

# Maven:

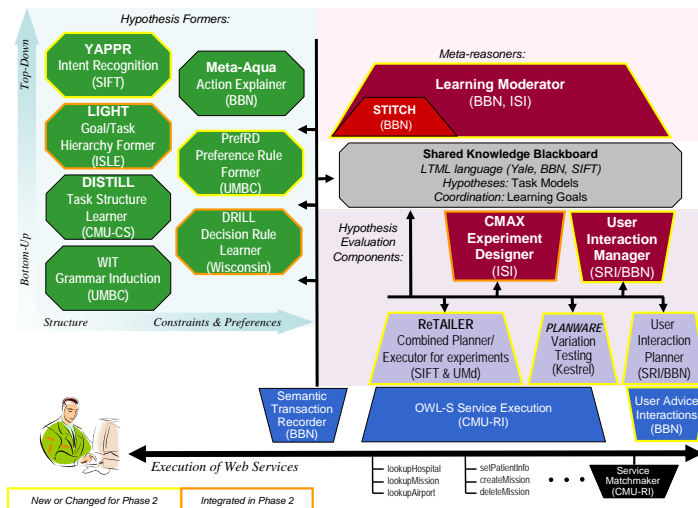
## Developing a Meta-Level Problem Solver for Integrated Learners

Jihie Kim and Yolanda Gil  
Information Sciences Institute  
University of Southern California

[jihie@isi.edu](mailto:jihie@isi.edu), [gil@isi.edu](mailto:gil@isi.edu)

# Background: Integrated Learning System

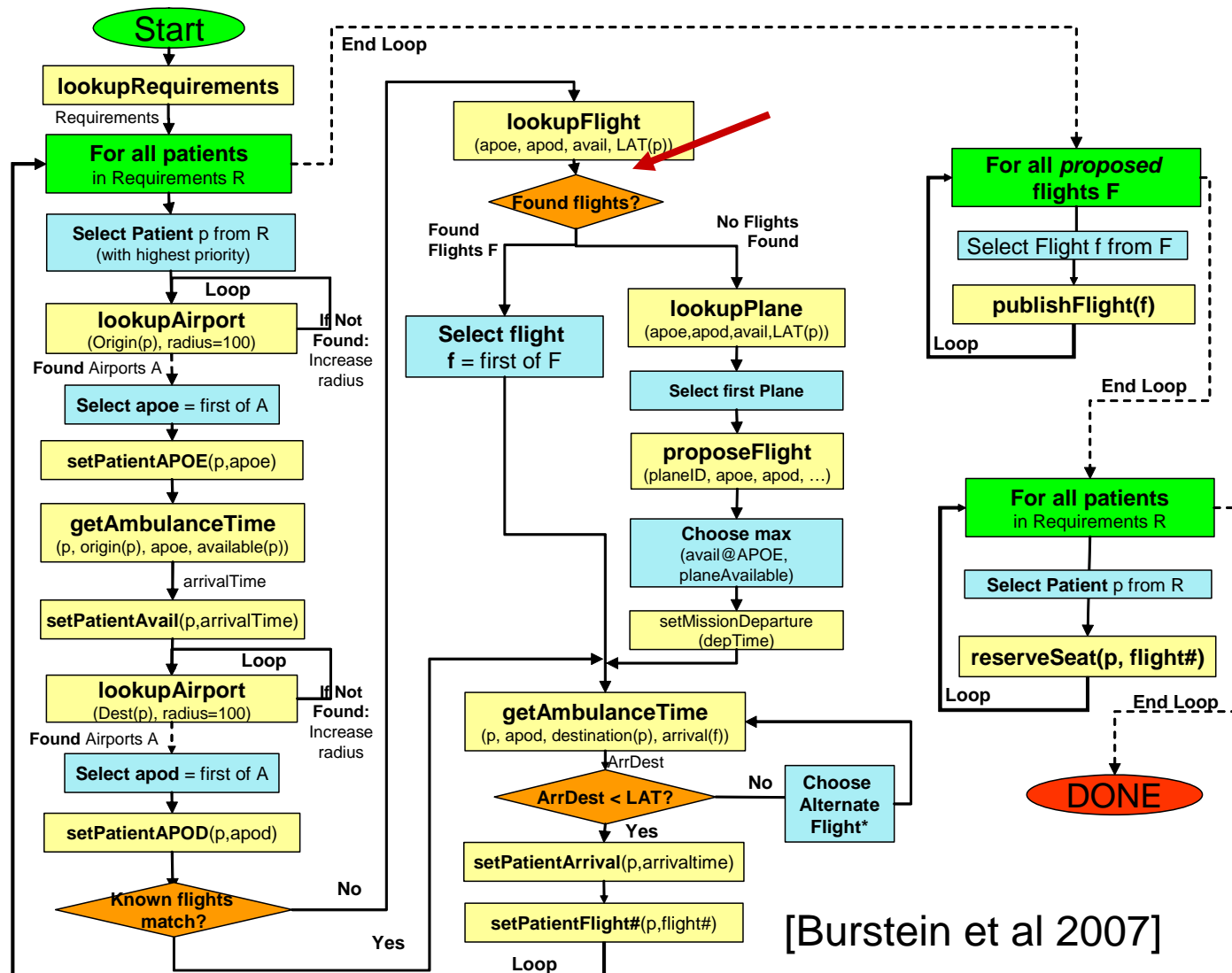
- Poirot [Burstein et al., AAAI-2008; Burstein et al 2007]
  - Learn how to perform complex procedures from an expert demonstration trace
  - Integrate a wider range of learning methods with different strengths
  - Knowledge validation with experiment and practice problems



## Integrated Learners

- Trace interpreters
- Trace generalizer
- Trace explainer
- Hypothesis integrator
- Workflow constructor
- Workflow executor
- Experiment designer
- Question Asking
- ...

# Learning Complex Procedural Knowledge from One Expert Demo Trace

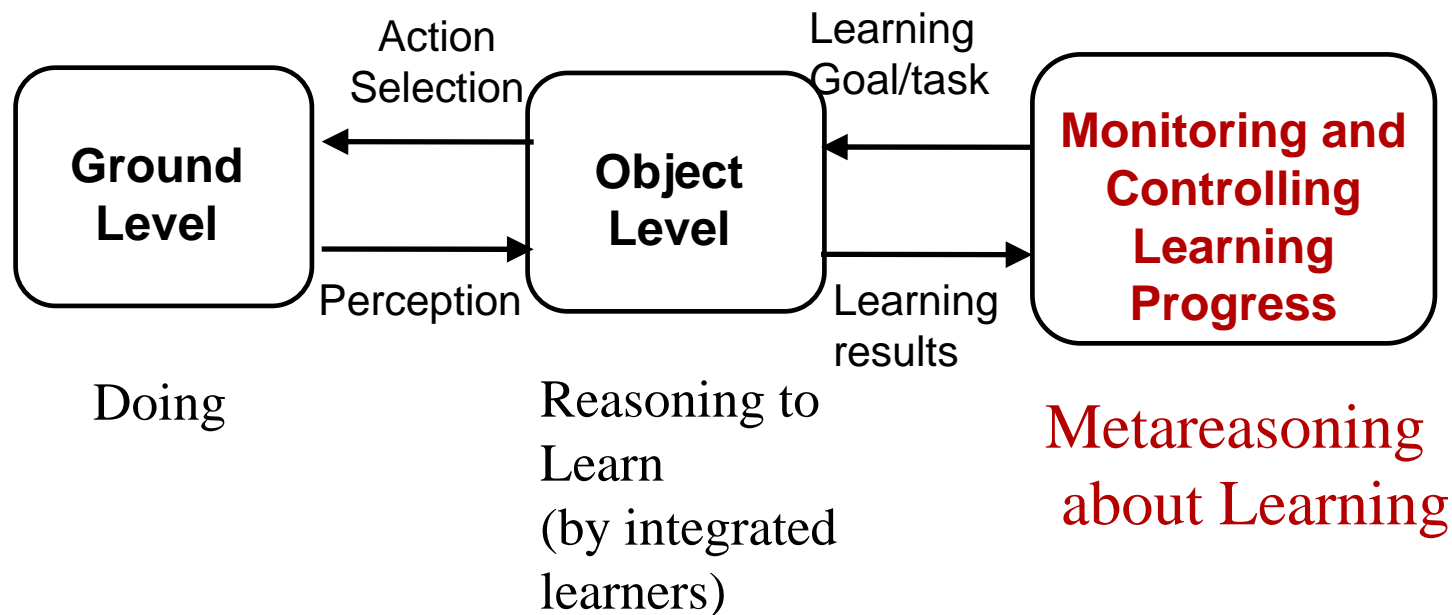


- Step Orderings
- Branches and loops
- Preconditions
- Task decompositions
- Object selection criteria
- ...

[Burstein et al 2007]

# Meta reasoning for integrated learners

- Coordinate activities of different agents based on their capabilities and status of learning



# Challenges

- Assess learning status and progress of learning over time
  - Systematically addressing issues (gaps, competing hypotheses and failures) that arise during the learning.
- Reason about what to do next
- Supporting flexible interactions among agents that pursue different learning strategies.

# MAVEN

(**M**oderating **A**cti**V**iti**E**s of **i**Ntegrated learners)

A meta-reasoner for organizing activities by integrated learners and other agents

Inspired by

- PRS-style BDI (belief, desire and intention) agents[Ingrand et al.,92]
- Meta-level reasoning capabilities for interactive knowledge capture [Kim and Gil 2007; Kim and Gil 2003]

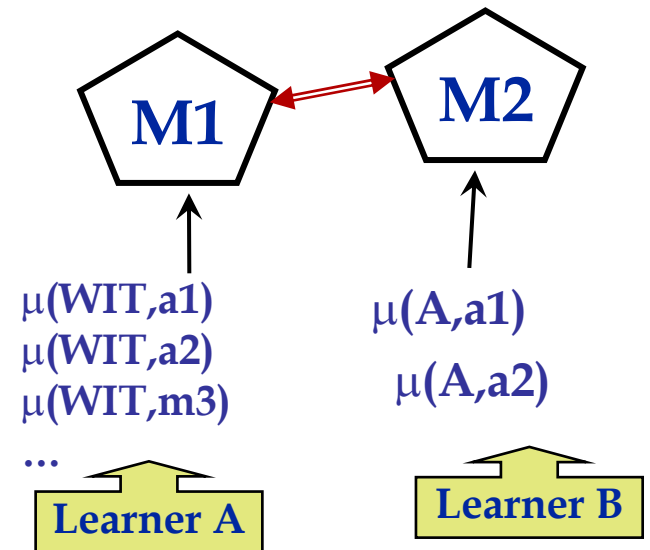
# MAVEN

A meta-reasoner for organizing activities by integrated learners and other agents

- **Internal state:** assess learning status and reason about what to do next
  - **Meta-Models:** keep track of body of knowledge formed and changes over time
  - **Meta-Goals and Meta-Plans:** Explicit representations of learning goals and a set of meta plans to reason about the situation and create goals
  - **Meta control criteria and goal selection strategies**
- **External state:** the blackboard where all agents post results
  - **Effectors(control):** Maven posts meta-goals (tasks) to blackboard that can be achieved by other learning agents
  - **Sensors (monitor):** Maven access learning results from agents

# Meta Models

- Maven creates Meta Models for knowledge formed or changed by agents
  - Interpretation of Input data
    - e.g. interpretation of expert demo trace
  - Methods, Choice criteria, Conditions, ...
- Maven assesses individual Meta Models
  - Issues, coverage
  - Workflow generatable or not
  - Validation status
- Maven relates MMs to one another
  - e.g. superseding, competing, find best models

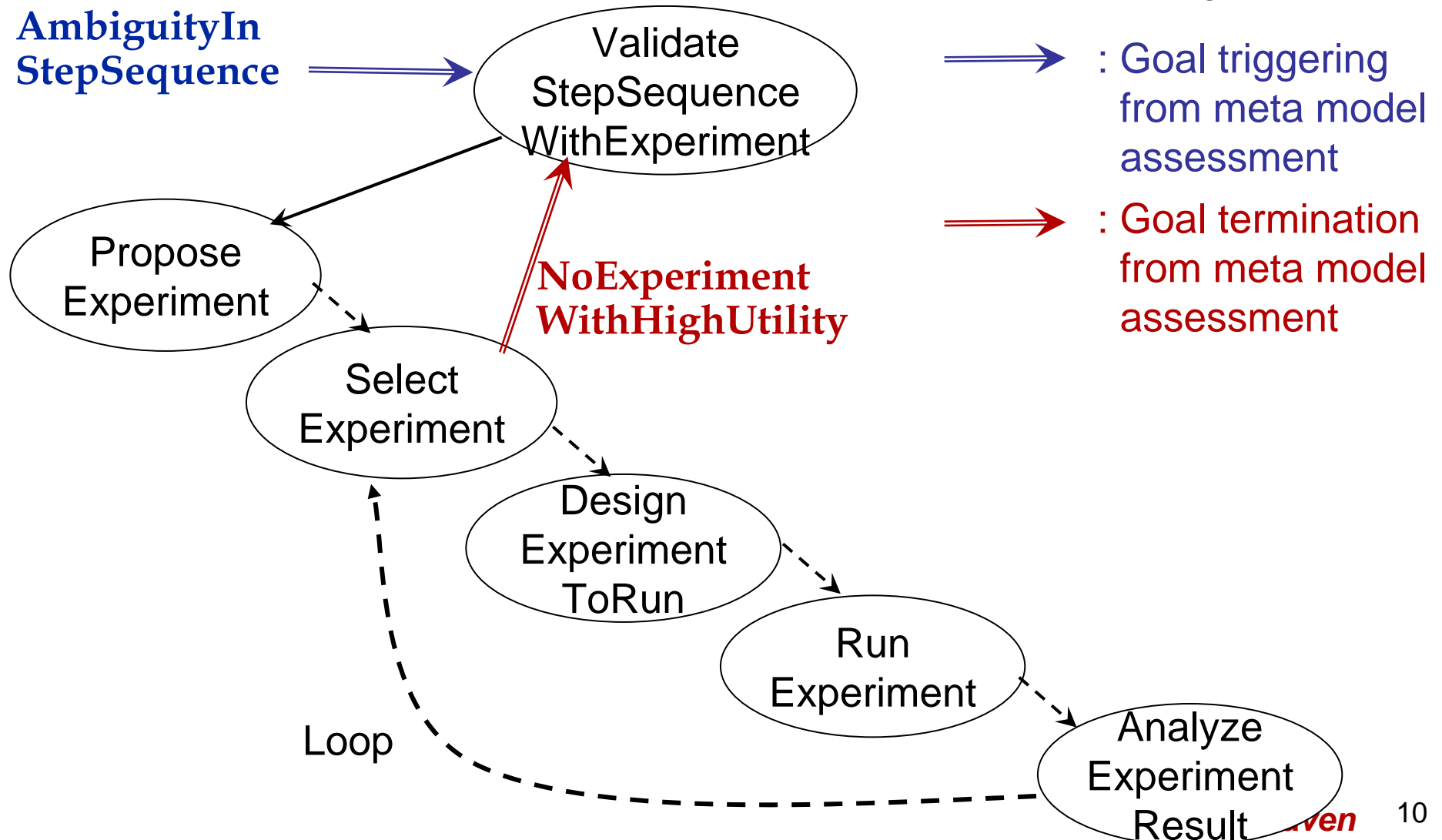




# Explicit Meta Goal Hierarchy in Meta KB (for Experiment Control)

: Subgoal

**AmbiguityIn  
StepSequence**



# C4ML: Four Control Criteria to Drive Learning

Built on Knowledge Acquisition Principles (Kim and Gil, IJHCS 2007)

- **Capability: achieve skill for one (or few) situations**
  - Example: I am capable of driving since I know the rules
  - System has a model about each action or method that can be used to generate+execute at least one plan (e.g. the user's demonstration)
    - Succeeds sometimes
- **Confidence: achieve skill for well-specified situations**
  - Example: I am confident that I can drive from the airport to my house
  - System has used models about each action or method successfully for a well-specified kind of situation
    - Knows when it will succeed
- **Coverage: achieve skill for a broad range of situations**
  - Example: I could drive a stick-shift and can handle NY traffic
  - System is able to generate+execute plans for a broad range of situations
    - It will succeed in a broad range situations
- **Competence: achieve skill that is not suboptimal**
  - Example: I can drive to the airport in less than 30 mins
  - System is able to generate+execute good quality plans for a range of problems
    - It can succeed by choosing better plans

# Phases of Learning

Build  
Capability



Increase  
Confidence



Increase  
Coverage

LearnFrom  
ExpertDemoTrace

ValidateK By  
ReproducingTrace

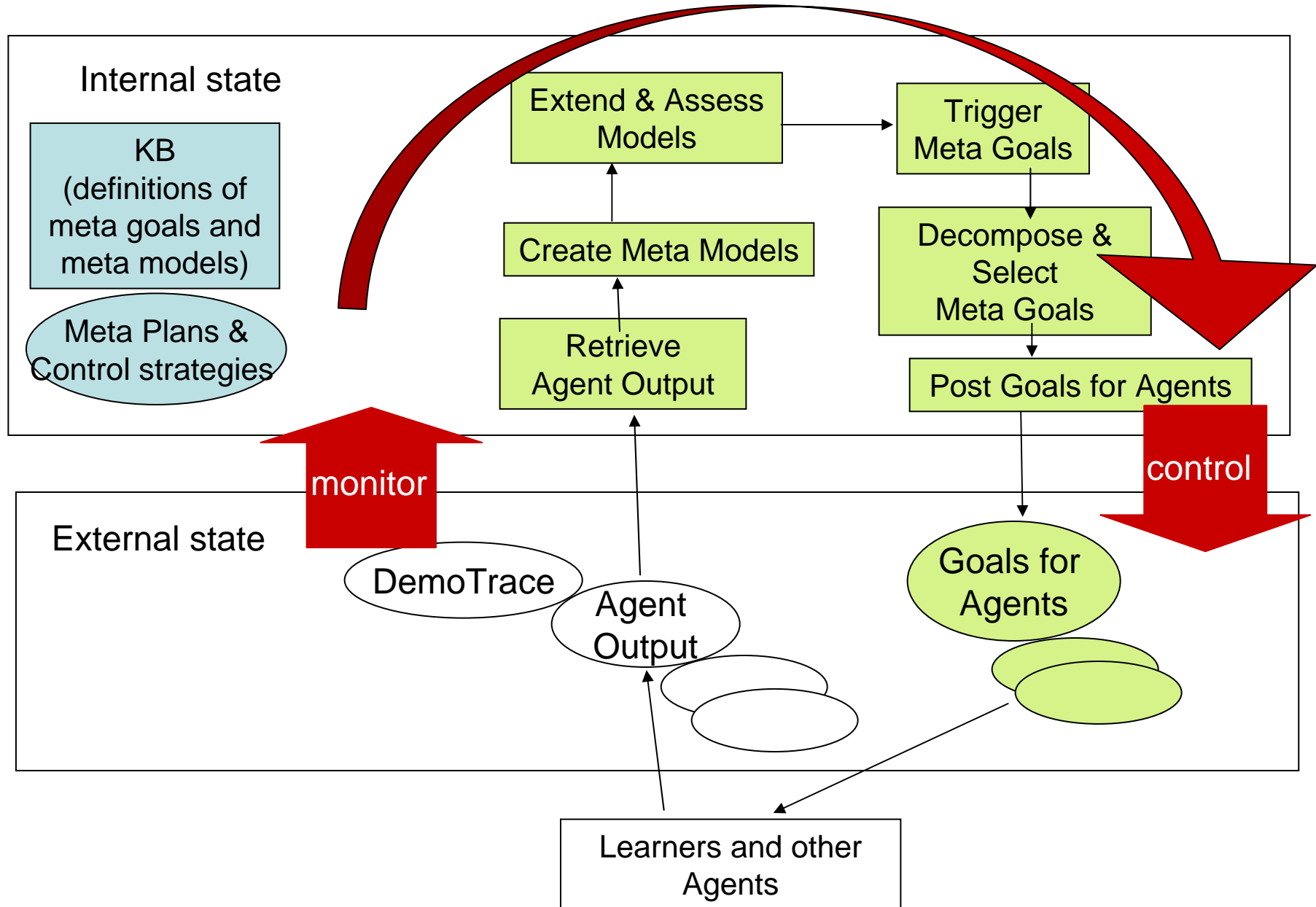
Check K  
with Practice  
Problems

ValidateK  
WithExperiment

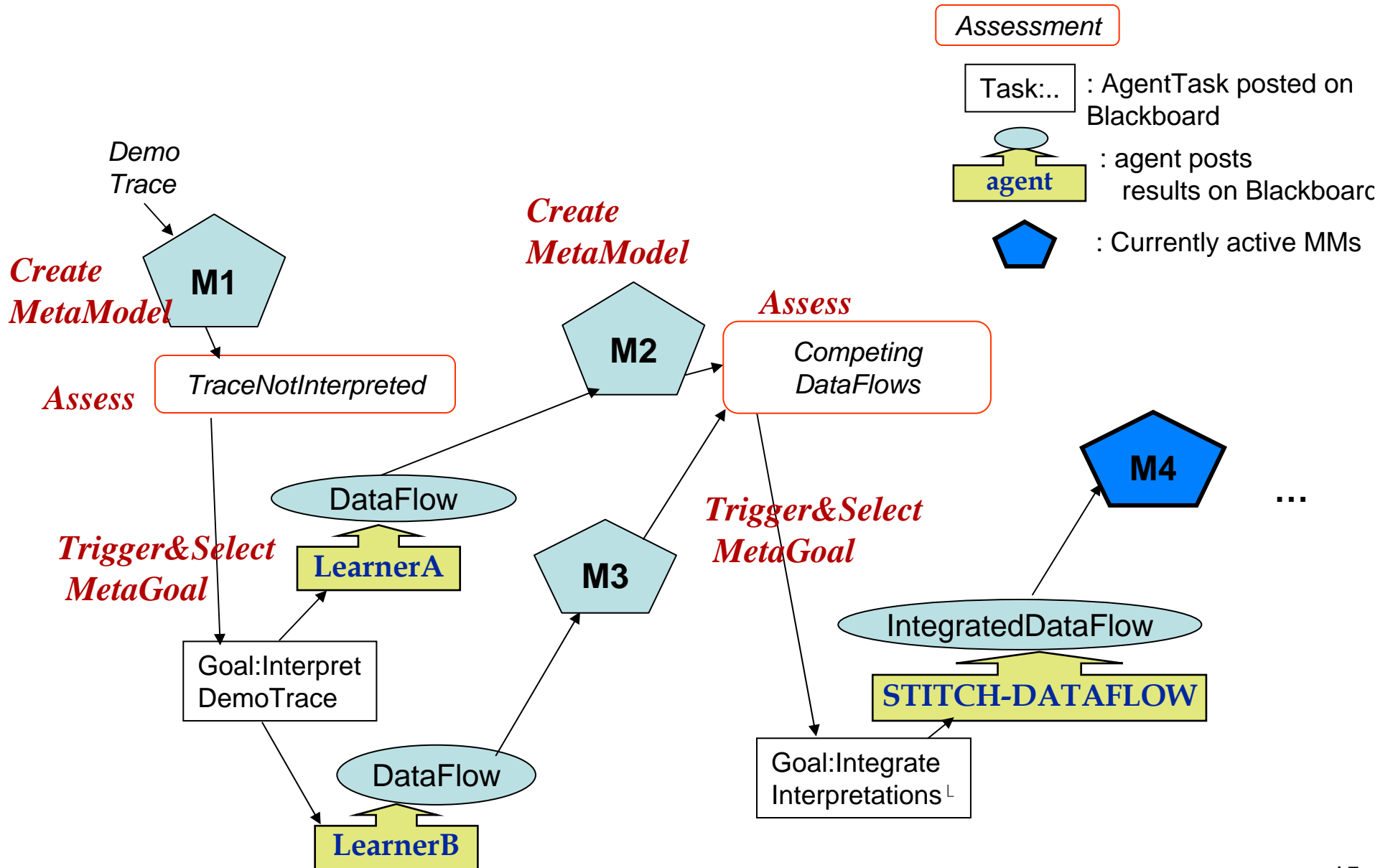
## 3-Layer of Meta-Reasoning

- Top-level layer: C4 criteria
  - Increase capability, confidence, coverage, (competence), currently in different *phases*
- Middle-layer: Meta Model and Goal assessment
  - Meta Model assessment: issues, coverage, status, ..
    - ⇒ Trigger and create meta goals
  - Assess cost/benefit of pursuing goals
    - ⇒ Influence goal selection
- Bottom-layer: Meta Models are created, refined, tested
  - Post meta-goals to be resolved by learners, planner, experiment designer etc.

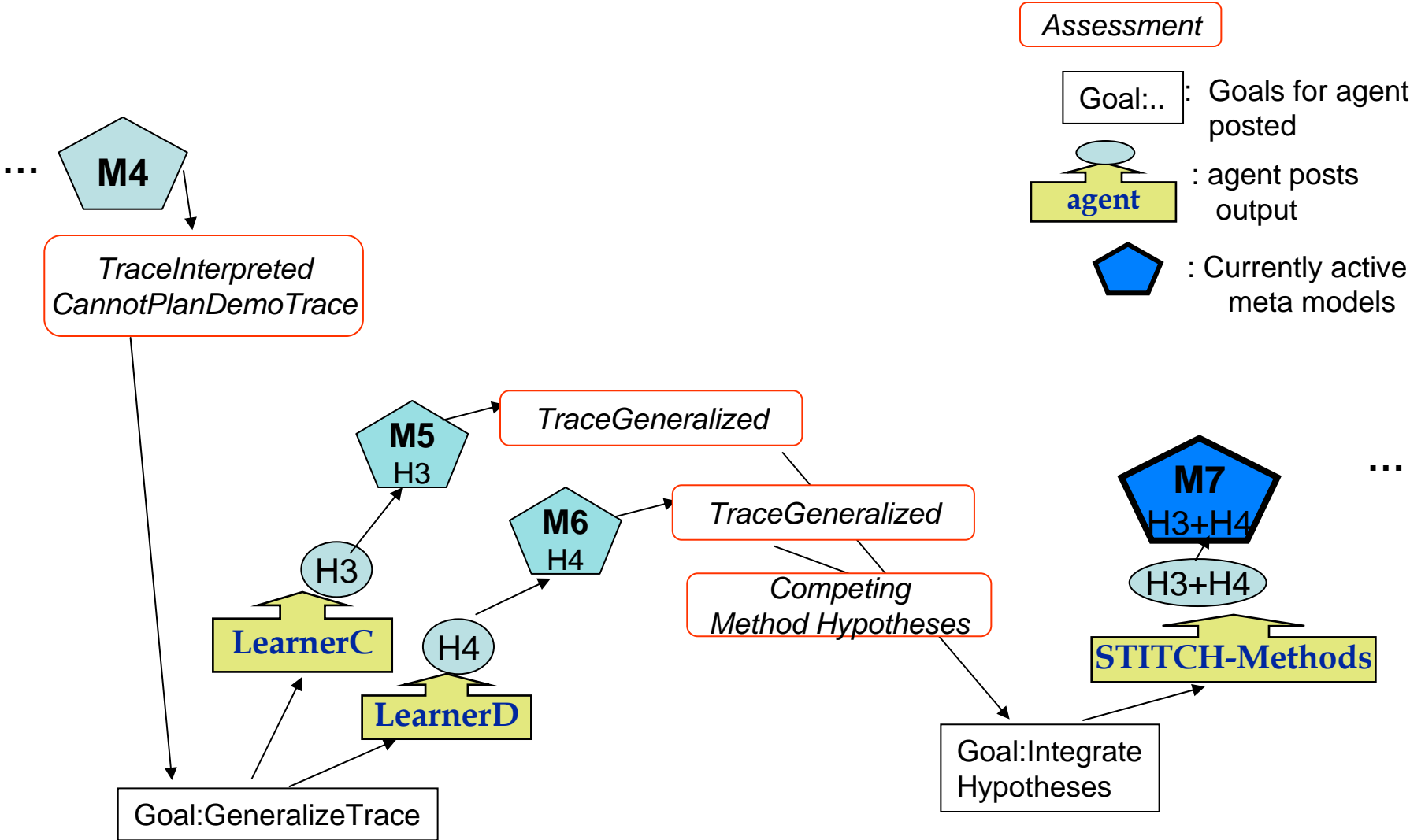
# Maven Cycle



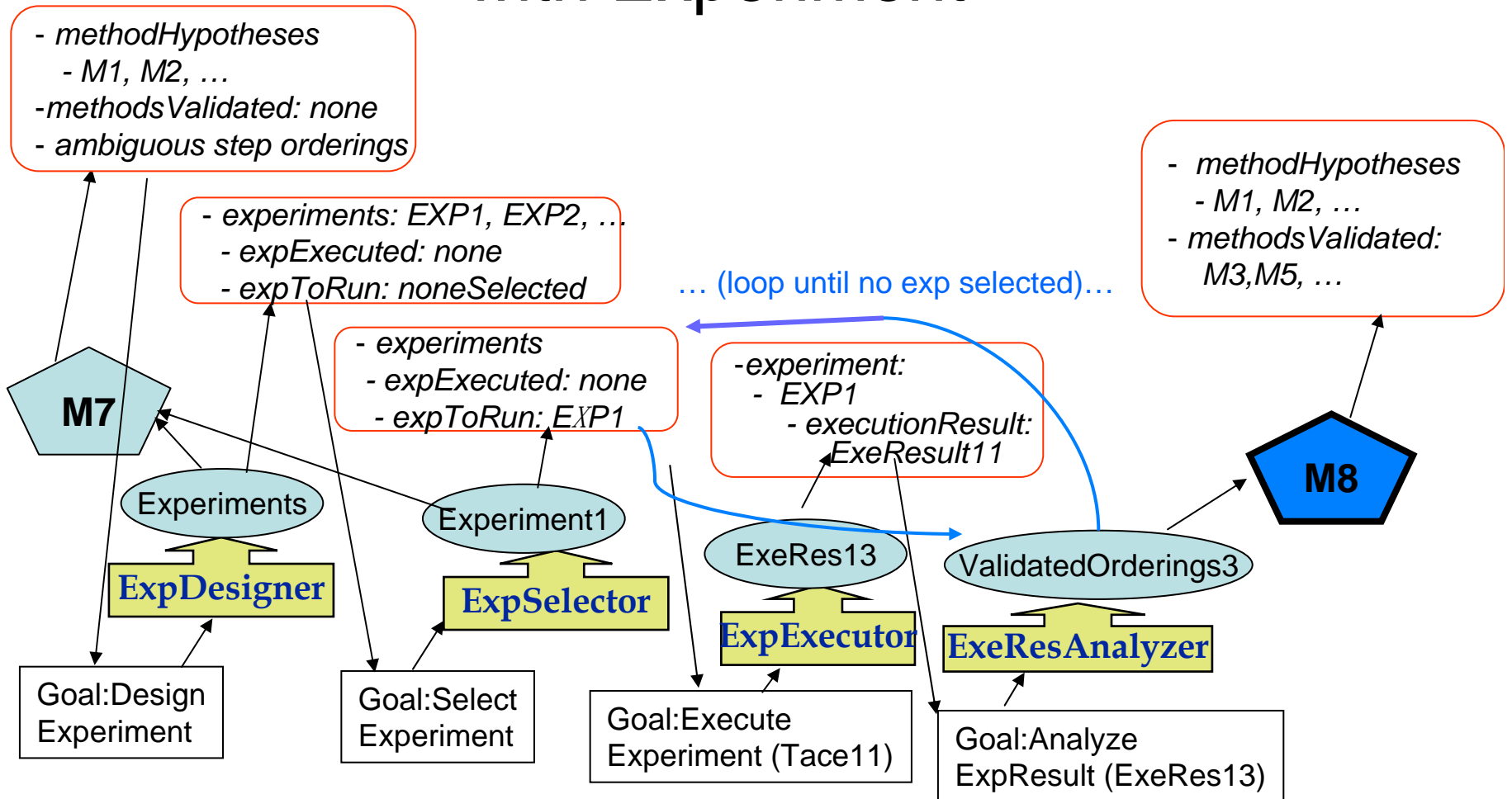
# Interpretation of Demonstration Trace



# Generalize Trace: Generation and Creation of Method Hypotheses



# Increase Confidence on Step Orderings with Experiment

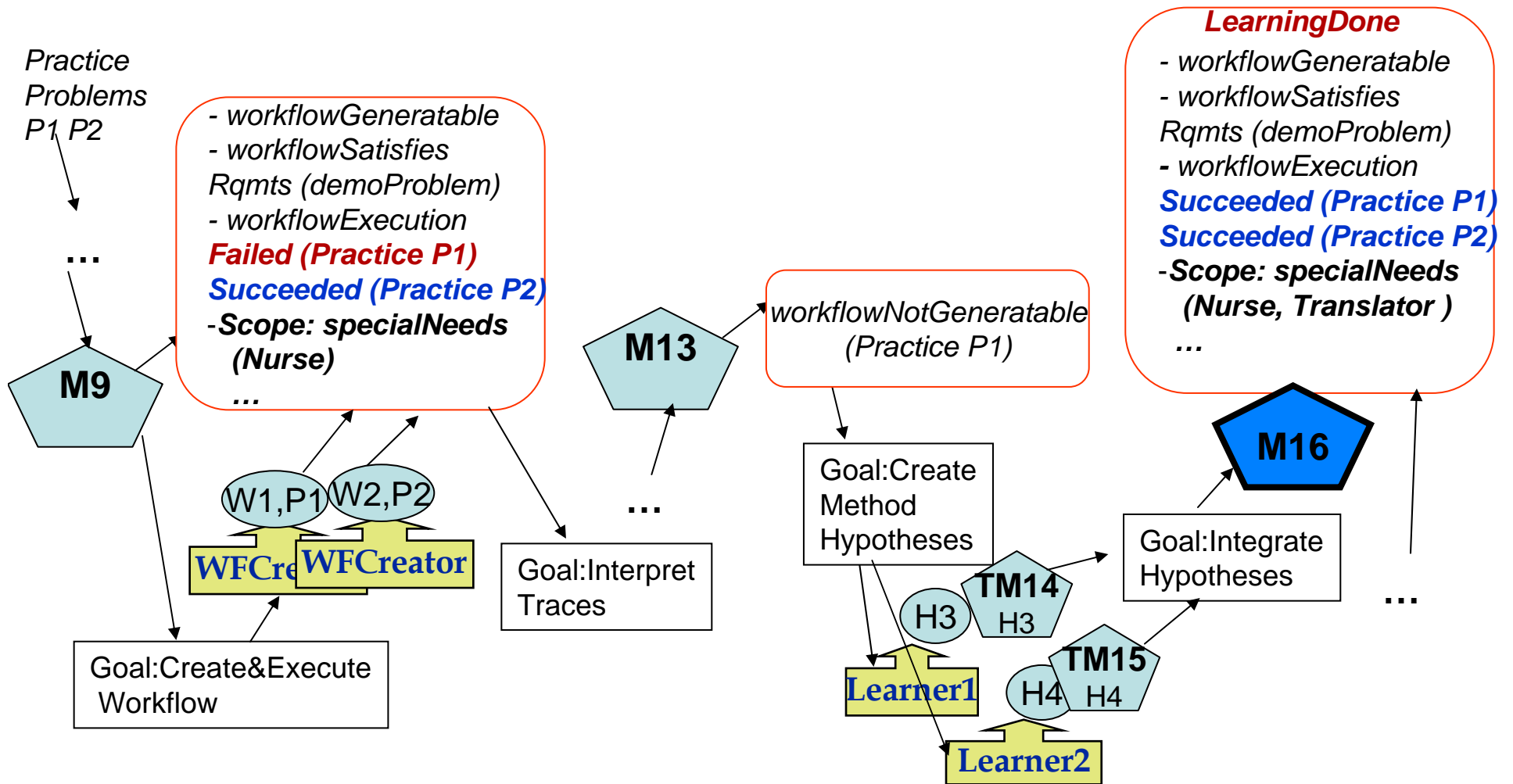


# Meta Reasoning for Experiment Control

(joint work with Clayton Morrison)

- **Deciding when to perform experiment**
  - Ambiguity in step orderings found during hypotheses integration
- **Deciding which experiment to perform**
  - Utility estimation for experiments (experiment execution goals)
    - Cost: number of remaining experiments in testing the sequence
    - Benefit: number of methods covered by the sequence
- **Deciding when experiment is done**
  - No experiment has high utility

# Increase Coverage



## Related Work

- Acquiring complex knowledge from user instructions (Allen et al., 2007)
- Learning goals for question asking (Gervascio and Myers 2008)
- Goal-driven approaches for learning (Ram and Leake 1995)
- Combining results from multiple learners (Dietterich 2000)
- Utility models for agent control (Russell and Wefald 1991)

# Current Status

- **Explicit hierarchical relations among goals and subgoals in ontologies of goal trees (in OWL)**
  - augmented with trigger conditions and termination/achievement conditions
- **Simple goal selection strategy**
- **Meta Model Assessment is simple**
  - One assessment annotation per meta model
  - Relating models: competing ones only
  - One best model
- **A simulation of agent interactions during demonstration interpretation and generalization**

# Future Work

- **Meta Plans and Meta Goals**
  - More complex non-hierarchical relations among goals and subgoals
  - More complex goal selection strategies
  - Apply competence criteria
    - e.g. Learn efficient procedures
- **Meta Models and Assessment**
  - Learning results are interpreted based on issue details
  - Rich assessment of meta models
    - Credit and blame assignment
  - Diverse relations among meta models
  - Handle multiple best models
  - Assessment based on learning history
- **Full integration with learning agents**