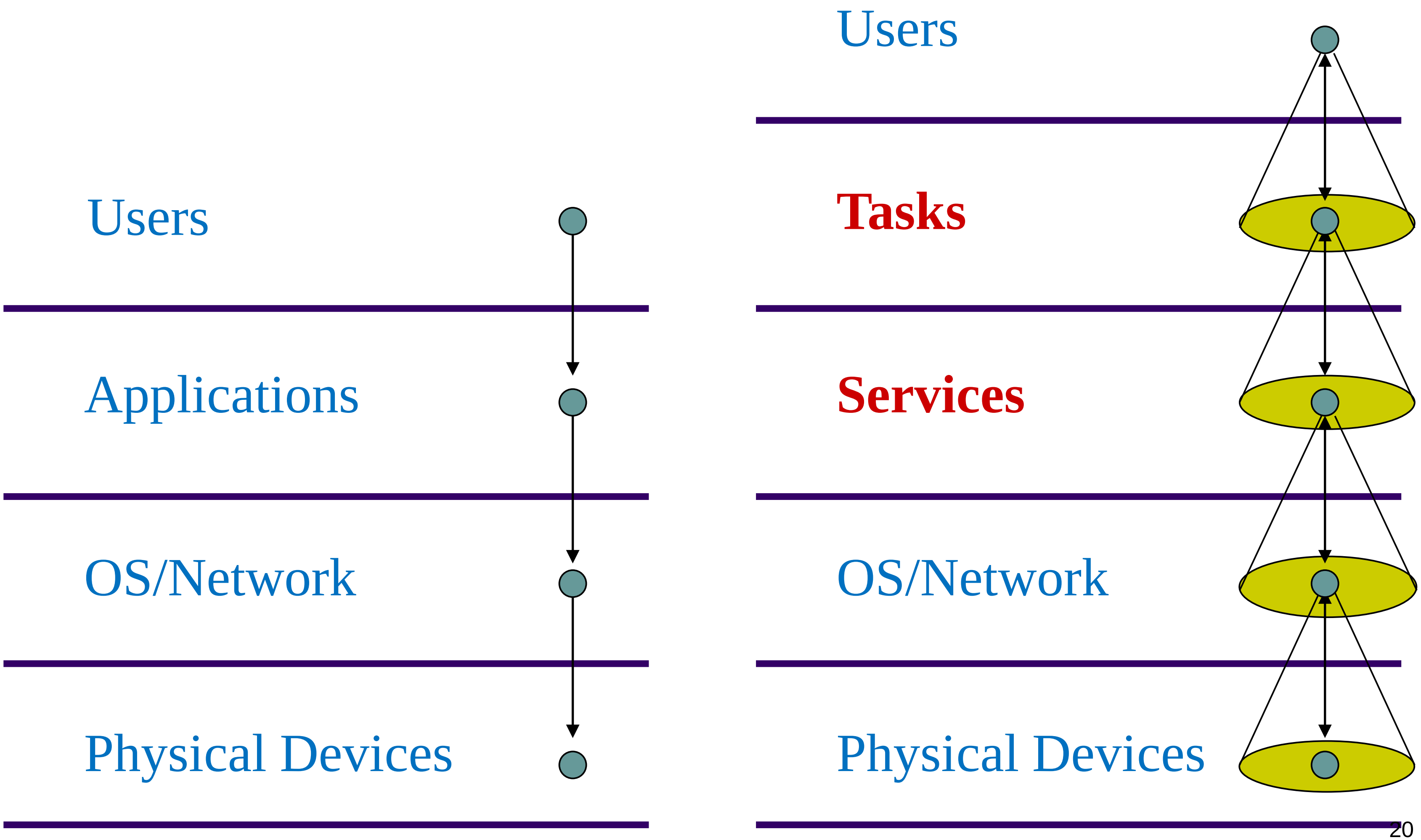
What is a Task?

- Many possible answers
  - A single application or service
  - A collection of coordinated services
  - A workflow
  - A set of goals and constraints



Today

Tomorrow



How to get Tasks into a System

Task Management

What the user needs

- monitor task, context, preferences
- map tasks to services
- manage complex tasks

Environment Management

How to configure environment

- monitor capabilities & resources
- map services & state to suppliers
- optimize to maximize utility

Environment

Support user tasks

- available services & resources
- probes to reflect current QoS

The RADAR Vision

- A **Personal Cognitive Assistant** (PCA)
  - Like a good secretary
  - Helps us with routine tasks, giving us more time for the fun/creative/challenging activities
  - Understands our needs and preferences
  - Adapts to our behavior over time
  - Helps where we desire it, but gets out of the way otherwise
  - Does not require us to change the way we like to do business

Example

- Professor X sends email saying he will be visiting CMU and wants to meet with me and give a research talk.
- My PCA reads the message and realizes that there are several automatable tasks involved
  - Reserves time on my calendar for a meeting with X
  - Locates a lecture room, and books it for X’s talk
  - Updates the departmental web site after requesting a title, abstract and bio from X.
- Along the way it confirms various actions with me to make sure it is on the right track.

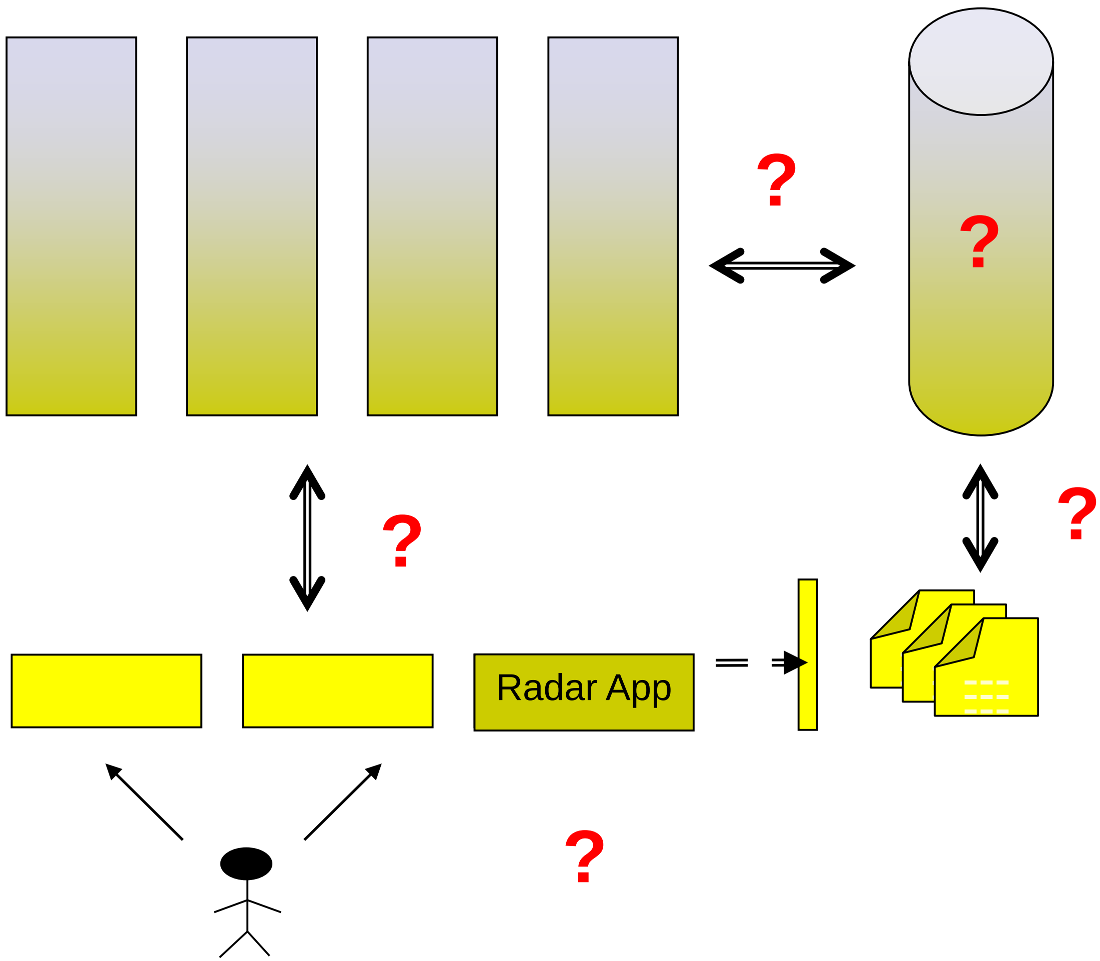
The CMU RADAR Project

- RADAR = “Reflective Agents with Distributive Adaptive Reasoning”
- About 40 researchers (faculty, staff, students)
- DARPA-supported (under PAL Program)
- Central focus on Learning
- Stringent evaluation requirements

The Radar Vision

Task Assistance
<Communications, Time,Space, Web,…>

Radar Information
<Plans, Goals, Preferences, Strategies,…>



Applications
<Outlook, Netscape, Word, Eudora, …>

Storage
<File Systems, Email, Storage, …>

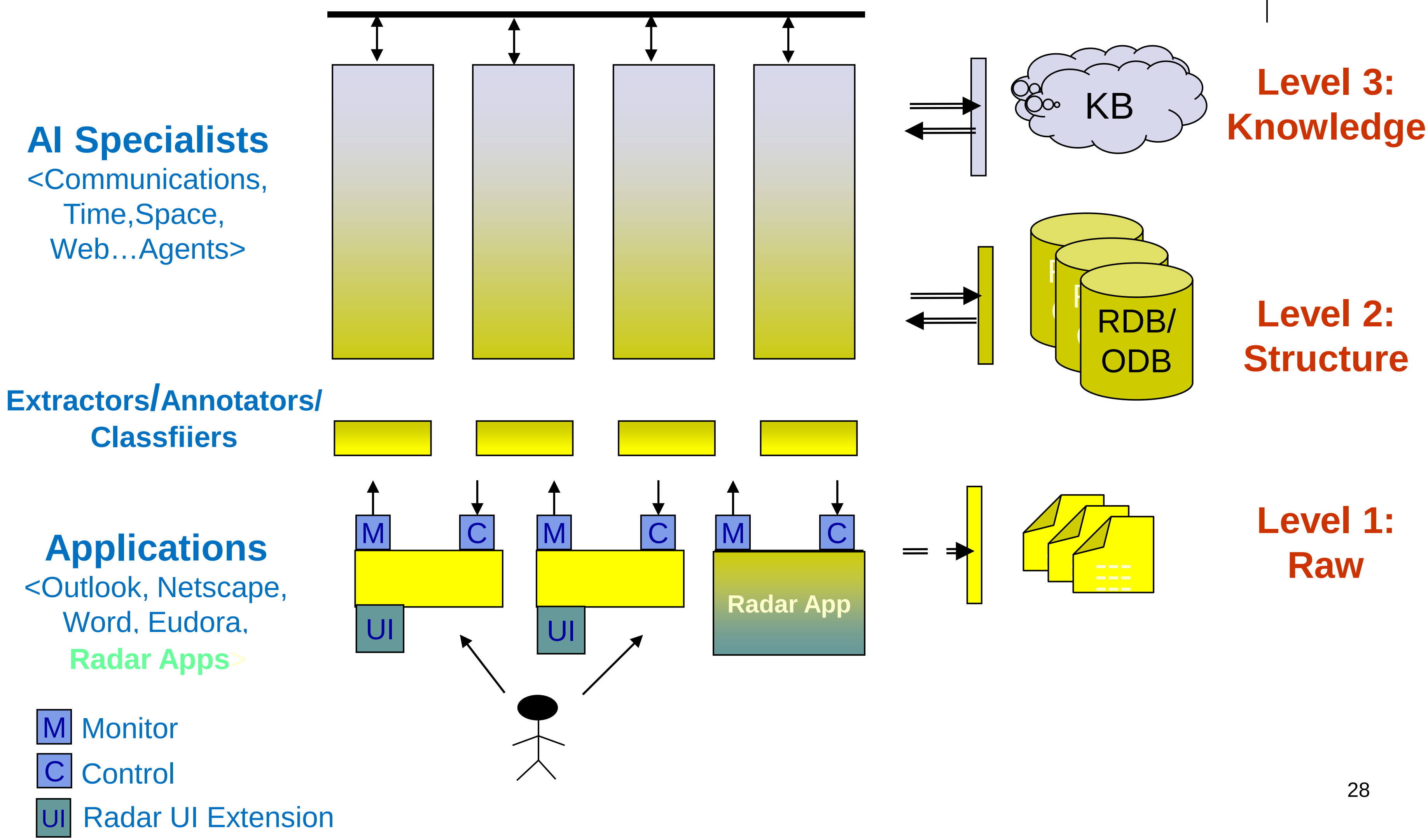
Key Architectural Requirements

- Compatibility
  - Must work with existing applications, information, processes, interfaces, policies
- Extensibility
  - Must be able to incrementally add new capabilities in support of new task domains
- Adaptability
  - Must adapt to individual user's needs and preferences over time
  - Must become more useful and helpful over time

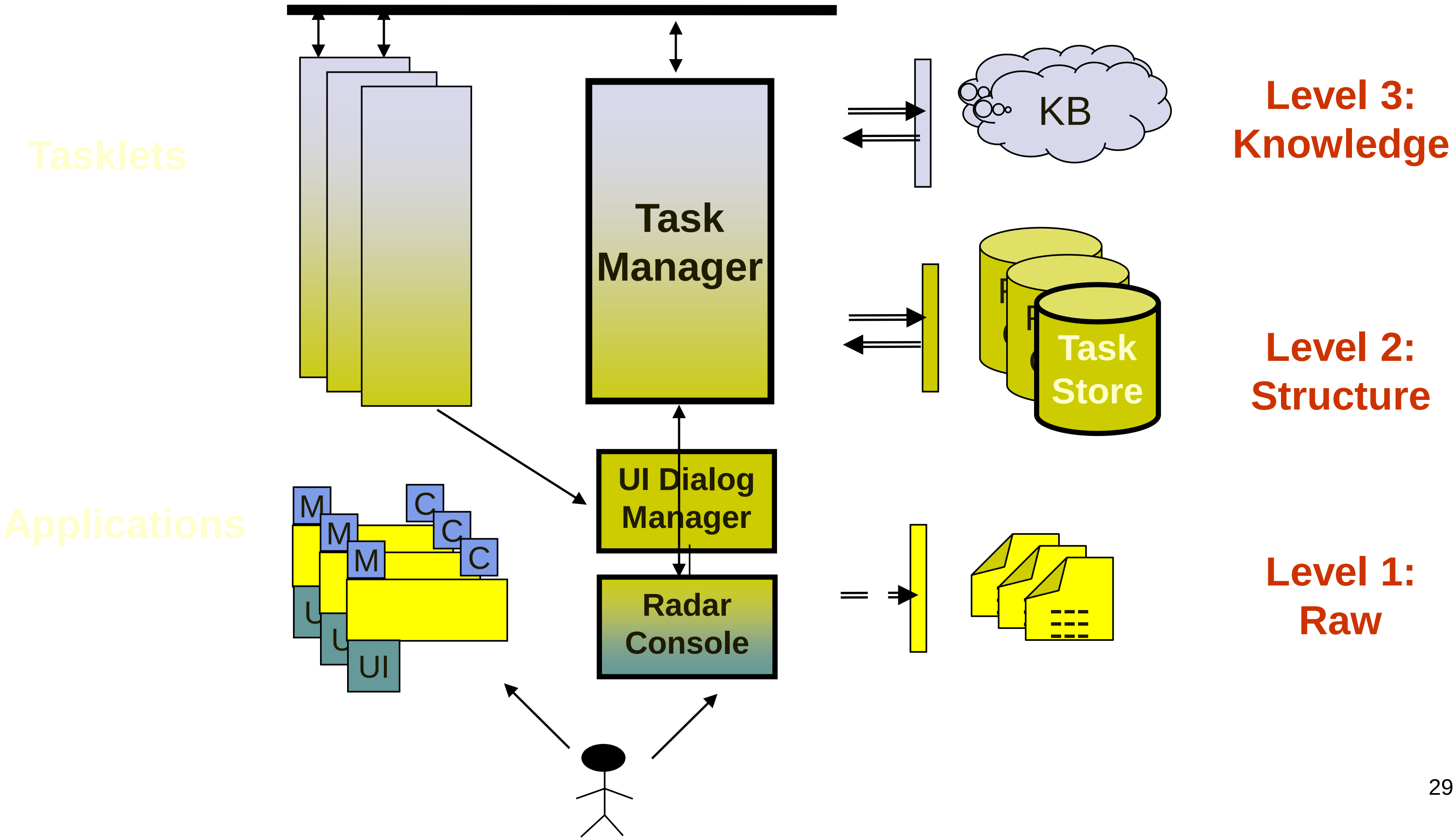
Other Requirements

- Reliability and Availability
  - Comparable to email systems
- Secure and private
  - Access to personal information should be controlled
- Scalability
  - Should support hundreds of users

The Radar Architecture



Task Coordination

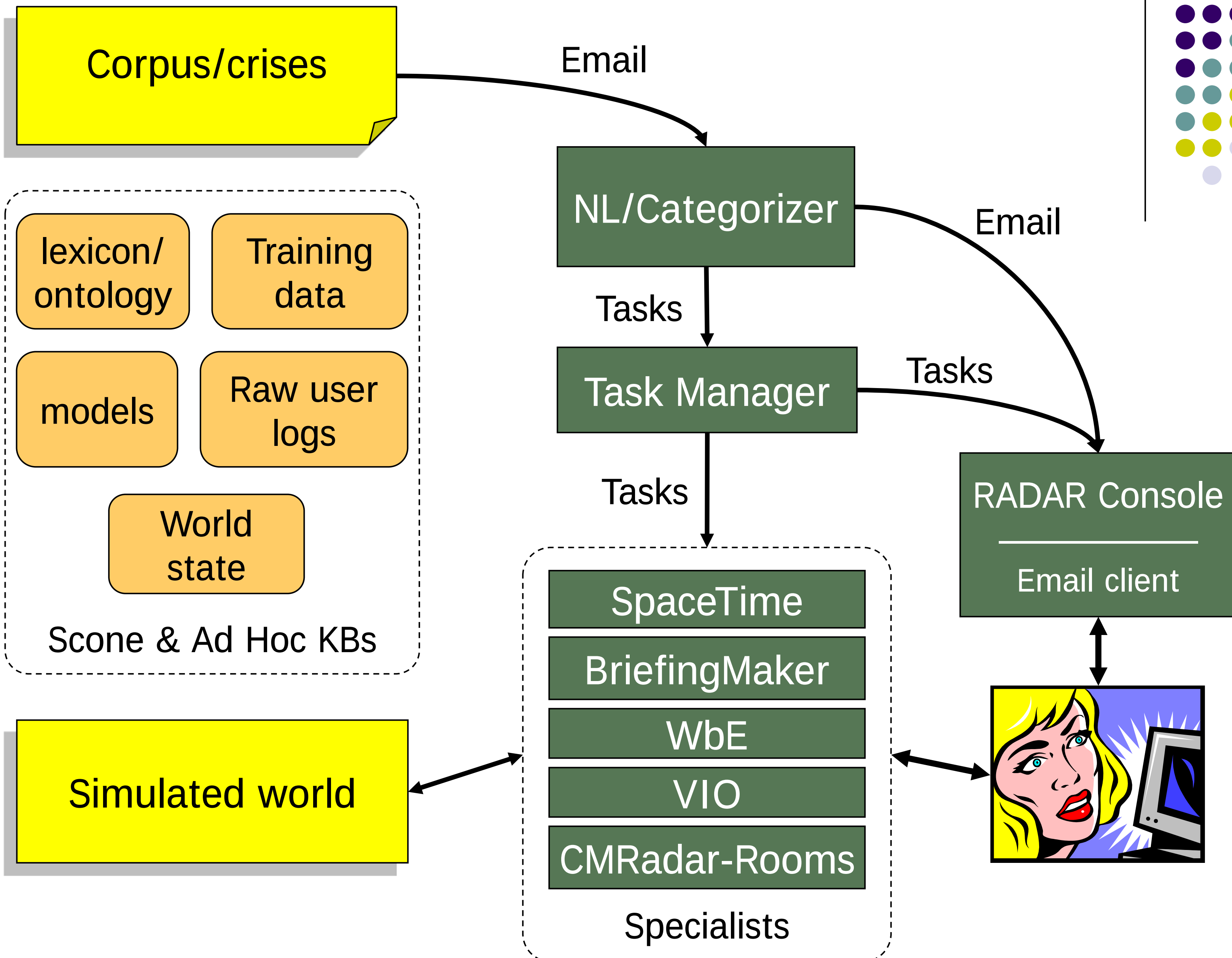


Radar Architecture Concepts

- Key concepts
  - Task Specialists carry out various kinds of task assistance, like calendar management, web management, etc. based on AI (planning,ML,etc.)
  - The Task Manager coordinates specialists, and provides other shared services (task dispatching, dispatching, notification, task history and status)
  - Categorizers and Classifiers use NLP on email messages to allow automated processing
  - A Knowledge Base stores semantically rich information and relationships learned by RADAR

Role of AI

- Intelligence
  - Learning is built in to everything
  - Uses common learning packages
  - Human interface allows RADAR to cooperate with user, adjusting degree of automation over time
- Examples of learning and adaptation
  - Email classification and feature extraction
  - Social network knowledge
  - Task prioritization
  - Task identification and dependencies
  - Calendar preferences


Corpus/crises
Email
NL/Categorizer
Email
lexicon/ ontology
Training data
models
Raw user logs
World state
Scone & Ad Hoc KBs
Tasks
Task Manager
Tasks
Tasks
RADAR Console
Email client
SpaceTime
BriefingMaker
WbE
VIO
CMRadar-Rooms
Specialists
Simulated world

Experimental Evaluation

- Application: existing conference in crisis
  - Working environment, backstory, and original plan
- Major crisis with widespread ramifications
  - E.g., Primary building unusable
- Perturbations
  - Many short, acute injected problems/constraints
  - E.g., exhibitor requests briefing, keynote speaker requests roses, etc
- 3 conditions: COTS, RADAR-L, RADAR+L

Evaluation (continued)

- Before: RADAR wargamed for +L condition
  - E.g., Over 750 email messages for Classifier training
- 2 cohorts of 15 subjects per day (3 hr each)
- Instruction on tools (no hands-on experience)
- Inbox has unread crisis email stack (107 messages)
- Backstory email in separate IMAP folders
  - ~30 high value emails from corpus in a folder
  - ~80 emails for 50 original vendor orders in a folder
- Subject works the problem for 2 hours
- Results are scored and lots of data is collected

Incomplete Actions (11)

Order ▼	Description	Subject	Sender	Created	Modified	Creator
1	Modify Event: *Demo M1: Driver Monitoring Systems*	Attendance figures and new #	Amy Lim <lim12@ardra.org>	Today, 3:32 PM		RADAR
2	Modify Event	note schedule chagnes	Spence Pierro <spierro@ardra.org>	Today, 3:54 PM		RADAR
3	Modify Room: *Flagstaff: Sternwheeler*	Sternwheeler Capacity	Meredith Lorenz <lorenze@pittsburgh.flagstaff.com>	Today, 4:07 PM		RADAR
4	Modify Room: *Flagstaff: Vandergrift*	Sternwheeler Capacity	Meredith Lorenz <lorenze@pittsburgh.flagstaff.com>	Today, 4:10 PM		USER
5	Optimize the Schedule	*no email*		Today, 3:45 PM		RADAR
6	Website Update (VIO): Modify Person: *Austin Parton*	Webpage	Austin Parton <aparton@ardra.org>	Today, 3:37 PM		RADAR
7	Website Update (VIO): Modify Person	Attendance figures and new #	Amy Lim <lim12@ardra.org>	Today, 3:32 PM		RADAR
8	Website Update (VIO)	Organization Wrong	Sonal Malhotra <smalh@ardra.org>	Today, 4:32 PM		RADAR
9	Website Update (WbE)	change phone numbers	Emily Halwizer <halwizer@ardra.org>	Today, 4:47 PM		RADAR
10	Place a Vendor Order	Tech. Request - flip charts	Maggie Foxenreiter <mfox@ardra.org>	Today, 3:33 PM		RADAR
11	Send a Briefing	Brief me, please	Jonathon Robertson <jrobertson@ardra.org>	Today, 4:42 PM		RADAR

Overflow Actions (1)

Order	Description ▼	Subject	Sender	Created	Modified	Creator
	Reply to Question	Vegetarian options?	Sandra Nubanks <snubanks@ardra.org>	Today, 4:02 PM		RADAR

Completed Actions (1)

Order	Description	Subject	Sender	Created	Modified ▼	Creator
	Modify Event: *Workshop 1a: Intermodal Passenger Screening*	Attendance figures	Amy Lim <lim12@ardra.org>	Today, 3:21 PM	Today, 3:45 PM	RADAR

Deleted Actions (1)

Order	Description	Subject	Sender	Created	Modified ▼	Creator
	Modify Speaker's Availability	Planning for History Week	Michelle Randal <mich-randal@gmail.com>	Today, 4:28 PM	Today, 4:34 PM	RADAR

Possibly Conference-Related Emails (1)

Read	Subject	Sender	Date ▼	
•	**for my presentation**	**Laura Timdale** <laurat2@ardra.org>	**Today, 3:24 PM**	**Add an Action**
	Blake, I didnt know who to contact about making sure to have a laptop available, and connected to teh AV equipment - ie projector. I want all that ready on the ...			

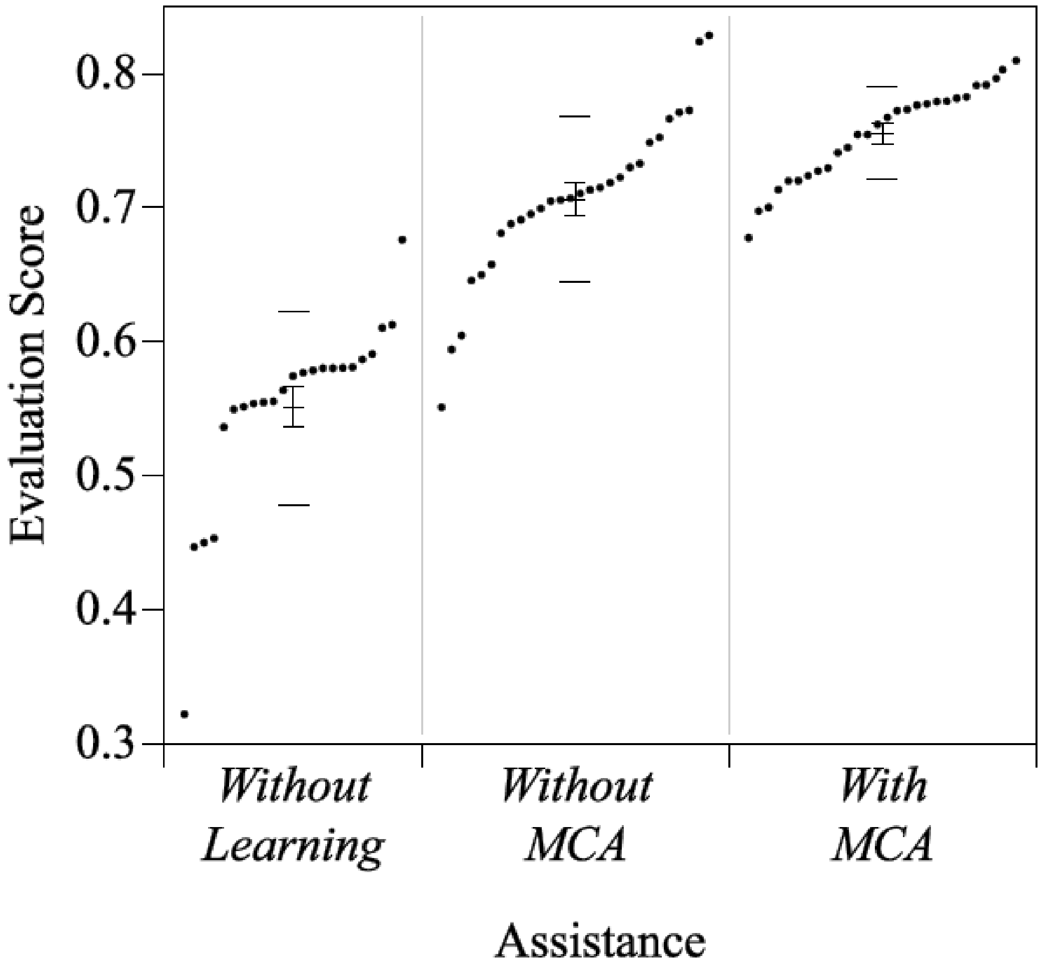
Other Emails (1)

Read	Subject	Sender	Date ▼	
•	**car arrangements**	**Angie Randal** <angiednacer6@gmail.com>	**Today, 3:23 PM**	**Add an Action**
	Ms K is counting on me to help out with the kids' dance class. The car is still in the shp. Can you drop me off over there? thanks :-)			

Deleted Emails (1)

Read	Subject	Sender	Date ▼	
	Precipitation Update	Weather Alerts <weather@weather.gov>	Today, 3:56 PM	Add an Action
	There is a 70% probability for thunderstorms with heavy rain in ALLEGHENY COUNTY this evening through tomorrow. Plan accordingly and be safe! Go to www.weather.gov ...			

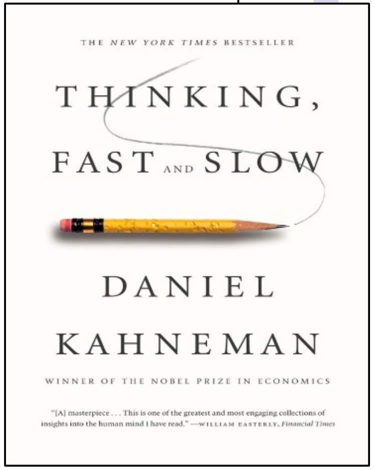
Results



- Improved performance
- Reduced performance variation
- Eliminated long-tail poor performance

Project 3: Hybrid Planning

- *Thinking, Fast and Slow* Kahneman, 2011
  - System 1: Reactive, fast, learned
  - System 2: Deliberative, slow, analytical
- Can systems behave like this?
  - Fast planning when time-critical decisions are needed.
  - Slow planning when there is time to come up with better solutions.



Example 1: Amazon

Amazon web services primarily cares about availability

“AWS will use commercially reasonable efforts to make the Included Services each available for each AWS region with a Monthly Uptime Percentage of at least 99.99%, in each case during any monthly billing cycle (the “Service Commitment”).”*

*https://aws.amazon.com/ec2/sla/

Example 2: Netflix

Netflix primarily cares about maintaining throughput

"When designing customer-facing software for a cloud environment, it is all about managing down expected overall latency of response."*

* http://techblog.netflix.com/2010/12/5-lessons-weve-learned-using-aws.html

What is common?

For both the systems, quick response is needed, particularly, under urgent circumstances, but ideally the response should also be close to optimal in terms of *all* relevant quality attributes

Key Requirements for Planning

- Timeliness - finding an adaptation plan in a timely manner
- Quality - the likelihood of a plan meeting the adaptation goals under the assumption that the plan is available instantaneously, when required

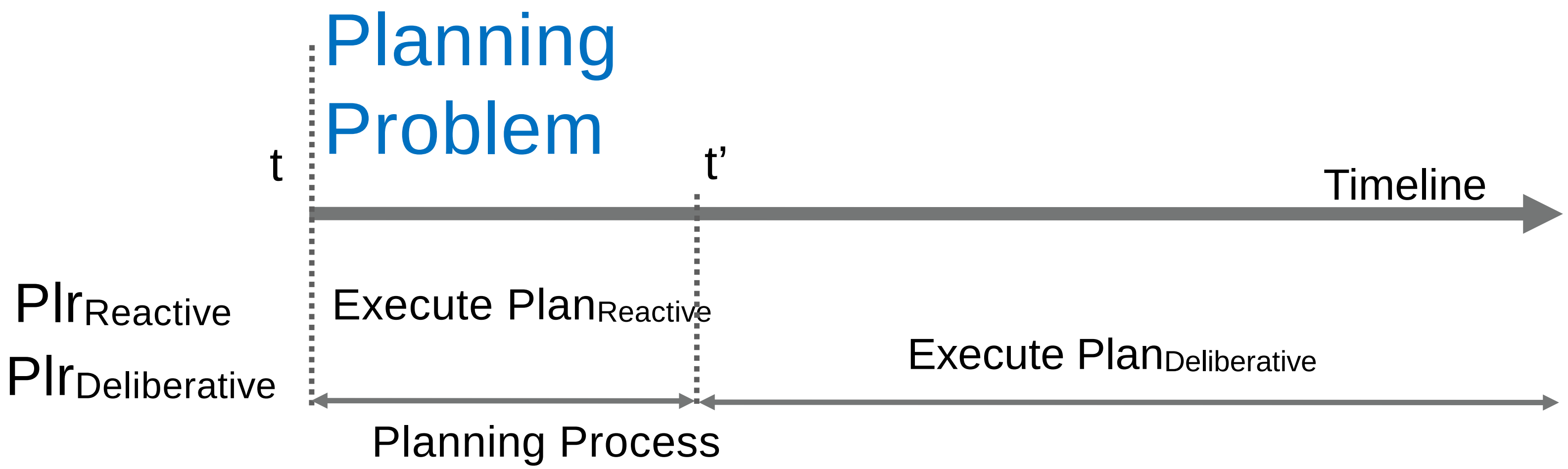
Difficult to decrease planning time and increase quality at the same time

Ashutosh Pandey Thesis

- Shows that you can combine fast and slow planners.
- Challenges that he had to solve
  - How do you make sure slow can take over when it is ready?
  - When is it better to do nothing, because the fast planner might make a big mistake?
  - Can you generalize beyond two planners?
  - How do you know which planners to use in the first place?

Hybrid Planning: Combine off-the-shelf planning approaches

Use reactive planning to provide a quick response (although potentially a sub-optimal), but simultaneously use deliberative planning to provide a plan that improves quality in the long run.



For more information

- Pandey thesis defense: November 27 from 10:00-11:00 GHC 6501.

Conclusions

- AI can be used in many ways to address emerging problems
  - Self-healing, task assistance, time-aware planning
  - Involves many AI technologies – not just ML
- Key issues from a software engineering point of view
  - What lives “around” the AI – architectures!
  - Architectures for AI integration allowing multiple kinds of AI to work together