Reflection in Action: Meta-Reasoning for Goal-Directed Autonomy

Ashok K. Goel

Design & Intelligence Laboratory
Georgia Institute of Technology

AAAI-2010 Workshop on Goal-Directed Autonomy
Atlanta, July 2010

Domain: Interactive Games

Freeciv (www.freeciv.org) is a popular, interactive turn-based strategy game.

A human plays the game against multiple software agents.

The goal is to conquer the world.

Case Study in Freeciv

- >300 changes to Freeciv in a ~4 month period.
- Most (>90%) changes to Freeciv are small Δs.
- None of the Δs are about goals as such.
- But about two thirds are changes to constraints and resources that impact achievement of goals.
- All of these changes to Freeciv require goal-directed proactive modifications to the Freeciv agents (and not failure-driven).

Adaptation Scenario - 1

New constraints:

An agent may not declare war on another player until more than N (say, 100) fighting units have been built.

Note that the goal of the agent remains the same. But the new constraint may impact mechanisms for achieving the goal.

Adaptation Scenario - 2

New resources:

Luxury resources on the map (e.g. gems, silks) will now contribute to happiness of the populace of a city if the city is connected (via roads, railroads) to the resource.

Note again that the goal of the agent remains the same. But the new resource may impact mechanisms for achieving the goal.

Why is this hard?

- . Complexity of the environment
  - - Huge state space
  - - Interacting goals, actions
  - - Partially observable environment
  - - Non-deterministic game playing

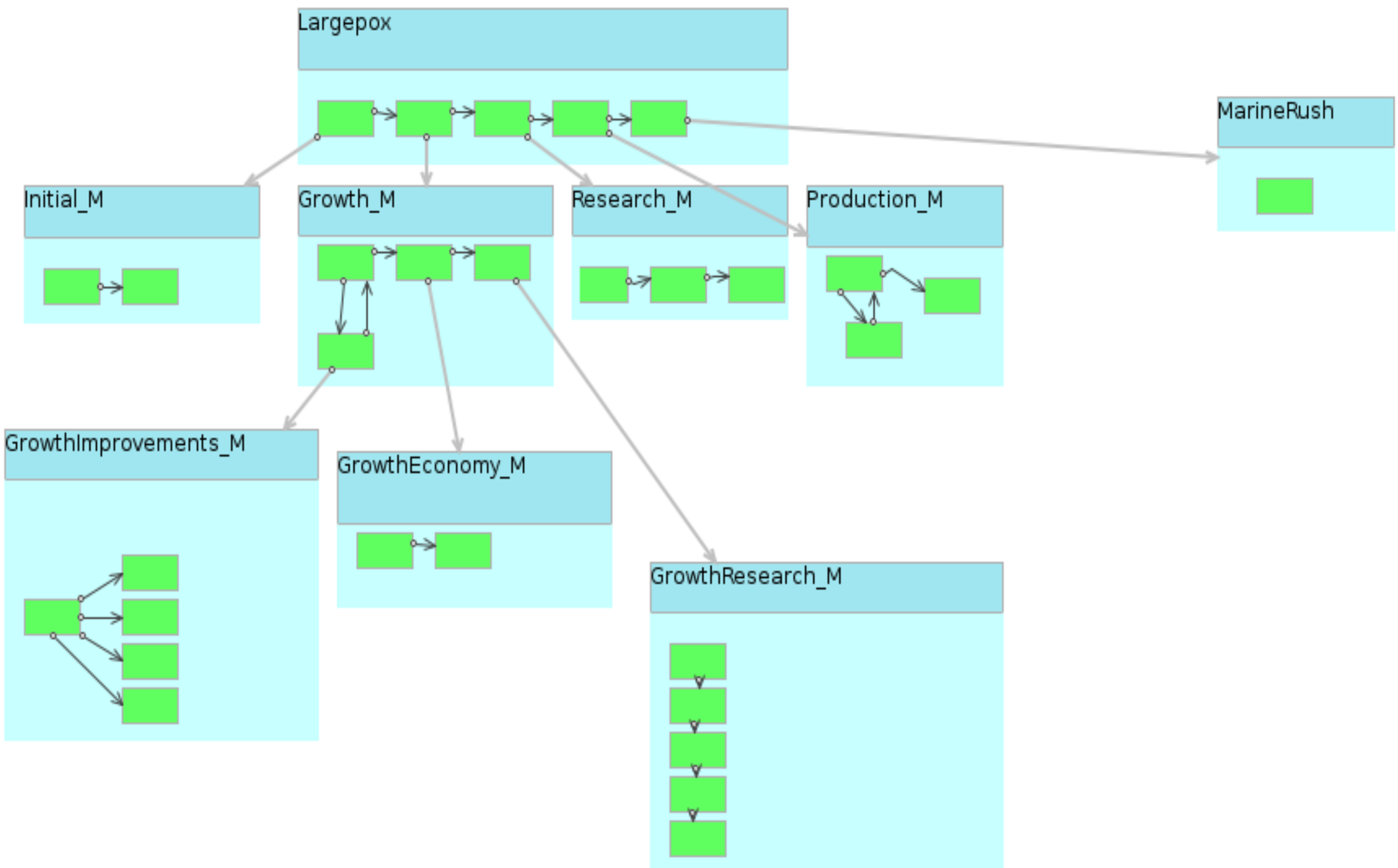
- Complexity of the agent
  - Program code
  - Many components and connections
  - Many, many paths through these elements and connections
  - Local change can have non-local effects

REM Hypothesis

- Specify the functions, mechanisms, and composition of the agent's design.
- Tasks express the functions the agent wants to accomplish.
- Methods express the mechanisms for achieving a goal.
- TMKL language. (Fensel & Benjamin's UPSML; also HTN – Munoz-Avila)

(Murdock & Goel 2008)

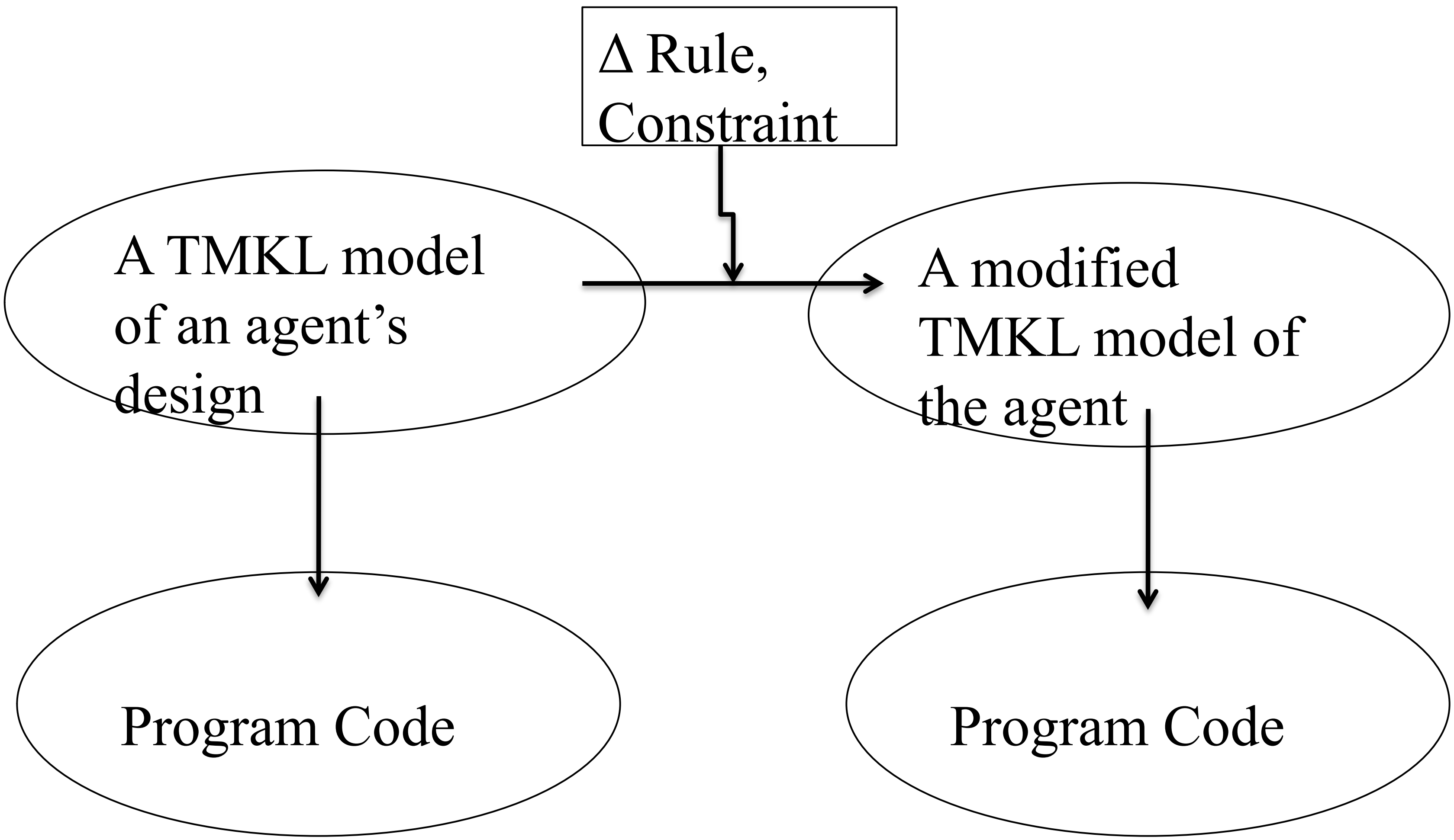
A TMKL Model of a “LargePox” Freeciv Agent



Freeciv Ontology

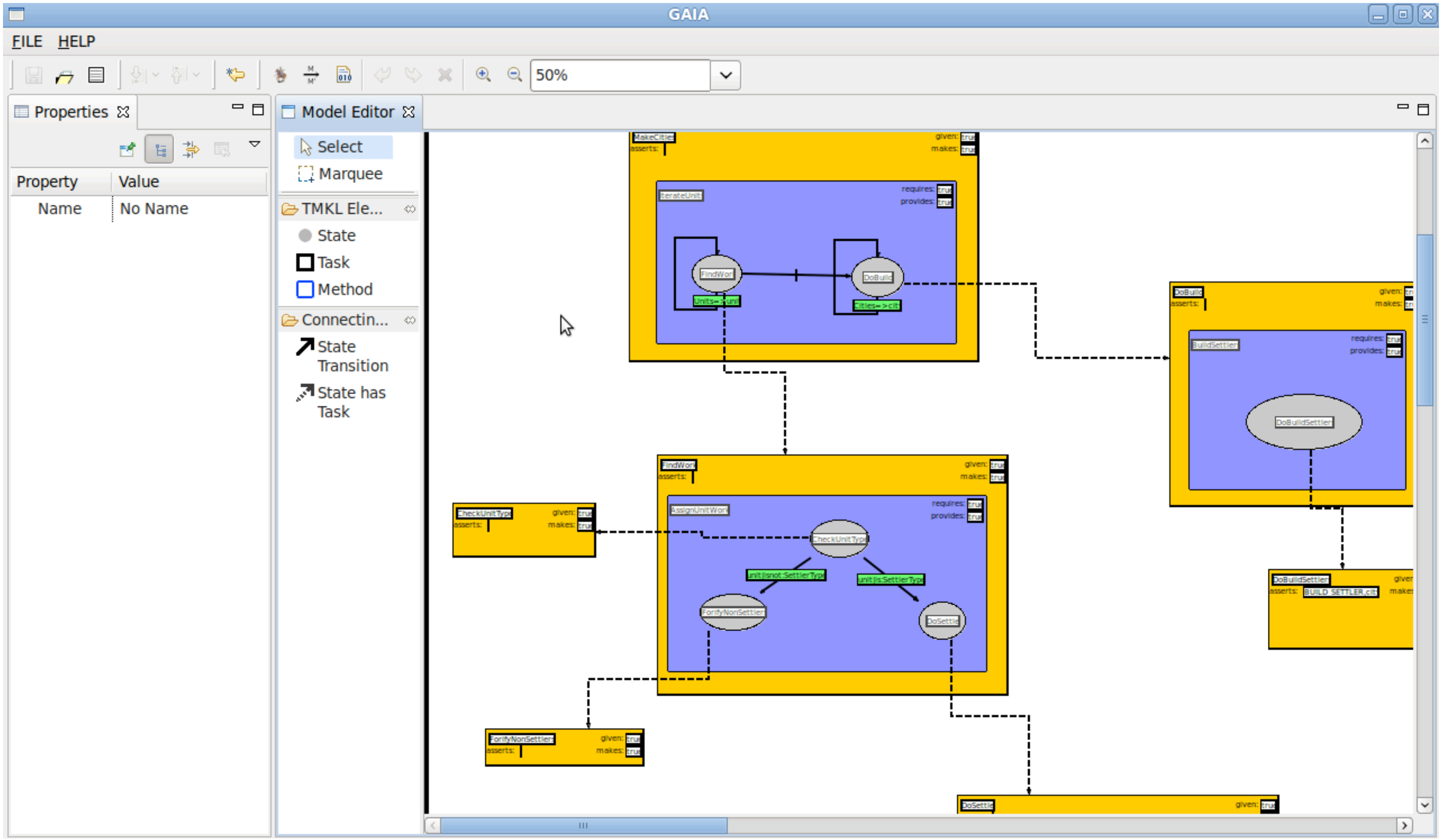
- We developed a Freeciv ontology (concepts, relations, classes, instances) in OWL using Protege.
- Then, in TMKL that uses FOL.
- We want to develop a Freeciv agent's ontology of goals and methods.

The GAIA Project

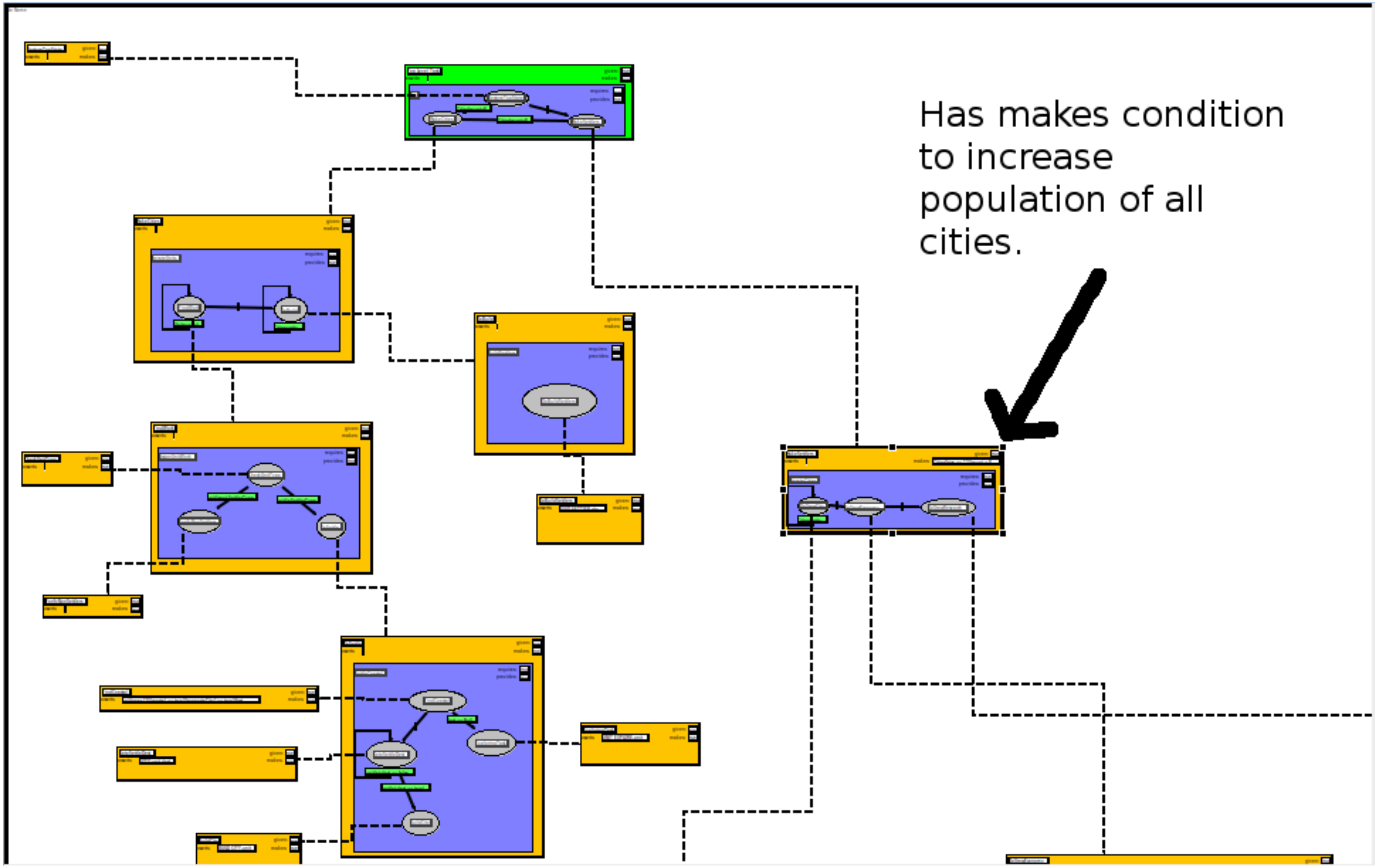


Three broad stances towards for adaptation:
(1) Interactive.
(2) Model-Based Meta-Reasoning.
(3) Meta-Reasoning + Generative Planning or Reinforcement Learning.

Opened Agent Model

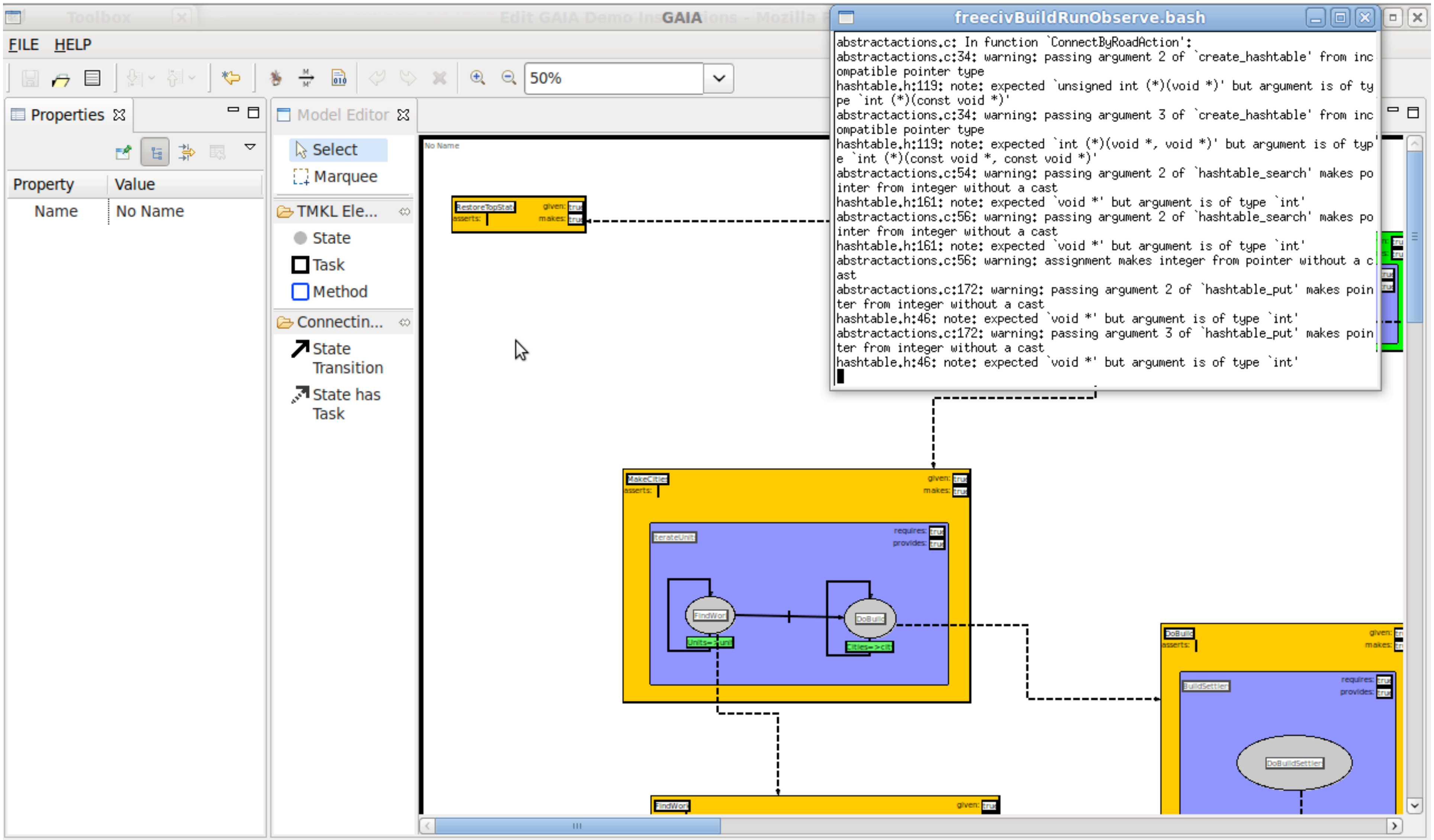


Opened Agent Model (zoomed out)

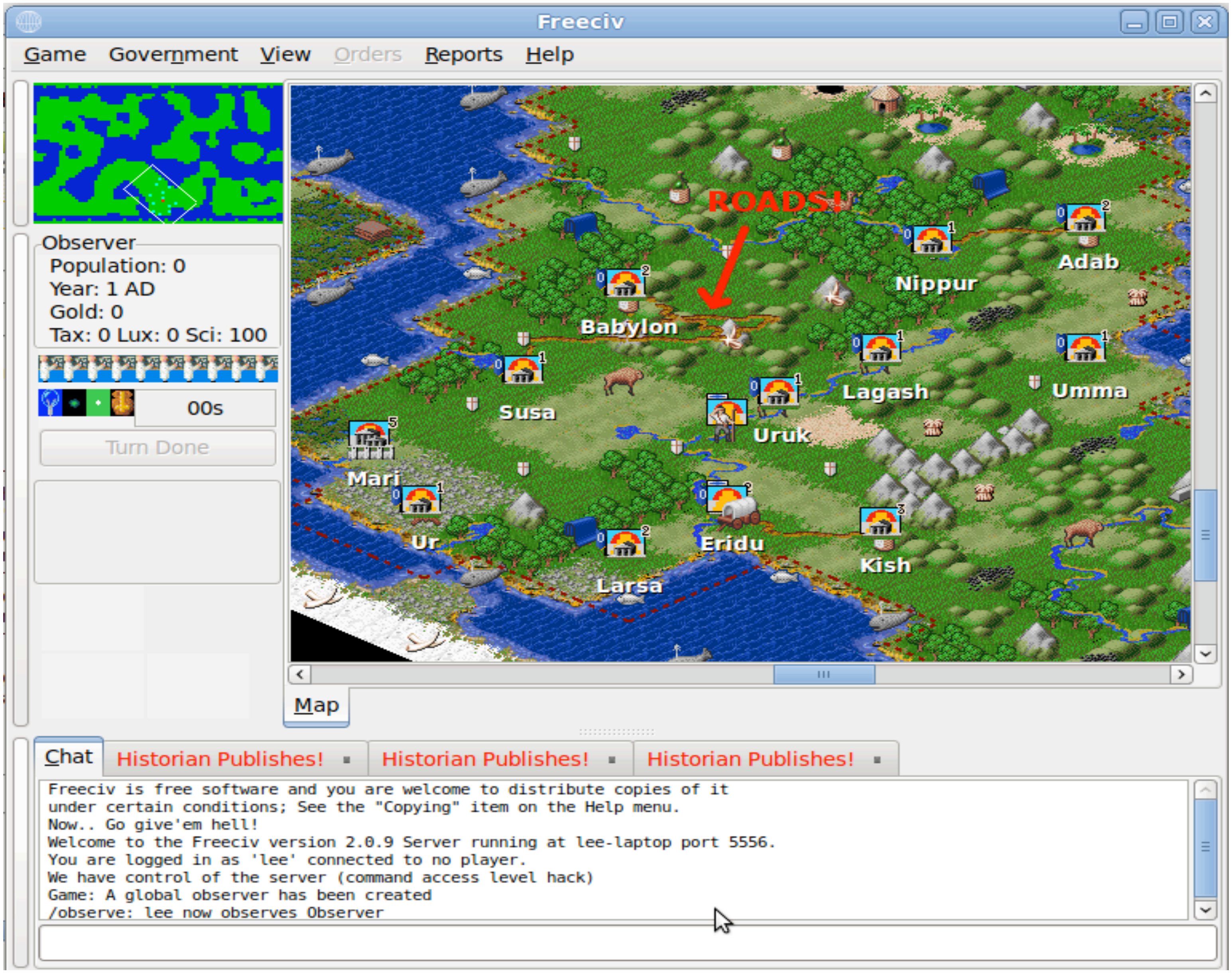


Has makes condition to increase population of all cities.

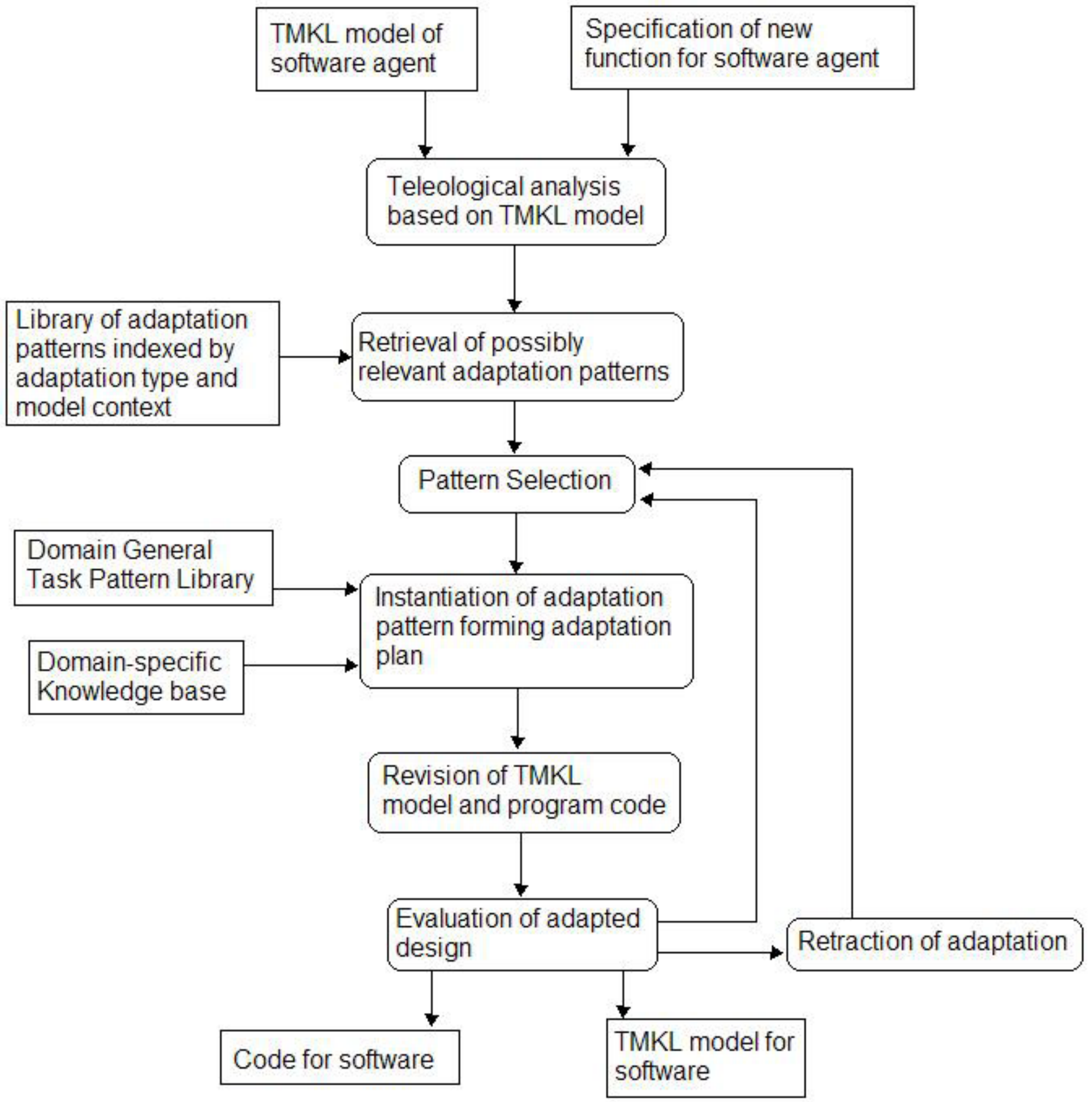
Compiling Agent Model



Simulating and Observing Adapted Agent



The GAIA Adaptation Process
(based on REM architecture)

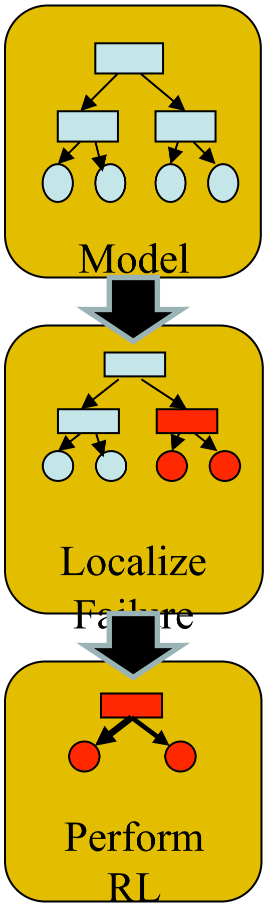


This works but …

This works for our adaptation scenario #2 but
Requires a *lot* of knowledge engineering.

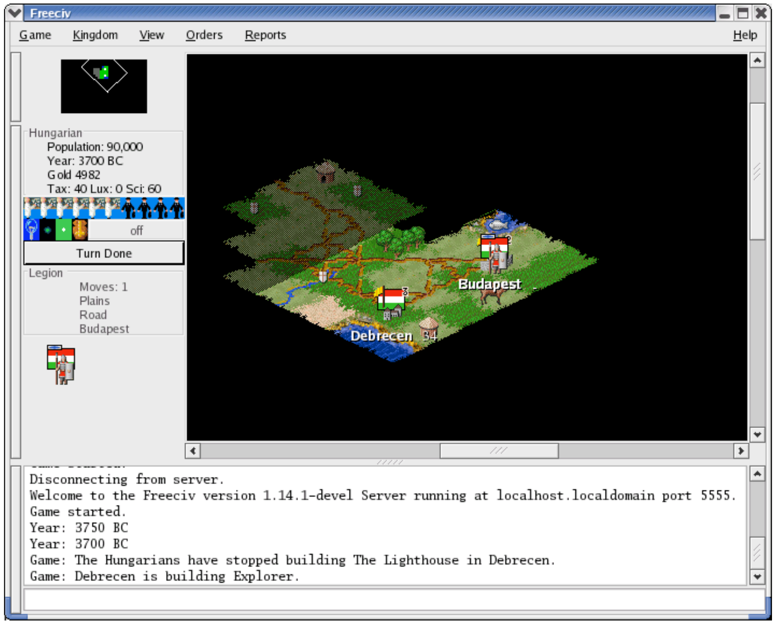
Using Model-Based Meta-Reasoning for Localizing Reinforcement Learning

- Endow the agent with a TMK model of its own design.
- Upon failure, use model-based meta-reasoning to localize the failure to a specific element of task execution
- Use reinforcement learning (RL) to adapt agent's reasoning at identified location



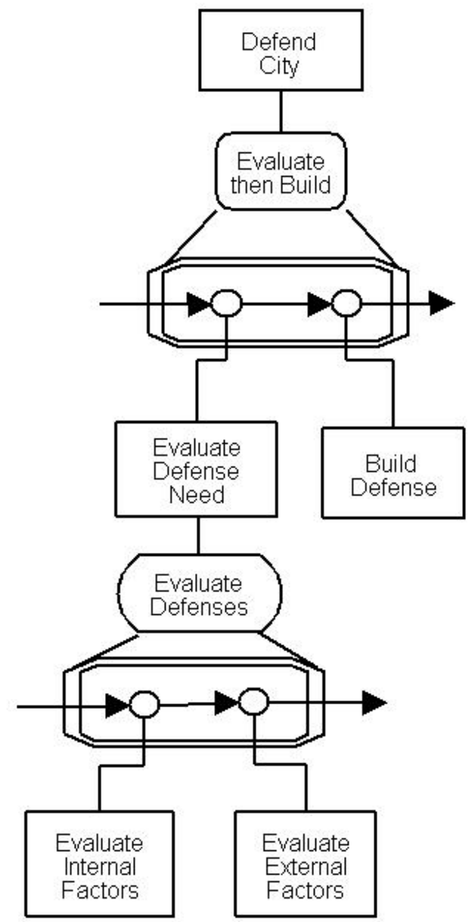
Defend The City Task

- Goal
  - Defend the starting city from enemy civilizations for as long as possible
- Possible actions
  - Building the unit with highest defensive value
  - Producing wealth
- Success conditions
  - Survive 100 turns
- Failure conditions
  - City is defeated
  - City revolts



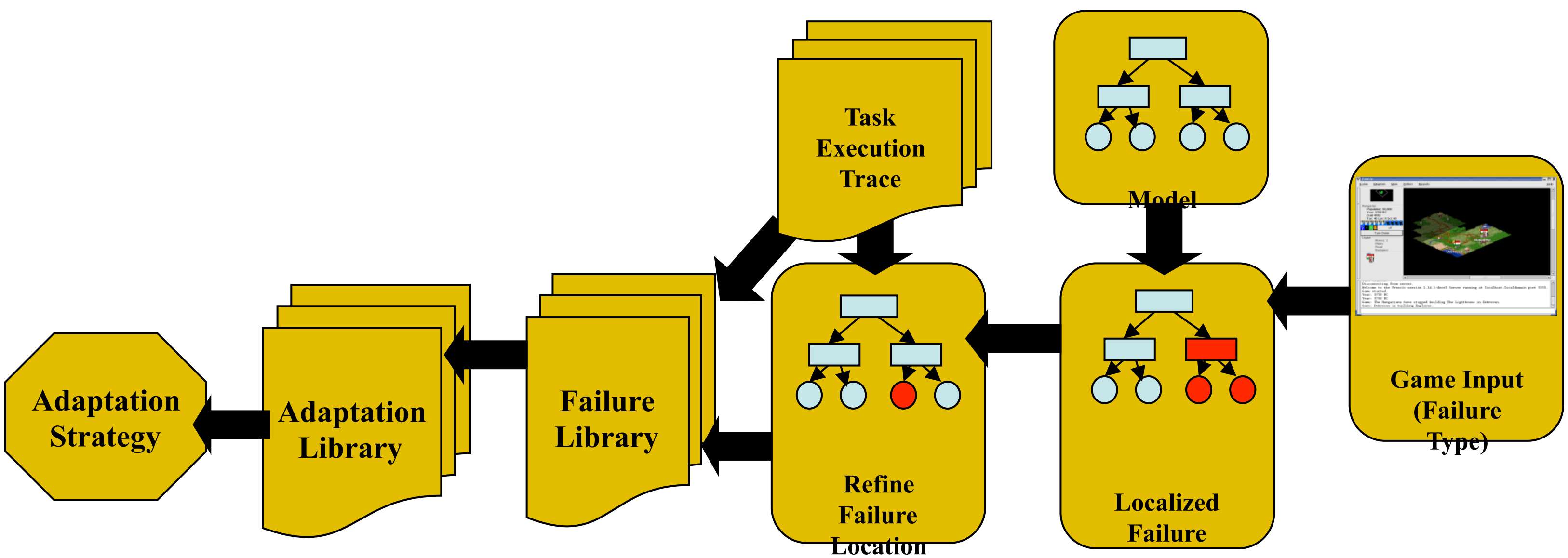
Defend the city model

- Tasks
  - Defend City
  - Evaluate Defense Needs
  - Build Defenses (Procedure)
  - Evaluate Internal Factors (Procedure)
  - Evaluate External Factors (Procedure)
- Methods
  - Evaluate and Build
  - Evaluate Defenses
- Knowledge
  - Different for each task
  - E.g. evaluate external factors produces knowledge about the number of enemy units nearby



Adapting the Defend City Task by Model-Based Meta-Reasoning

- Upon task failure, failure type used to localize failure within model
- Execution trace used to further narrow space of possible failures locations
- Adaptation for each type of failure provided via user-suppplied adaptation library
- Adaptations consist of small changes to leaf nodes in the TMKL model,

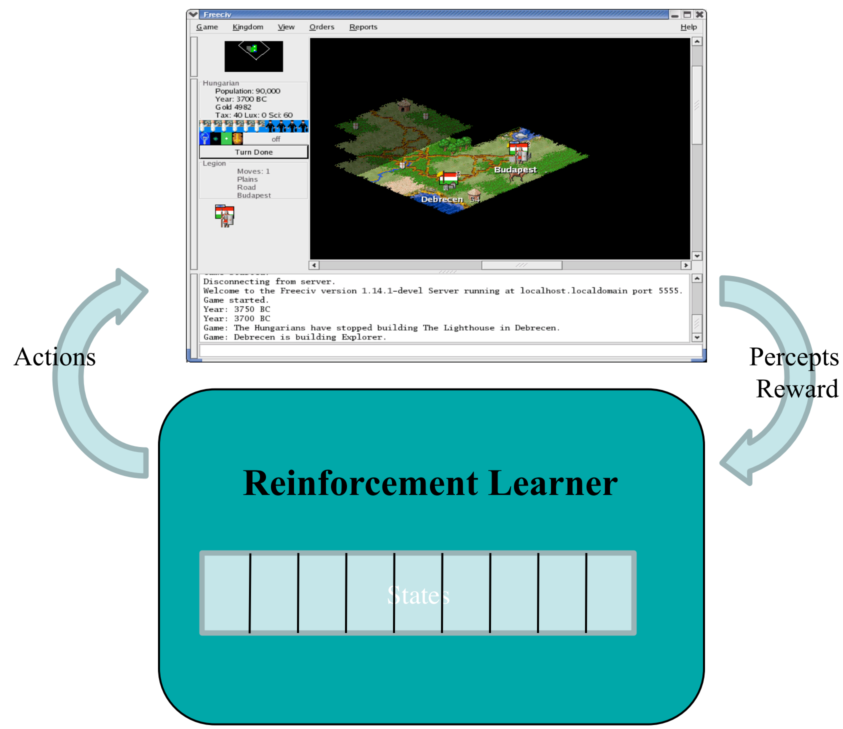


Adapting the Defend City Task by Reinforcement Learning

- State space consisting of 9 binary variables
  - E.g. Are there are less then X enemy units near the city?
  - E.g. Are there are less then Y defensive units currently stationed at the city

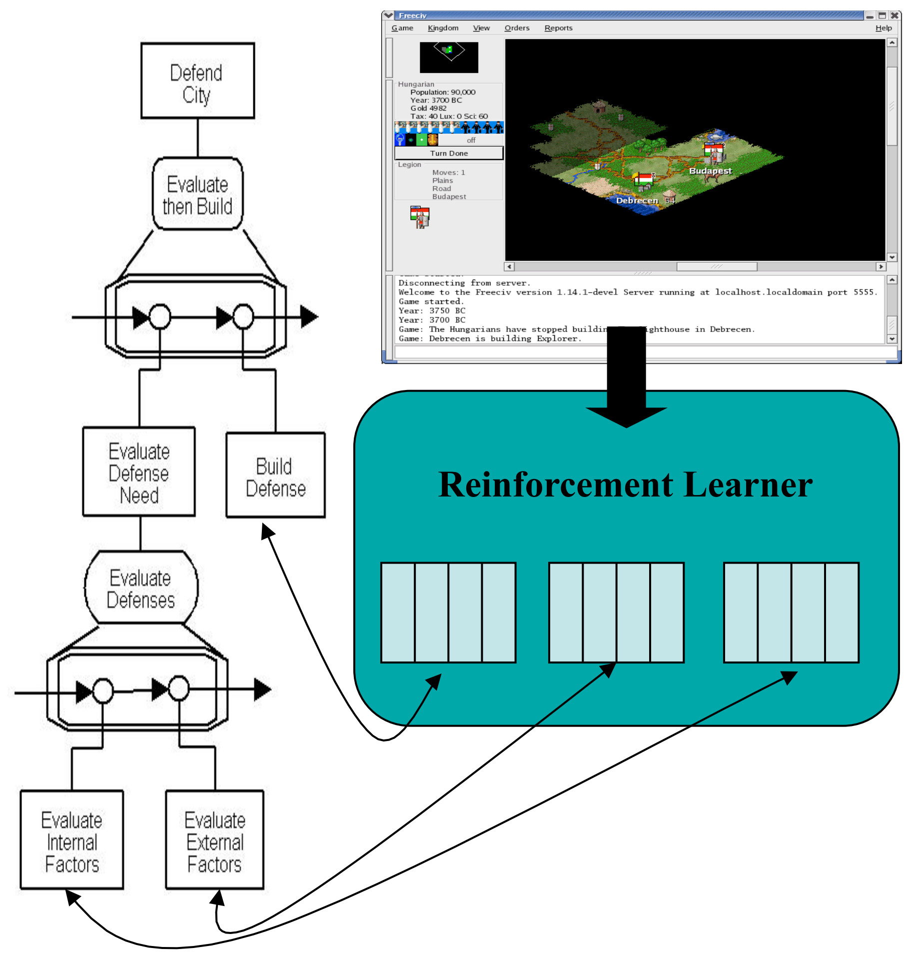
- Two actions
  - Build defensive unit
  - Build wealth

- Negative reward signal received upon failure



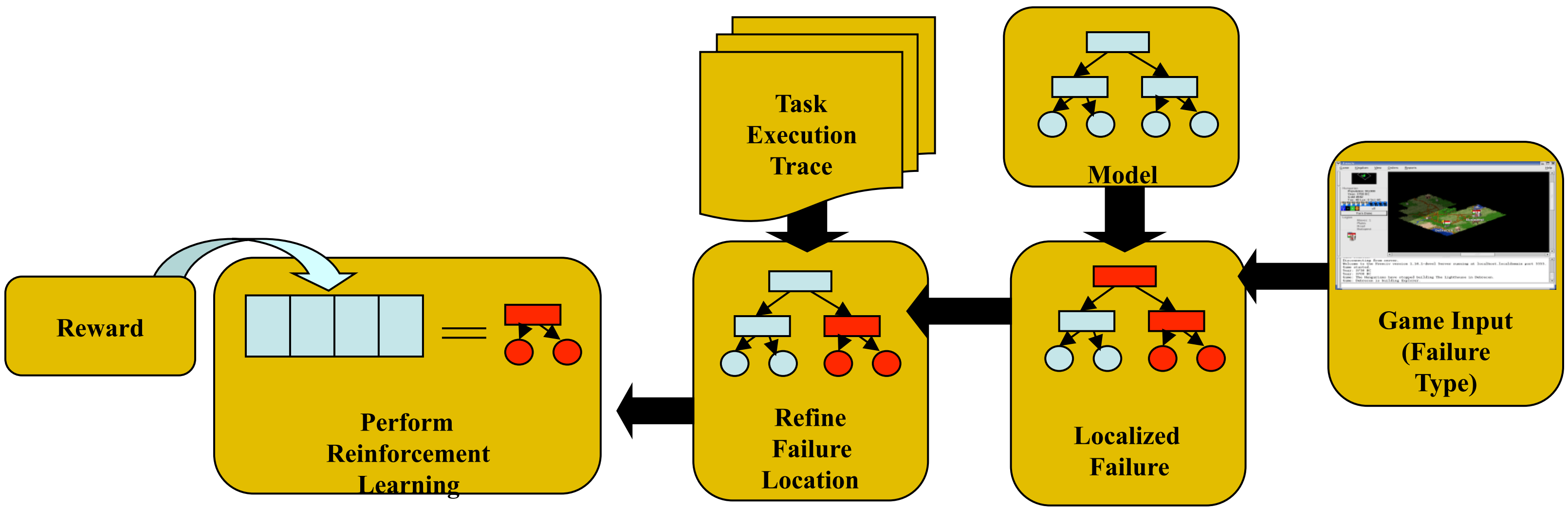
Combining Model-Based Meta-Reasoning and Reinforcement Learning

- Use model as guide for divide state space
- Associate each subdivision with specific portion of model
- Each small reinforcement learner can receive separate reward signal



Adaptation in the Hybrid Technique

- Upon task failure, failure type used to localize failure within model
- Execution trace used to narrow space of possible failures locations
- Only portions of model identified receive reward signal



Ulam, Jones & Goel 2008

Evaluation

- Experimental Setup
  - Performed 100 trials of 100 turns for each agent on smallest map setting
  - Each trial for each agent shared same map
  - 8 Built-in AI opponents at hardest difficulty
  - Agent limited to a single city

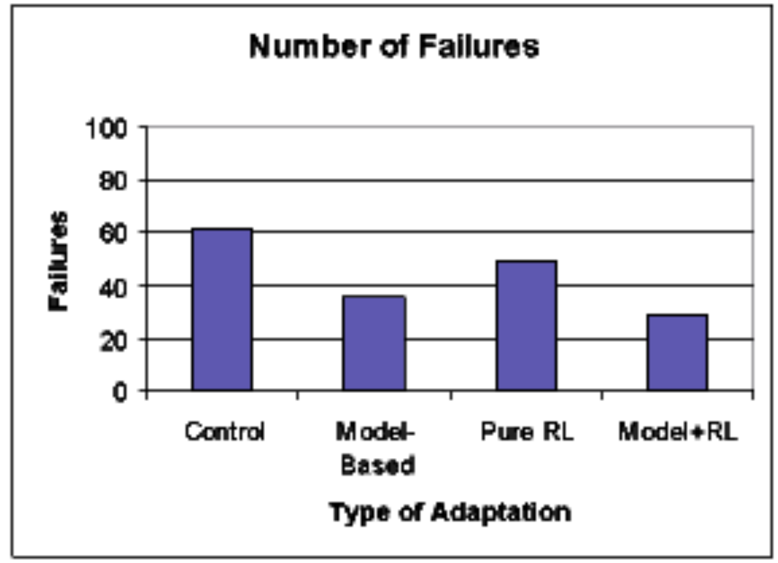
- Evaluation Metrics
  - Number of failures
  - Mean time between failures
  - Number of attacks successfully defended

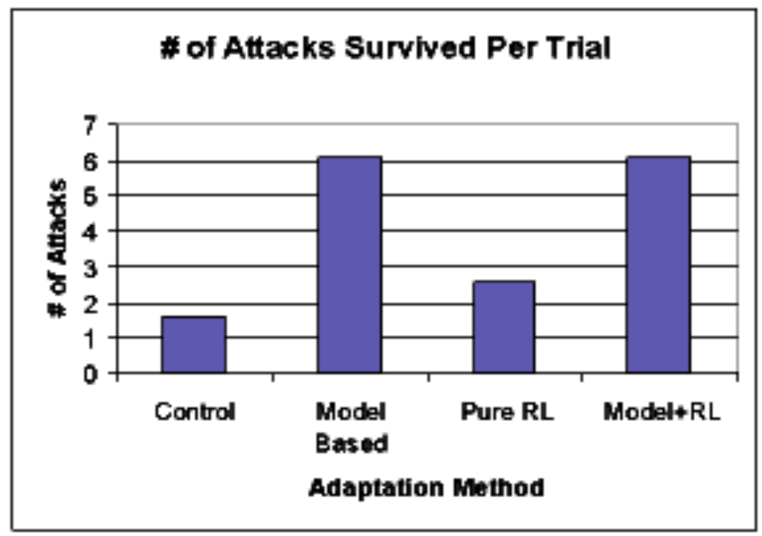
Experimental Agents

- Control
  - Agent attempts to maintain 1 defensive unit, no adaptation
- Pure model-based reflection agent
  - Starts from control
  - Adapts via user defined adaptation library
- Pure reinforcement learning agent
  - Initialized to always build wealth
  - Q-Learning
- Hybrid agent
  - Initialized to always build wealth

Experimental Results

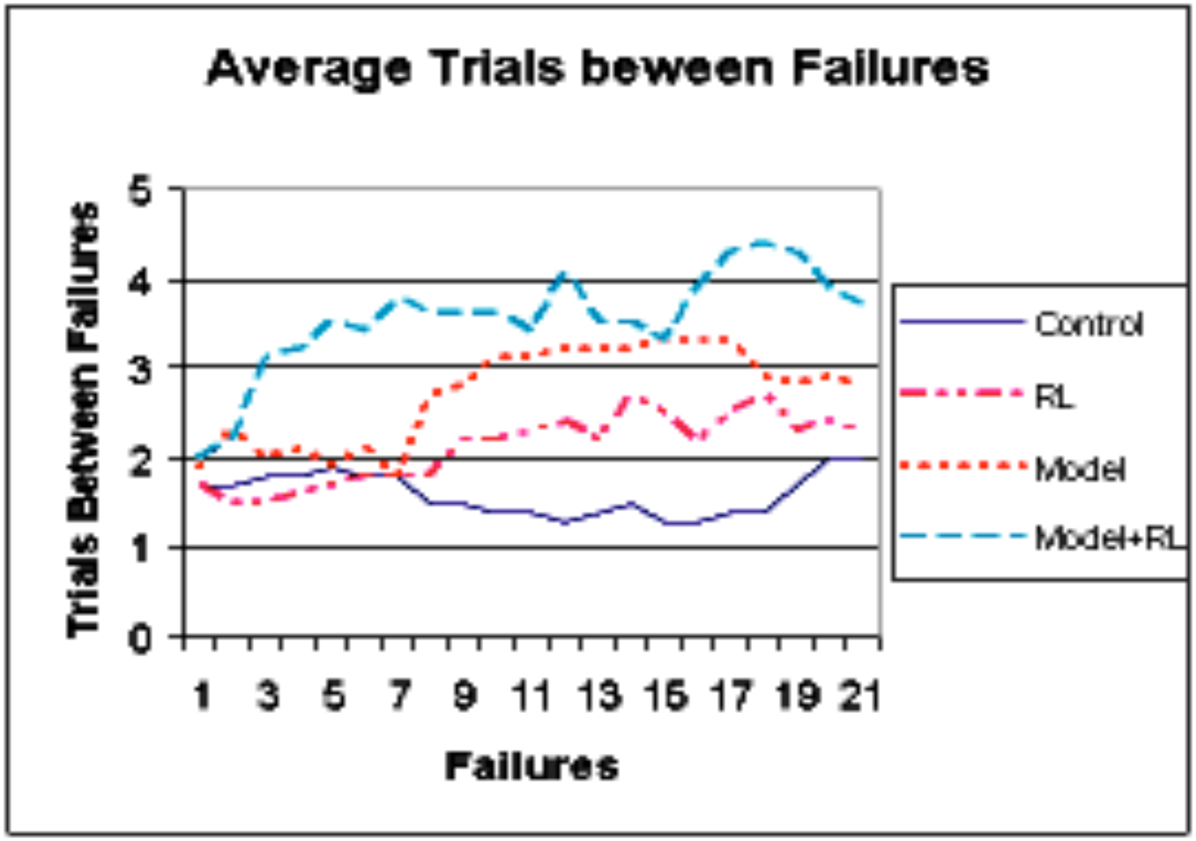
- Number of trials failed directly measures success of agent
  - Less failures indicates better performance
- Number of attacks per trial
  - More attacks survived indicates higher performance
- Hybrid model-based/RL method combines low failure rate with good survival rate.





Experimental Results

- Average time between failures
  - Assumes the better the agent learns the task, the longer the period between failures
  - Rate of increase indicator of speed of learning



Summary

- Constraints, resources pertinent to an agent's goals in the world constantly evolve (but not necessarily goals themselves).
- Proactive, goal-directed adaptation of game playing software agents.
- Design stance: Teleology is the fundamental organizing principle of agent design adaptation.
- Meta-reasoning is effective for some problems but … requires extensive knowledge engineering.
- Combine with generative planning and situated learning: meta-reasoning for localization.

Acknowledgements

Spencer Rugaber

US National Science Foundation (Science of Design)

Design Stance

- What knowledge of the design of an intelligent agent may help make adaptations efficient and accurate?
- How may we endow an agent with knowledge of its own design?
- How might an agent use this self-knowledge to adapt itself?
- How may we design intelligent agents so that they can be adapted efficiently and accurately?