

# IPTO

INFORMATION PROCESSING TECHNIQUES OFFICE



## SELF: Self-Explanation Learning Framework January 2009

Michael T. Cox



February 2009

Distribution Statement A - Approved for public release, distribution unlimited

- **Self models are absent.**
- **So learners cannot explain anomalies.**
- **Thus systems cannot tolerate surprise.**

## Without Self-Explanation



*MIT vehicle crashes into Cornell car during DARPA Urban Grand Challenge*



# Program Vision



**Goal: Provide machines with an ability to reason about their own reasoning and to explain themselves during learning.**

- **Program Deliverables:**

- A reusable self-explanation module that can be wrapped around existing cognitive systems to improve performance.
- A learning system that helps the knowledge engineer develop cognitive systems.

- **Military Application Domains:**

Armored Combat Missions.  
Tactical Air Missions.



- **Programmatic Approach:**

- Provide cognitive agent as GFE.
- Contractor builds self-explaining module for agent.
- Experimenter inserts bug into agent.
- Human debugs agent with and without self-explanation.
- Debugging gets progressively harder over phases as metric becomes more demanding.



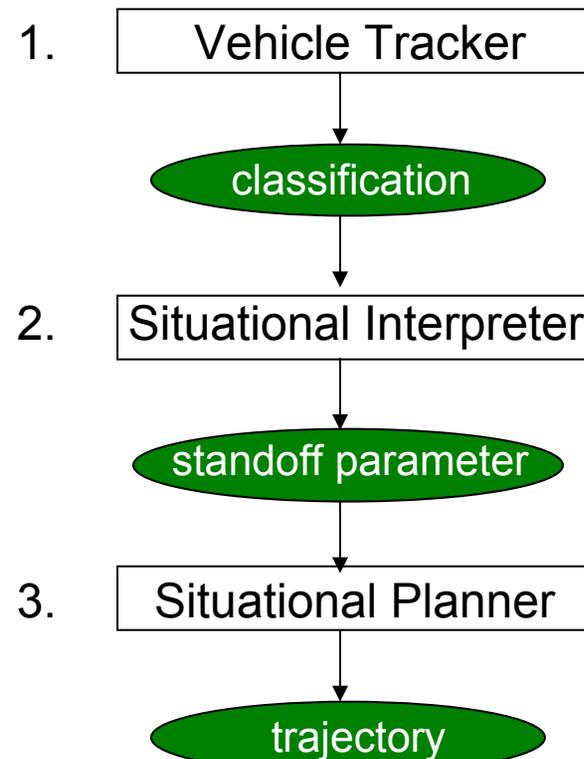
# Example Problem Solution



**After anomaly is detected,**

- **Examine the reasoning trace;**
- **Generate Self-Explanation;**
  - The Cornell vehicle was misclassified as a stationary object.
  - This enabled the passing of a small standoff parameter value.
  - The planning algorithm then generated a trajectory that brought it too close to the opponent vehicle.
- **Learn a more accurate model.**
  - To better handle future situations, bias the classifier in the Vehicle Tracker.
  - Subsequently vehicles will be recognized as requiring greater standoff distance.

## *Reasoning Trace*



**Traditional Machine Learning improves performance without knowing why.**

## Problems:

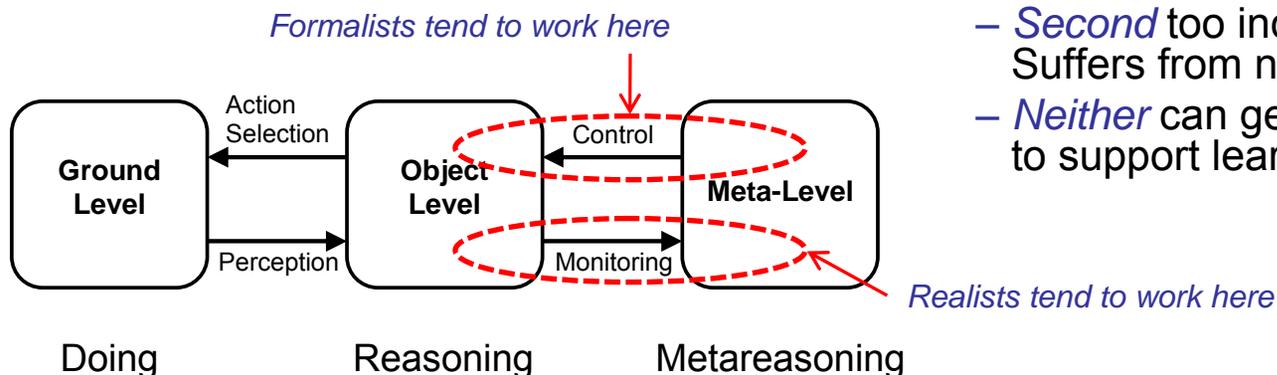
- **Good for modeling rats, not people.**
- **Lacks transparency for human understanding of machine learning.**
- **Current metareasoning research fractured and narrow.**

- **Computational Metareasoning is divided into *formalists* and *realists*.**

- Metareasoning formalists based on decision theoretic principles and statistical reasoning under uncertainty.
- Realists represent common-sense approach with cognitive science and case-based reasoning.

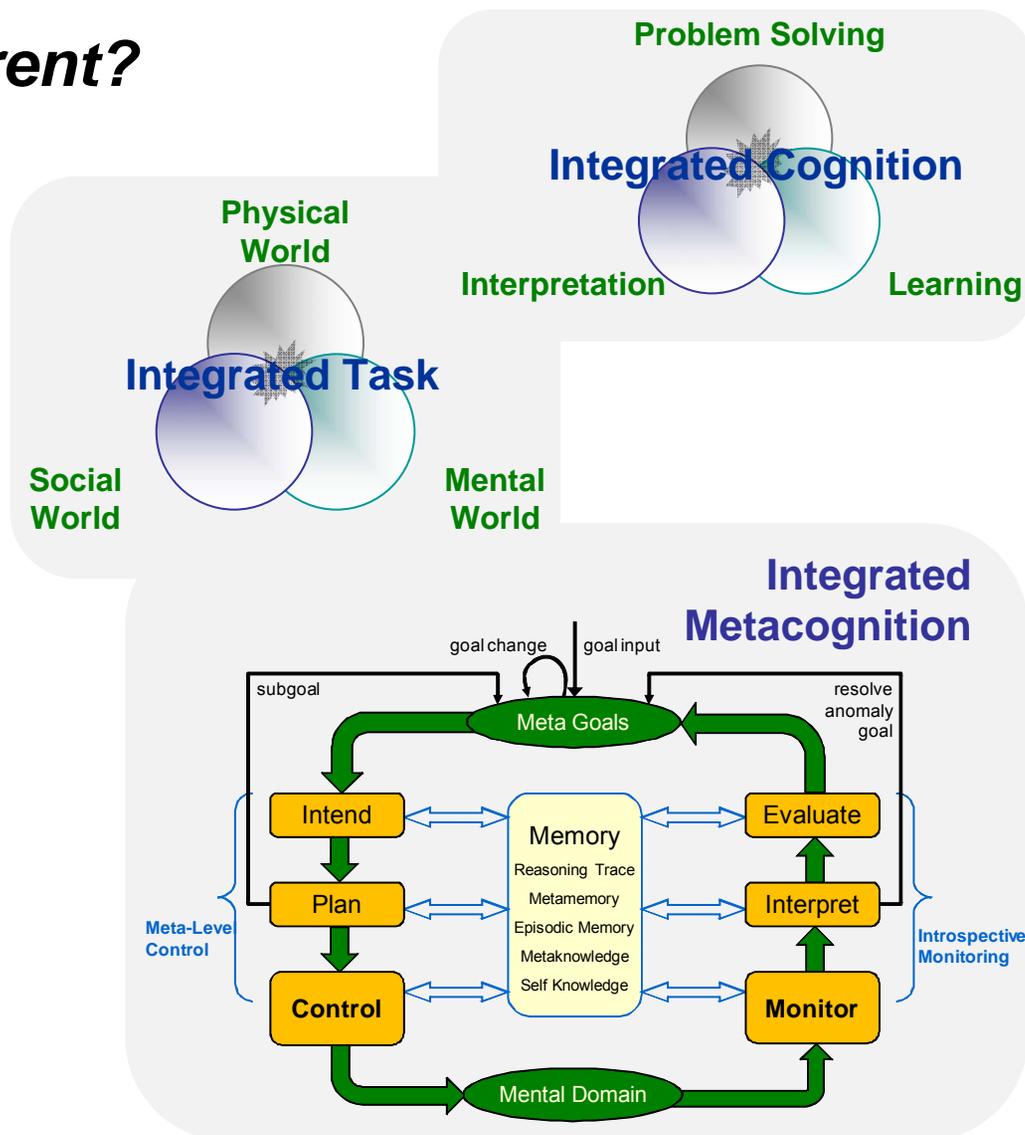
- **Limitations:**

- *First* camp too detailed. Suffers from complexity problems.
- *Second* too incomplete. Suffers from narrow focus.
- *Neither* can generate representations to support learning effectively.



## Why is this program different?

- **Focus fully on meta-level, because base-level agents supplied by government (GFE).**
- **Comprehensive Integration:**
  - **Integrated Task:**
    - Physical + mental + multi-agent performance tasks.
  - **Integrated Cognition:**
    - Problem solving + interpretation + learning.
  - **Integrated Metacognition:**
    - Control of cognition + monitoring of cognition.
    - Metaknowledge + self-knowledge.





# Program Impact



*If a cognitive system understands itself enough to help us debug it, engineering bottlenecks disappear.*

## Long-range Application

### Cognitive-System Engineering

- **Mixed-initiative software development:**  
**Software that helps us build it!**
  - Cognitive systems are so complex that development using traditional methods is becoming intractable.
  - Represents a *revolutionary* change in thinking about software development and learning.
- **Program results increase the Tooth-to-Tail ratio on the software development task.**
  - Reduced software team sizes.
  - Quicker development and maintenance cycles.

## Near-term Targets

### Military Application Domains

- **Armored Combat Missions.**
  - **Tactical Air Missions.**
- Task Benefits:**
- **Improved goal satisfaction through self-explanation and meta-control module.**
  - **Self-explaining systems lead to better calibrated trust for human users.**

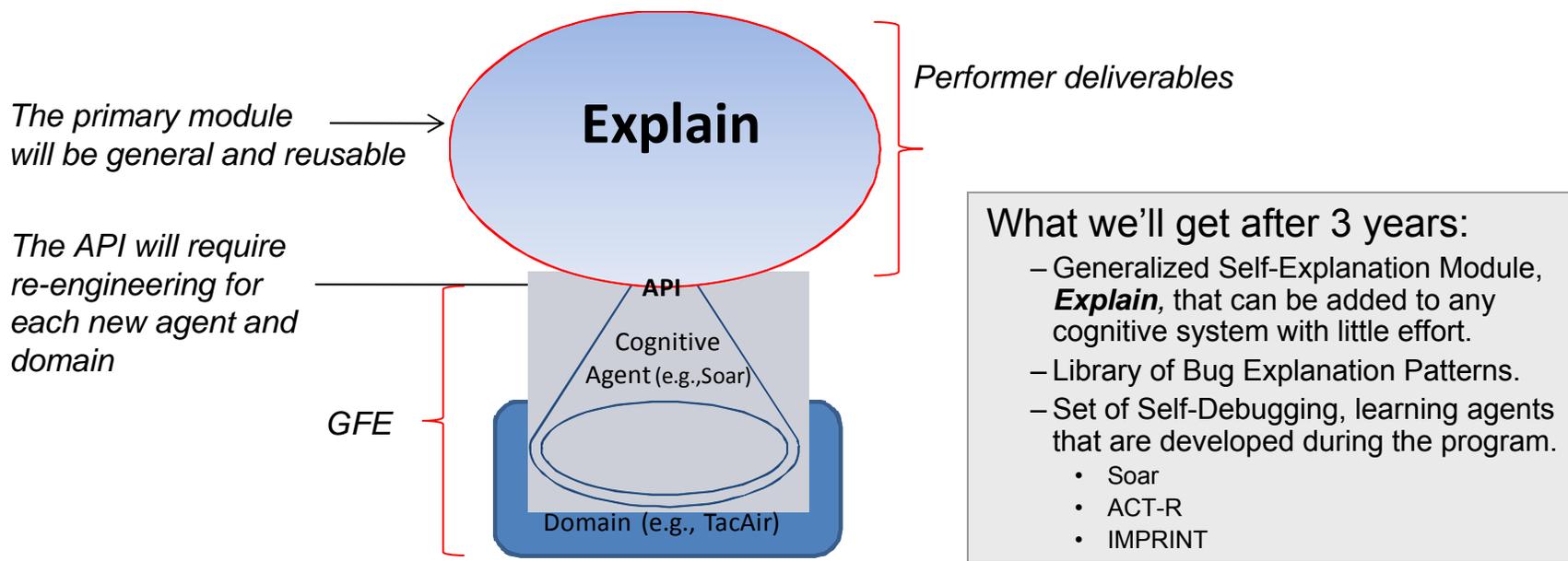




# Supporting Generality



- **Key program deliverable** is to build a generalized software module, *Explain*, that helps cognitive systems explain anomalies.
- **The core of the approaches** will be general and reusable. A thin layer of interface code (i.e., the API) that is specific to the domain and to the cognitive system is necessary to wire *Explain* to the system.
- **To enforce generality**, we will change domains each Phase and we will add a blind government chosen task to the gate. We will give them the agent/task specifics on a Monday and test them on a Friday.

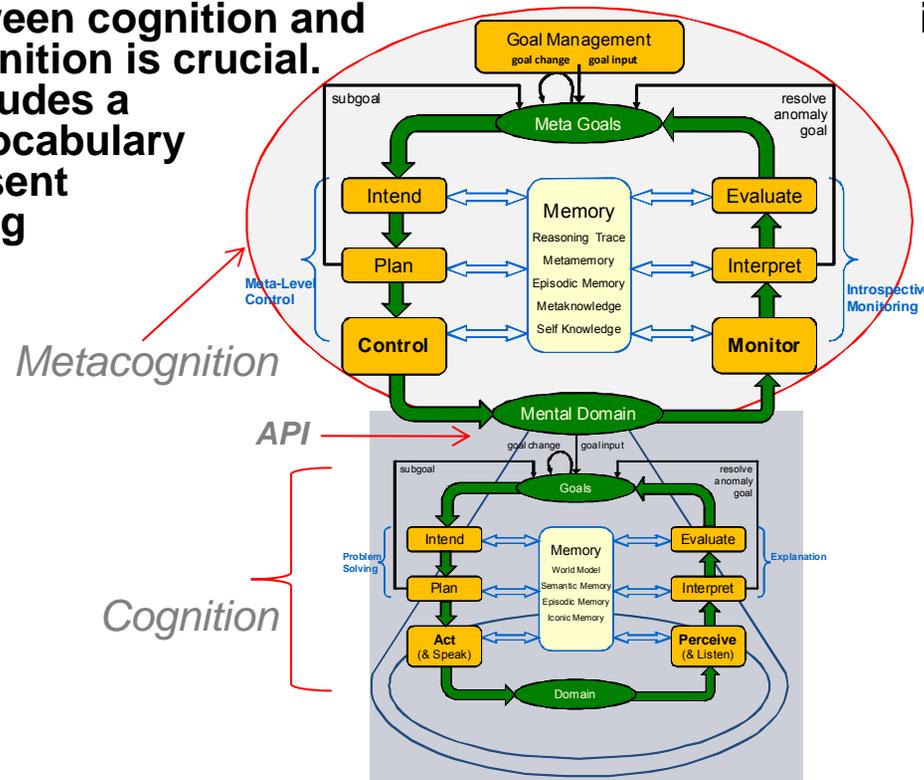


## Risks

- **Combinatorics** increase when adding a meta-level, so performance gain must overcome loss.
- **Cognitive performance task** chosen must be stable and require little development.
- **API** between cognition and metacognition is crucial. This includes a sound vocabulary to represent reasoning traces.

## Payoffs

- Metacognition is the key to **making learning tractable**.
- Self-Explanation helps the system and the user.
- Metacognition is key to establishing effective learning in a **multi-agent context**.
  - Given right architecture, the systems will explain themselves, debug themselves, and understand themselves.
  - Understanding oneself leads to understanding others.





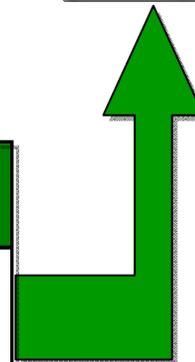
A=not(Safe); E=Safe → weapon-range too short

## Anomaly Cause (Fault) Taxonomy

	Knowledge States		Goal States		Strategy		Environment	
	Domain Knowledge	Knowledge Selection	Goal Generation	Goal Selection	Processing Strategy	Strategy Selection	Input	Input Selection
Missing	Novel Situation	Missing Association	Missing Goal	Forgotten Goal	Missing Behavior	Missing Heuristic	Missing Input	Missing Context
Incorrect	Incorrect Domain Knowledge	Erroneous Association	Poor Goal	Poor Selection	Flawed Behavior	Flawed Heuristic	Noise	Incorrect Context
Correct	Correct Knowledge	Correct Association	Correct Goal	Correct Association	Correct Behavior	Correct Choice	Correct Input	Correct Context

## Anomaly Symptom Taxonomy

	Expectation (E) exists	Expectation (E) does <i>not</i> exist
Actual (A) event exists	Contradiction or Unexpected Success	Impasse or Surprise
Actual (A) event does <i>not</i> exist	False Expectation or Self-fulfilling Prophecy	Missed Opportunity



- **Explicit Mental Representations** store memory traces of *how* reasoning occurred.
- **Meta-Level Monitoring** of traces produces explanation of *why* reasoning failed.

*Explain works by using a Symptom-to-Fault mapping*

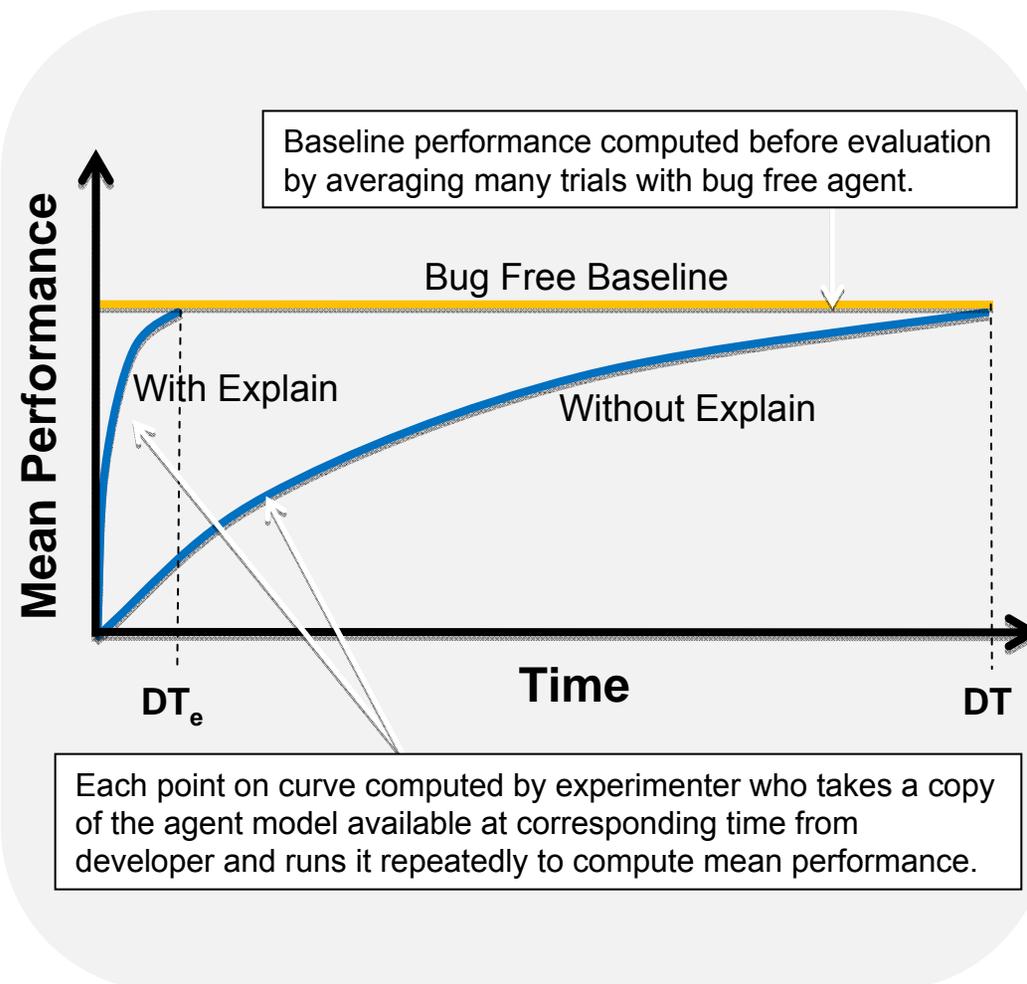


# Research Assessment



- Metric is *Reduction in Debugging Time*.
- Technology developed under SELF will lead to a *tenfold reduction* in cognitive-agent debug time, while restoring baseline domain performance.
- This improvement will be shown for agents having multiple, interacting bugs across *all* bug categories.

- DT = Manual Debugging Time.
- $DT_e$  = Debugging Time with Explain.
- Reduction of Debugging Time =  $(DT - DT_e) / DT \geq 90\%$ .





# Phase Gates



## Go/No Go Overview

- DARPA provides cognitive agent that can achieve goals in domain.
- Central task is to debug agent.
  - Manually: Human debugging and code repair without *Explain* component.
  - Mixed-initiative: *Explain* aids human in testing, analyzing, and fixing agent system.
    - Self-explanation maps performance problems to bugs at knowledge level.
    - Human uses explanation to better understand code level and make fix.

## Experiments

- For each trial (problem), experimenter inserts into agent model multiple, interacting bugs from *any* bug category.
- Humans tested in two conditions: with and without *Explain* turned on.
- Reduction of Debug Time averaged across multiple trials.
- Contractors compete for greatest Reduction of Debug Time above minimal threshold.

	Phase I	Phase II	Phase III
Number of Domains	1	2	3
Bug Types	4	4	4
Objective Function returns to baseline	100%	100%	100%
Reduction in Debug Time	$\geq 30\%$	$\geq 60\%$	$\geq 90\%$

## NEW DOMAIN EACH PHASE



Armored  
Combat



Tactical Fighter  
Missions



Time Critical  
Targeting (AOC)

## BUG TYPES



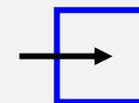
Knowledge



Goal



Strategy



Environmental  
Input



# Self-Explanation Learning Framework



## SELF ACHIEVEMENT

STATUS QUO

**Machine Learning Improves Performance without Knowing Why**

**Problems:**

- Good for modeling rats, not people
- Lacks transparency for human understanding of machine learning
- Current metareasoning research fractured and superficial

**Self-Explanation Enables Improved Learning**

**Self-Explanation** is model-based symptom to fault mapping:

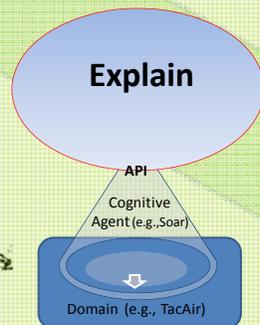
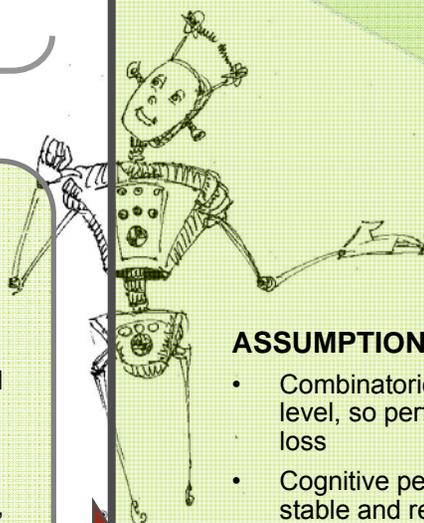
- **Failure Symptoms:** Contradiction, surprise, impasse, unexpected success, false expectation
- **Failure Faults:** Knowledge, goals, processes, environment

NEW INSIGHTS

**MAIN ACHIEVEMENT:**  
**Introspective Cognitive Learning Agent**

**HOW SELF-EXPLANATION WORKS:**

1. **Explicit Mental Representations** store memory traces of *how* reasoning occurred
2. **Meta-Level Monitoring** of traces produces explanation of *why* reasoning failed
3. **Introspection** enables construction of explicit learning strategy driven by self-model



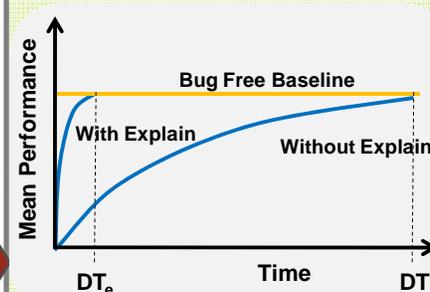
**ASSUMPTIONS AND LIMITATIONS:**

- Combinatorics increase when adding a meta-level, so performance gain must overcome loss
- Cognitive performance task chosen must be stable and require little development
- API between cognition and metacognition is crucial. This includes a sound vocabulary to represent reasoning traces

QUANTITATIVE IMPACT

**Evaluation of Self-Debugging**

- Reduction in Debug Time = Time Savings/ Manual Time



END-OF-PHASE GOAL

**Self-Debugging Applications**

**Learning systems that can debug themselves**

- Start with correct performance system, then insert known bugs
- System perceives degraded performance, then explains
- Test learning with and without metacognitive module



**If a cognitive system truly understands itself, it can explain how it learns**



# QUALIFICATION



**The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.**

Approved for public release, distribution unlimited.  
Contact Michael Cox ([Michael.Cox@DARPA.mil](mailto:Michael.Cox@DARPA.mil)).