

# **MAHCA Technology**

**Wolf Kohn**

**Hybrid Systems are models for networks of digital and continuous devices in which digital control programs sense and supervise continuous and discrete systems governed by differential or difference equations. Hybrid Systems provide the basic framework and methodology for the synthesis and analysis of autonomous and intelligent systems. Modern industrial society is filled with hybrid systems used for such varied purposes as aircraft control, computer synchronization, manufacturing, communication networks, traffic control, industrial process control, and enterprise management and control.**

**Hybrid Systems theory provides the backbone for the formulation and implementation of learning control policies. In such policies, the control acquires knowledge (discrete data) to improve the behavior of the system as it evolves in time.**

**Hybrid Systems has become a distinctive area of study due to opportunities to improve on traditional control and estimation technologies by providing computationally effective methodologies for the implementation of digital programs that design or modify the control law in response to sensor detected events, or as a result of learning.**

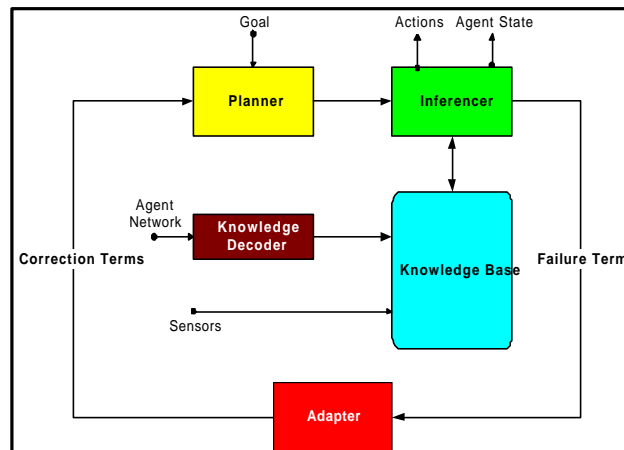
## *Hybrid Systems*

Hybrid Systems is a new control system and computation discipline which has emerged in the last five years from control theory, engineering, computer science, and mathematics. The objects of study in hybrid systems are processes and systems, which exhibit a mixture of evolution (continuous time) models and logical control and constraints. Wolf Kohn P.H.D. and Professor Anil Nerode of Cornell have developed a general theory for the automatic extraction of controls to ensure that a hybrid system meets user-defined requirements. The Kohn-Nerode theory is based on results and techniques from a variety of disciplines including optimal control, differential geometry, differential equations, automata theory, logic programming and data structures.

Hybrid Systems is an active area of research and development in both industry and universities. There are a number of well-attended conferences, workshops, and special sessions at both Computer Science and IEEE conferences each year.

## **Hynomics Technology and the Multiple-Agent Hybrid Control Architecture (MAHCA)**

Hynomics Corporation has developed implementations of the Kohn-Nerode extraction procedure via agent-based programming. In particular, Hynomics has developed a Multiple Agent Hybrid Control Architecture (MAHCA) which is ideal for the control and synchronization of distributed systems, which involve both logical, and evolution models.



*Figure 1. MAHCA Agent*

*Networks of Digital Signal Processors are an ideal hardware framework for the implementation of the MAHCA architecture. See Figure 1.*

Hynomics multiple-agent, declarative, hybrid-controller approach is a new technology with broad application since it allows for the integration of logic (set-based) and evolution (continuous-time) models of complex system behavior.

The Kohn-Nerode approach for control of hybrid systems emphasizes on-line generation of code in the format of automata, which meet real-time and logical constraints. The approach has linear scalability.

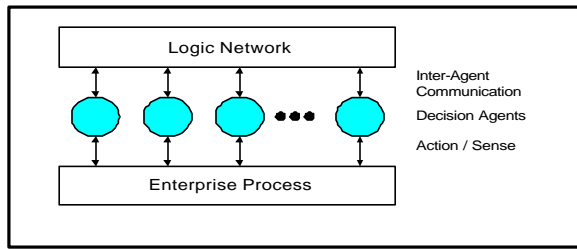


Figure 2. Logic Agent Network

Each MAHCA agent operates as a real-time theorem prover in the domain of relaxed variational theory. A *customized* version of this theory, enriched with elements of differential geometry and equational logic provides a general representation for the dynamics, constraints, requirements and logic of complex computer-controlled systems. This representation is encodable in general-purpose processors and also in DSPs.

The architecture is composed of two items: the MAHCA Agents, and the Logic Agent Network. These items are illustrated in Figures 1 and 2 respectively.

## Architectural Elements of a Control Agent

A control agent is composed of five functional elements - a Planner, a Knowledge Base, an Inferencer, a Knowledge Decoder and an Adapter.

### Planner

The Planner generates a plan of action that represents the desired behavior of the process to achieve the agent goal as viewed by the agent. This plan is expressed as an optimization criterion: the behavior criterion.

For a permanent agent, the planner is encoded from a *global criterion goal* specification.

### Knowledge Base

The Knowledge Base is dynamic. It is an updateable model representing the current knowledge the agent has about the process. Knowledge is encoded in a collection of rules called Equational-Relational Clauses.

### Inferencer

The Inferencer is the intelligence element of an MAHCA agent. The Inferencer implements a declarative, deductive paradigm to infer, on-line, valid assignments of values to the variables in the plan generated by the Planner. An inferred assignment is valid if it satisfies the clauses in the Knowledge Base. If there is no valid assignment, the Inferencer generates failure terms. These terms are fed to the Adapter for correction or modification. Part of the assignments computed by the Inferencer are the actions executed by the agent. The inferred actions remain in effect until the Inferencer determines that the plan generated by the Planner is not feasible.

The Inferencer uses inference operators. These operators are applied in succession to obtain an executable procedure called Inference Automaton that, when executed, generates either the actions and agent state update or the failure terms and the state update. This on-line code generation is unique to MAHCA's agents, and is chiefly responsible for their versatility, robustness, and learning capabilities.

## Adapter

The Adapter implements generic transformations on failure terms to convert them into correction terms to the plan. The Adapter executes this function by applying composites of correction operators on the failure terms.

## Knowledge Decoder

The Knowledge Decoder translates knowledge data from the network into the agent's Knowledge Base by updating the inter-agent specification clauses. These clauses express the constraints imposed on each agent by the rest of the network.

## Network

The inter-agent communication network's main function is to transfer inter-agent synchronization data among agents. These data include application dependent information.

Hynomics' implementation of an MAHCA network also allows a network to create or eliminate agents depending on system requirements. See Figure 3.

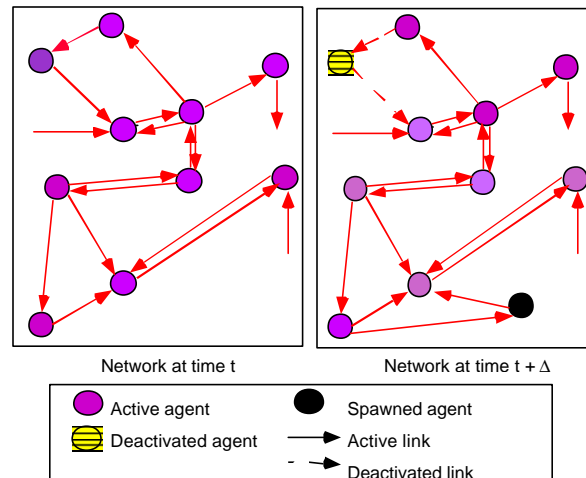


Figure 3. Spawning and Deactivation of Network Agents

The advantages and features of the MAHCA architecture include the following.

- **Reactive** -- The theorem proving function of each agent operates according to a first principles feedback paradigm.
- **Adaptive** -- The knowledge base of each agent is open and modifiable by sensory data. Theorem failure triggers tuning and corrective action.
- **Distributive with Coordination** -- The theorem proving is carried out distributively over the agents. The coordination scheme is implicit without umpire.
- **Dynamic Hierarchization** -- The architecture can operate simultaneously at different levels of abstraction.
- **Figure of Merit** -- The behavior of the closed loop distributive system is determined by proving that there exists a command trajectory that minimizes a goal depending on the state of the process.
- **Real-Time** -- Constraints for real-time performance are explicit and part of the knowledge base. This is important because real-time constraints cannot be fully instantiated at design time.

MAHCA agents will allow for control and synchronization of a variety of real-time processes. See Figure 4.

## ***MAHCA Control of a Distributed Process***

MAHCA agents provide unique features for the intelligent control of distributed processes. A dynamic process that is controlled by more than one decision maker is a distributed process. Figure 4 illustrates the performance enhancements introduced by MAHCA agents affecting the behavior of a distributed process. Specifically, these are robustness, adaptation, intelligence and real-time synchronization.

The agent mechanisms responsible for these enhancements are feedback control, reactive inference, learning and real-time synchronization.

### **Feedback control**

Feedback is the information causally extracted by sensors, which estimates the state of the distributed process. Feedback control is the action taken based on feedback data.

The MAHCA agent architecture has two feedback loops: a control loop and a reactive learning loop. See Figure 5. The control loop maintains the performance of the process, as viewed by the agent, which satisfies the current active plan.

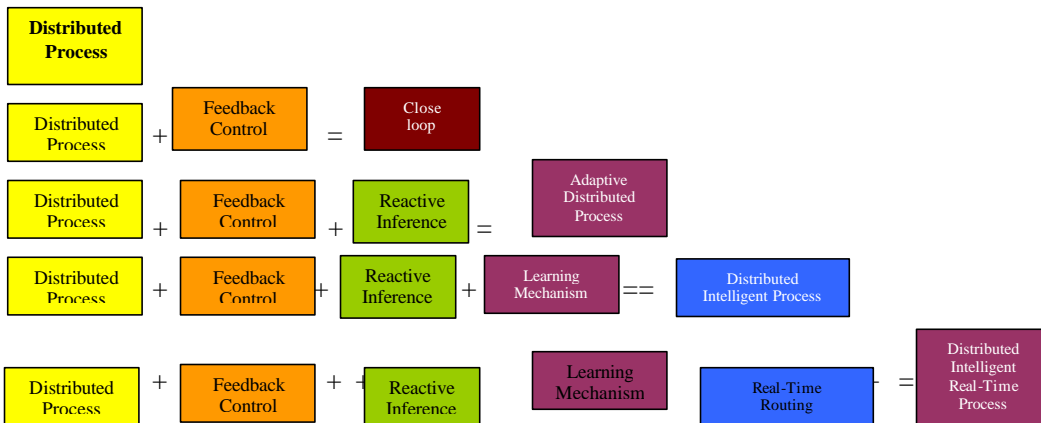
The reactive learning loop monitors the compliance of the plan by the actions generated by the inferencer consistent with the current status of the knowledge base. If conflict arises between the model in the knowledge base and the current plan, the reactive learning loop directs the construction correction terms to the plan resulting in a new plan. This feature is unique to MAHCA agents.

### **Reactive inference**

Reactivity is the dependency of inferred information as a function of current knowledge. Reactive inference is extraction of action information from the current knowledge. The inferencer of an MAHCA agent uses a symbolic procedure to compute optimizing actions.

### **Learning**

Learning is the improvement in process performance due to the availability of more refined actions. If the inferencer determines that there is no solution to the optimization problem encoding the plan, it generates diagnostic elements, which are used to construct the correction terms.



*Figure 4. MAHCA performance features*

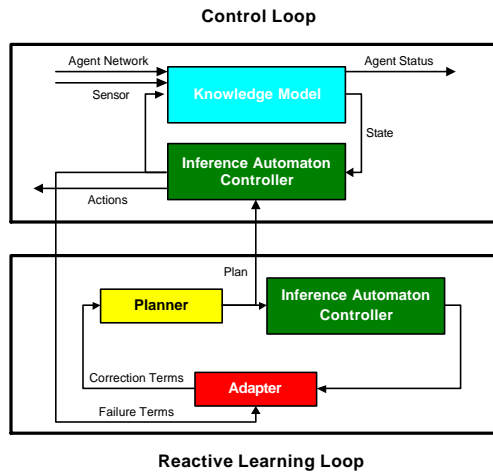


Figure 5. Agent feedback Loop

## Reactive inference

Reactivity is the dependency of inferred information as a function of current knowledge. Reactive inference is extraction of action information from the current knowledge.

The inferencer of an MAHCA agent uses a symbolic procedure to compute optimizing actions.

## Learning

Learning is the improvement in process performance due to the availability of more refined actions. If the inferencer determines that there is no solution to the optimization problem encoding the plan, it generates diagnostic elements, which are used to construct the correction terms.

### Real-Time synchronization

Real-Time synchronization is the encoding of constraints, which force the generation of compatible agent action trajectories.

An MAHCA agent is synchronized to its peers if a rule in its knowledge base, instantiates a value equal to the invariant parameter of the agent. If the value of the invariant parameter changes during the inference process, then the agent has become unsynchronized with its peers and the solution of the optimization problem fails resulting in a plan adaptation cycle.

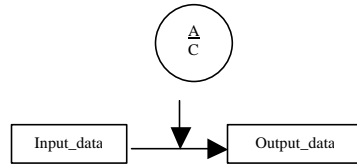
## Agent Programming and Construction

### Agent Programmability

The steps in programming an agent are:

1. Build a Knowledge Model encoding a collection of rules describing the process as viewed by the agent. In addition to these rules, the knowledge base includes a set of system rules associated with the operation of the inferencer, in particular, the rules that direct the construction, execution and diagnosis of automata. The programming is also responsible for classifying the model rules into two categories: adaptable rules and non-adaptable rules. Non-adaptable rules cannot be changed as a result of an inference failure. Adaptable rules can be modified by the inferencer to repair a failed plan. The system rules are non-adaptable rules.

2. Build a criterion that expresses desired behavior, which provide the planner with the knowledge to be used to generate a plan which is then used to drive the process under control towards the goal.
3. Build a goal that expresses desired performance. This step requires the specification, of a goal, or class of goals, which the agent is directed to achieve. This programming step is carried out using a step called the 'agent builder.' A block diagram of the agent builder is shown in Figure 6.



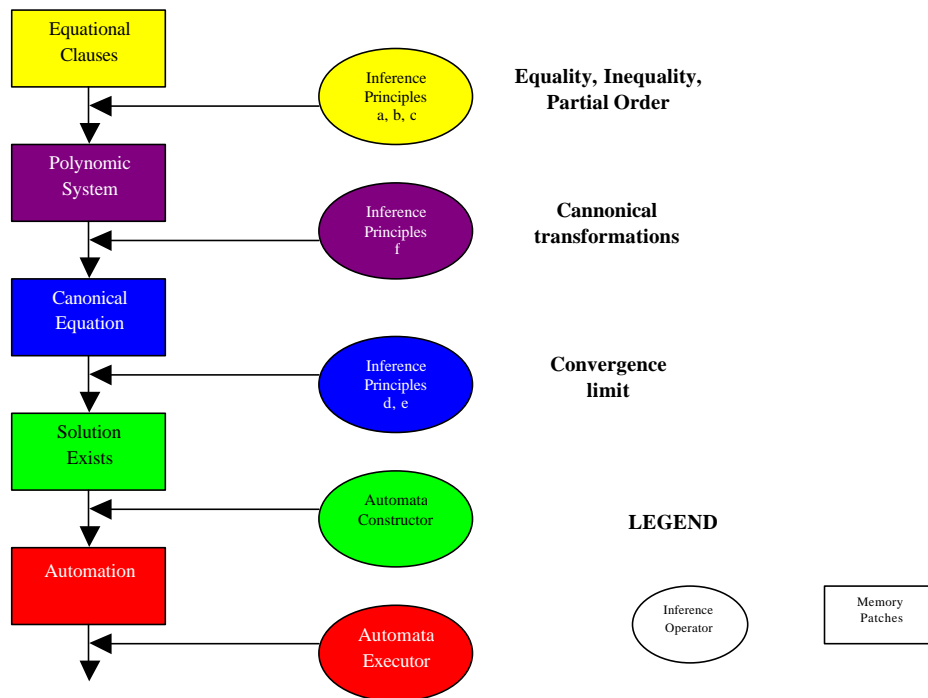
*Figure 7. Basic programming step*

**Programming Schema**

Each module of an agent is encoded using primitive elements called inference operators. An inference operator is in the form of:

Antecedent (A) / Consequent (C)

Where A and C are rules. A procedural step in an agent module is a triplet of the form shown in Figure 7. A procedure is a collection of these procedural steps organized in a directed graph where the nodes are steps and the edges are labeled by inference operators. For example, Figure 6 shows a simplified procedure for executing the inferencer function.



*Figure 6. Block diagram of the agent builder*

## **About the Company**

Hynomics Corporation was founded in February of 1995 to develop hybrid systems and agent based programming applications. The founders of the company are Wolf Kohn and Anil Nerode. The core technology of the company was developed over a number of years, first by Wolf Kohn and later by Kohn and Nerode with contributions from Jeff Remmel. Hynomics has two patents on this technology. Much of the research was sponsored by various Department of Defense (DOD) contracts and NIST grants. In addition to its high level internal technical staff, Hynomics has strong research ties to U.C., Berkeley, Cornell, and U.C. San Diego.