Piagetian Autonomous Modeler

Michael S. P. Miller
Los Angeles, California
piagetmodeler@hotmail.com

Abstract. The Piagetian Autonomous Modeler (PAM) is a proposed architecture for a system that constructs an internal representation of a real or simulated environment based on its interaction with the environment. The novel aspects of PAM are: (1) how it spreads activation; (2) its use of two kinds of schemata (structural and behavioral) to connect the representational units (monads); (3) its use of multi-strategy inference to extend the internal model; (4) its use of a consolidation component to provide automaticity and forgetting; and (5) its evolution of successful behaviors through genetic techniques. The system is called Piagetian because it employs the notion of Monads (fundamental representational units), Schemata (patterns of structure and behavior), Assimilation (incorporating external elements) and Accommodation (modifying internal structures in accordance with environmental feedback) which are essential to the theories of Human Cognitive Development espoused by Piaget [7] [8].

1 BACKGROUND

The work in "*early developmental AI*" as surveyed by Guerin [17] is replete with examples of artificial intelligence computer programs that can interact with an environment, learn, and synthesize new concepts. Most prominent among them is Gary Drescher's seminal program, the Schema Mechanism [1], which employed the theories of Jean Piaget to demonstrate aspects of learning and planning in infant cognitive development.

The PAM architecture inter-connects and advances the work of earlier system architects such as Drescher [1], Heib & Michalski [2], Tecuci & Michalski [3], Holland et al. [4], Goldberg [5], Riesbeck & Schank [6], Chaput [14], and others.

This architecture is compatible with the developmental theory and embodied-social cognition theory of language learning as described by Kaplan, Oudeyer, and Bergen [22].

Although embodiment (sensing and acting upon the environment) is central to the PAM system, this work deliberately does not address attention, curiosity, motivation, drives, beliefs, desires or intentions. This omission was made in order to limit architectural concerns in the initial design of the system. These phenomena may be revisited in later phases as the PAM system evolves.

2 RESEARCH GOALS

The PAM effort is a multi-phased inquiry into early developmental AI which has several objectives:

(1) to replicate Piaget's sensorimotor and pre-operational stages of cognitive development including language acquisition;
(2) to create smarter computer systems based on Piaget's genetic epistemology that (a) are capable of modeling their environment, (b) exhibit stages of development, (c) predict transformations in their environments, (d) learn from failure, and (e) perform multistrategy inference;
(3) to explore the validity of the hypothesis that monads and schemata can be used to model a learner's environment;

and (4) to unify the work of Drescher and Michalski.

3 ARCHITECTURE

The PAM architecture described herein represents the first phase[1] of the research effort.

3.1. Assumptions

The system assumptions for PAM are:

(1) Human learners construct a mental model to represent (a) the structure of and (b) transformations within their environment.
(2) Monads and schemata are sufficient to construct a predictive model of an environment.
(3) The PAM system is implementable on existing computing technology.
(4) The system performs in real time, is resilient, available, and scalable.
(5) The system is domain agnostic. Any domain specific percept and effect assertions made to the system are irrelevant since all assertions are transformed into an internal representation of monads and schemata. Therefore, only the concurrence and recurrence of the assertions are salient.

3.2. Views

Figure 1 shows PAM interacting with its environment.
Figure 2 depicts the data tiers of the evolving model.
Figure 3 describes the structural and behavioral schemata that PAM employs.
Figure 4 shows a sample inference operation on a portion of the heterarchy.
Figure 5 shows the decomposition of the system elements by process and object.
Figure 6 shows the use cases for each system element.
Figure 7 shows the system elements as components exchanging data.
Figure 8 shows the actual data flows across the elements of the system.

3.3. Monads

A monad is a data structure which represents a percept, effect, or concept. Percepts represent encodings of sensor data from an external environment. Effects represent the status of actions that have been performed on the external environment. Concepts are internally synthesized monads which represent a completely new entity within the model arising from some underlying pattern of structure or behavior. Hence, a concept is

---

[1] Note that language acquisition is a long term goal and will not be addressed in phase 1.

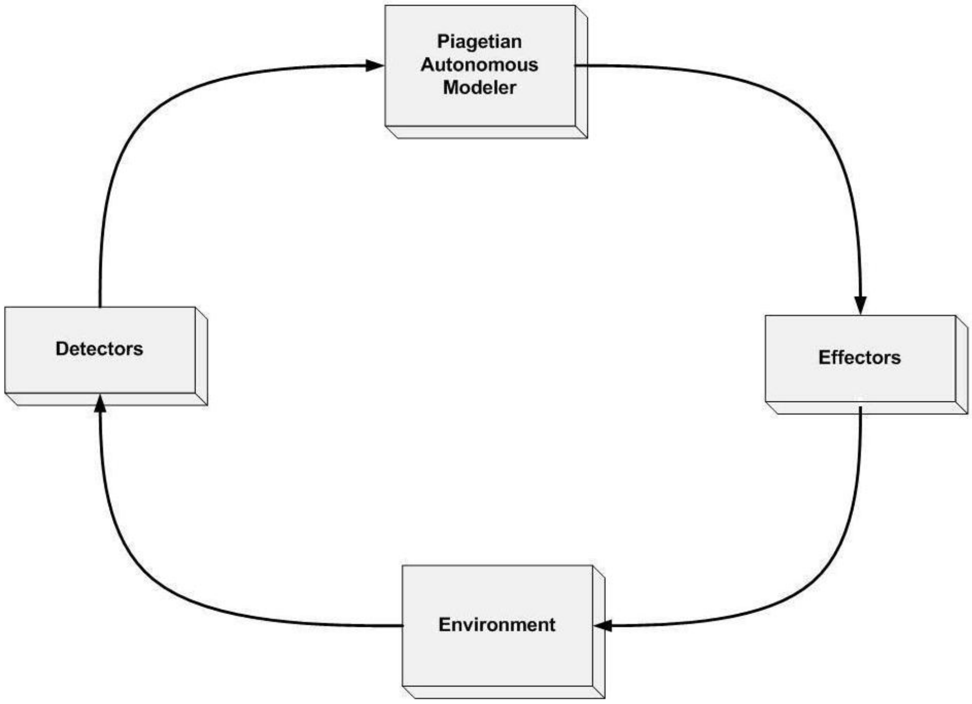

Figure 1. PAM Context view.

a schema. Schemata are structural relationships among monads, or behavioral patterns identifying transformations in the environment.

In Drescher's Schema Mechanism, concepts are called "Items" (which can be in a Boolean state of On or Off). Drescher's "items" correspond to "monads". In PAM monads are not Boolean and hence do not represent a binary On or Off state. Instead, they are continuous and use an activation time which denotes when they were last considered active. This strategy establishes an implicit notion of "decay" which is novel.

Monads actually have two activation times: fact activation and goal activation. These denote when they were last perceived or inferred (as a fact) and when they were last needed to enable a prediction (as a goal). Monads also contain the concept of Tier which sorts them into levels of abstraction and allows them to form hierarchies within the larger heterarchy[2].

3.4. Detectors and Effectors

To use PAM, one or more detector programs and one or more effector programs must be constructed. Each detector program provides PAM with continuous or discrete sensor data transformed into PAM's internal representation, percept monads.

When sense data arrive in the detector program, the program makes assertions in the model (Figure 2). Each assertion either creates a new percept monad or retrieves an existing percept monad, which is then marked active.

Each effector program allows PAM to issue commands to a device and retrieve feedback about the status of the command issued. The status (unknown, pending, executing, failure, or success) is asserted to PAM's model and its corresponding monad is created or retrieved and marked active.

[2] The model heterarchy is the sea of monads akin to Lenat's sea of assertions in Cyc[16].

3.5. Schemata

In contrast to Drescher's Schema Mechanism, which has one type of schema, PAM has two types of schema: structural and behavioral. The two types of schemata are needed because of the system's primary assumptions: that both structure and behavior exist in an environment, and that they are different. Structure pertains to the relationships among entities within the environment, while behavior pertains to the transformations occurring within the environment. Structural schemata are defined in PAM in order to allow PAM to perform inference above and beyond what would be encompassed by behavioral schemata alone because a human (our archetypal learner) can make subtle inferences which go well beyond predicting the effects of actions.

Drescher's schema consisted of a context, action, and result. PAM's schemata differ substantially (see Figure 3).

As behavior, a schema defines a predicate ***then(C, P, s)*** that posits: when the context ***C*** is true, the prediction ***P*** will also be true within a given time span ***s***. Thus, PAM's behavioral schemata contain a context and a prediction. The context contains two lists: enablers and impeders. The prediction also contains two lists: enables and impedes. Monads can be present in these lists within a behavioral schema.

As structure, a schema defines a relation $\boldsymbol{R(a_1..a_3, i)}$ among monad collections $\boldsymbol{a_j}$ in $\boldsymbol{a_1..a_3}$ at a given instant in time ***i***. PAM uses several types of structures including unary relations, binary relations, ternary relations, to form cases, events, types, plans, goals, and other concepts.

3.6. Activation

In PAM activation is defined as "recency," and therefore a system lifespan time function is used to mark active model entities. A system-wide activation interval parameter is also defined which demarcates the cutoff between active and inactive model entities.

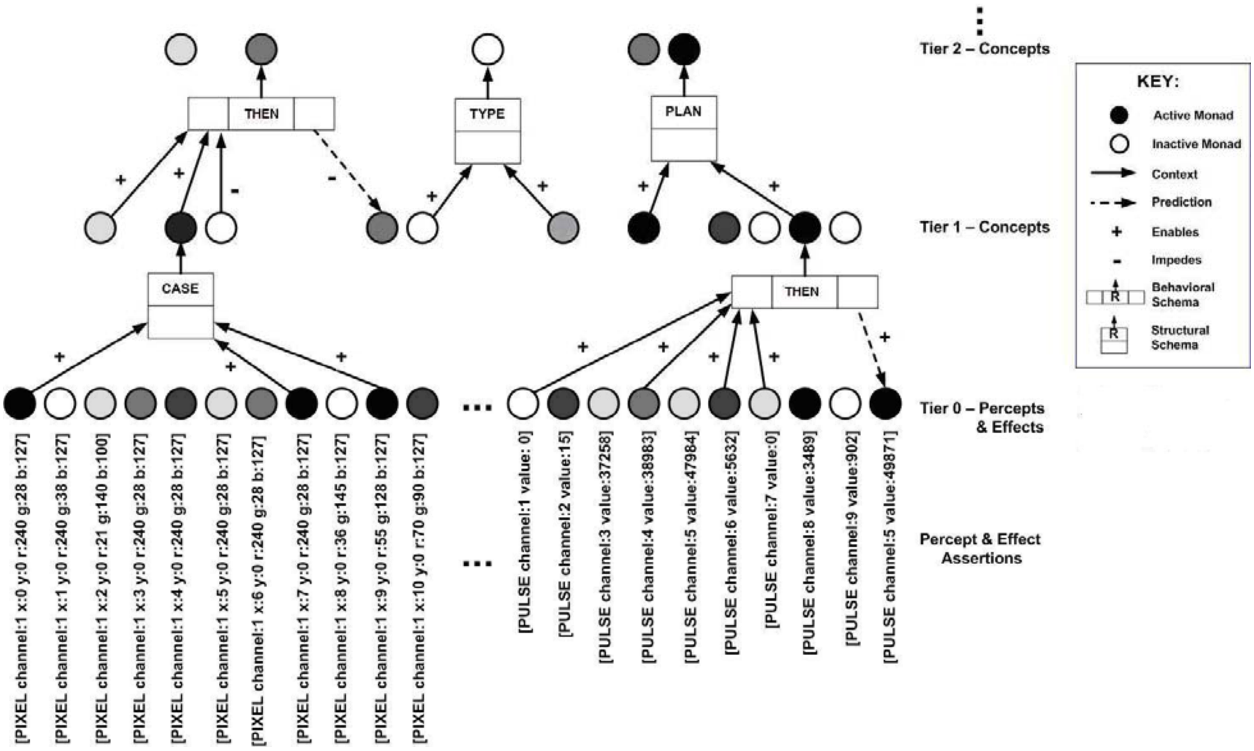

Figure 2. Assertions, monads and schemata.

The Tier Activator system element is responsible for activating monads representing structural schemata, and the Prediction Matcher system element is responsible for activating monads representing behavioral schemata. Percept monads are activated by Detectors and Effect monads are activated by Effectors.

3.7. Cases and Events

Holland et. al [4] describe mental models as "assemblages of synchronic and diachronic rules organized into default hierarchies and clustered into categories." The PAM system contains processing elements which use structural schemata to form synchronic (concurrent) and diachronic (sequential) relationships among monads.

As monads become activated within PAM, a concurrence associator process connects groups of concurrently active monads into "cases." A case represents a synchronic relationship (existing at one instant in time) as specified by Holland et. al.[4]. Similarly, a sequence associator process clusters monads into temporal "events." An event represents a diachronic relationship (existing across a period of time)[4].

3.8. Types and Plans

Cases represent instances of types (i.e., Classes). An inductor process synthesizes new types and clusters existing cases into these types. Types can also be clustered to form hierarchies of types. In a similar fashion events represent instances of plans. The inductor process aggregates events into newly synthesized plans, and may further form hierarchies of plans. Pickett & Oates [20] have done extensive work in concept formation -- as demonstrated by their work on the Cruncher. An incremental concept formation algorithm based on the Cruncher is used in the Inductor.

A reasoner processing element in PAM builds upon these cases, types, events and plans by using structural schemata to form other higher level relationships.

3.9. Inference

Ryszard Michalski [1] [3] [12] has long been involved in multistrategy learning and inference. His work has largely focused on logical models of inference in Artificial Intelligence systems. He and his co-authors have developed a method of inference involving Dynamically Interlaced Hierarchies. The premise is that language is organized into disparate hierarchies or taxonomies which are connected by traces (i.e., sentences, in PAM, cases). Inference then is simply a matter of performing transformations on traces (i.e., substitutions of words within sentences) according to the placement of the nodes (words) in the related hierarchies. (See Figure 4).

In their work on Multistrategy Inference, Heib and Michalski [2] define some basic knowledge generation transmutations which can be performed by making simple substitutions of select nodes in a case or event (referred to as a "trace" in the literature) from various related taxonomies within a model. By substituting constituent monads of a case (or event) according to specific transmutations, new cases (or events), and by extension new inferences, can be formed.

Hauser [13] defines thinking as navigating through the content of a word bank (a flat ontology, or heterarchy). "Navigation is the temporary activation of certain propositions in a potentially vast amount of content for purposes of inference and conceptualization (selecting what to say)." This view is consistent with Michalski's traces in Dynamically Interlaced Hierarchies.

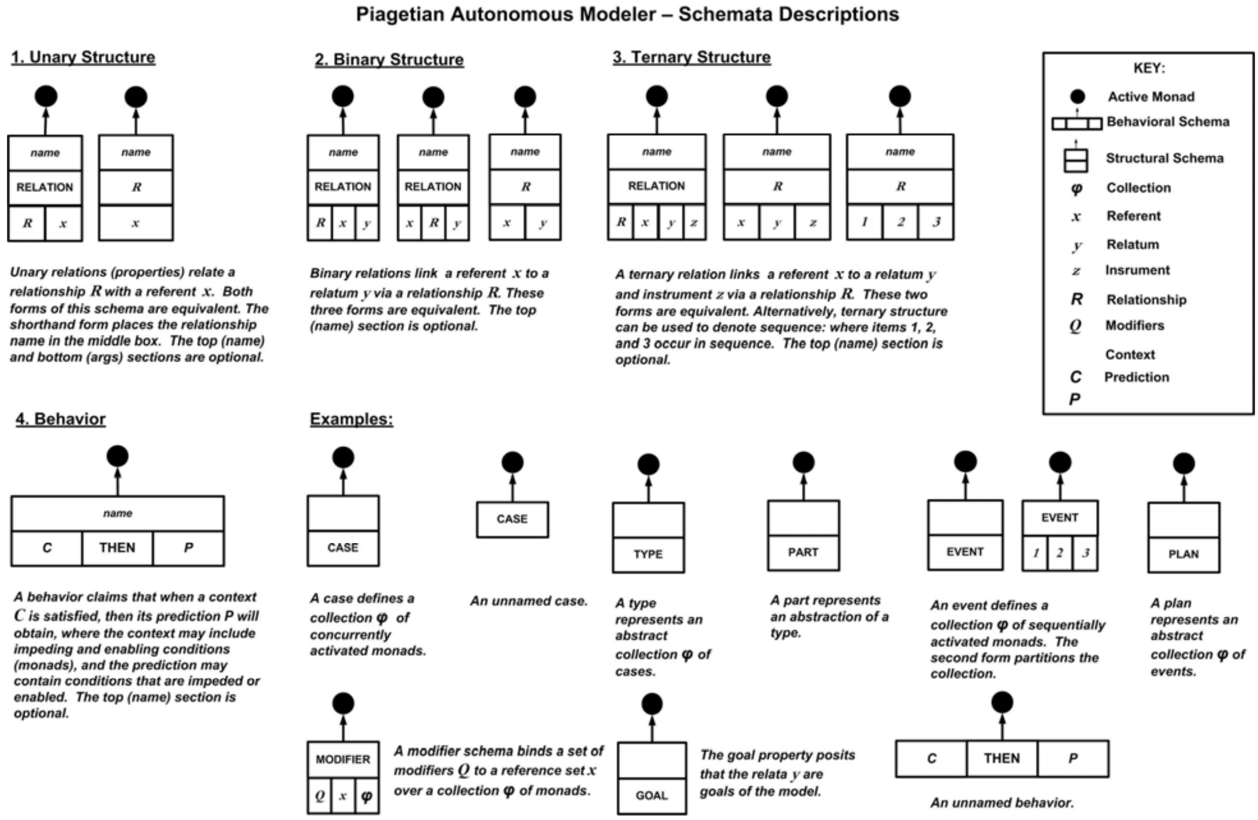

Figure 3. Schemata varieties in PAM.

Riesbeck & Schank [6] discuss the utility of Case based reasoning and implement a system to demonstrate their theories, Unfortunately, their system is largely constructed a-priori and does not employ dynamically constructed cases and events based on interaction with an environment. The Reasoner element in PAM is responsible for performing inference based on Michalski's theories of Inference [1] [3] [12]. This combination of interactionist model construction and multi-strategy inference is novel.

Tecuci and Michalski [3] further define specific transmutations which can be applied to cases and events to make inferences: Generalization, Specialization, Abstraction, Concretion, Similization, Dissimilization, Agglomeration, Decomposition, Prediction, Explanation, Selection, Generation, Characterization, Discrimination, Association, Disassociation, Reformulation, Randomization, Insertion, Deletion, Replication, Destruction, Sorting, Unsorting.

3.10. Equilibration

Piaget [7] [8] discusses the notion of equilibrium and equilibration. Soros [9] also discusses the notion and use of equilibrium. For Soros, equilibrium occurs when predictions are consistently successful (with minor divergences). Disequilibrium, conversely, is when predictions are consistently failing [9]. Convergence with reality means trending towards more and more successful predictions. Divergence with reality means more and more failed predictions. (George Soros' theories of Human Uncertainty and Reflexivity are instructive here.)

Soros further theorizes that divergences occur in two ways: through Static and Dynamic Disequilibrium. Static Disequilibrium occurs when reality changes and the mental model does not change. Dynamic Disequilibrium occurs when a mental model changes but the underlying reality has not changed

The PAM system contains an Equilibrator component which modifies predictions based on prediction success or failure. The Equilibrator regulates the accuracy and consistency of the systems predictions. Failed predictions are refined to identify a failure cause through a process called Marginal Attribution (Drescher [1]).

In addition, PAM applies genetic techniques to successful predictions. Behavioral schemata which are successful in predicting outcomes of actions become candidates for genetic transformations such as crossover and mutation per Goldberg [5].

3.11. Consolidation

This component performs the automaticity and forgetting functions within PAM and serves to reclaim any low utility or useless model entities.

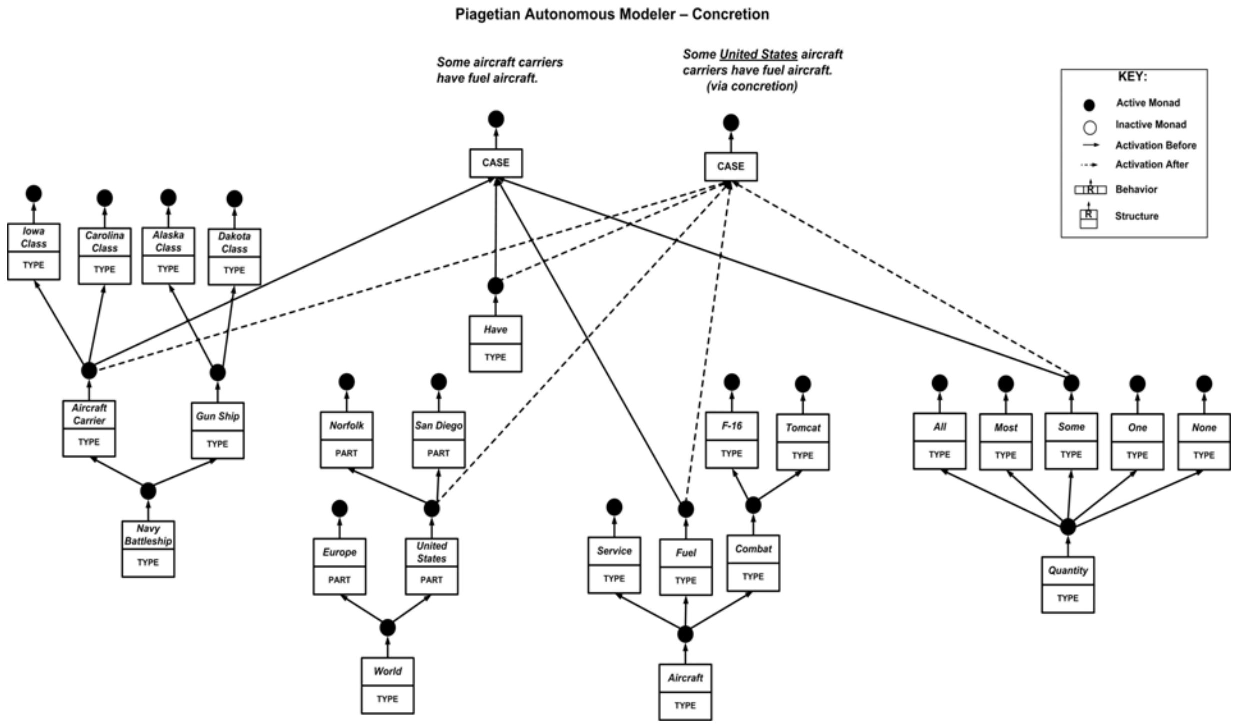

Figure 4. An inference example (adapted from Heib & Michalski [2]).

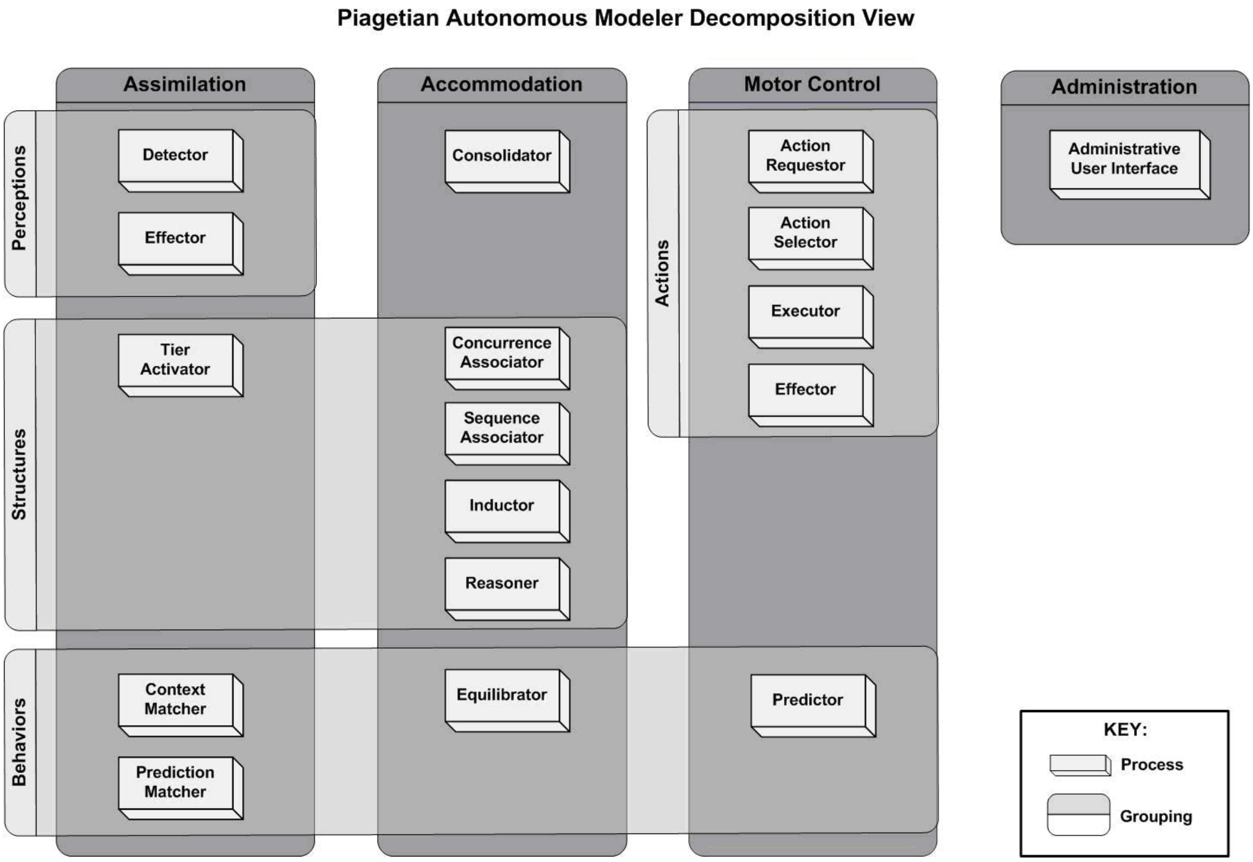

Figure 5. PAM Decomposition. Note that perception elements interact with the environment, structure elements activate and create new associations among structural schemata, behavior elements activate and reward behavioral schemata, and action elements determine which actions should be performed.

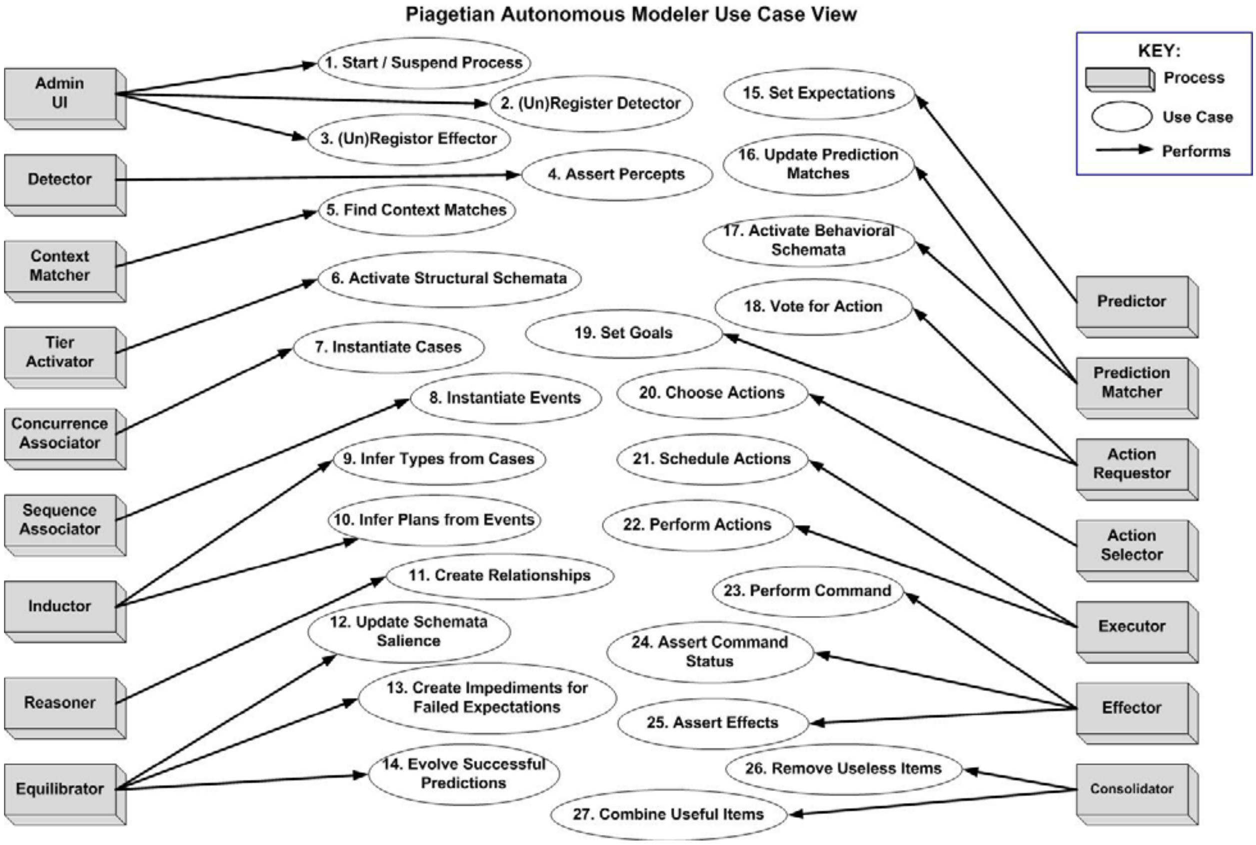

Figure 6. PAM Use Cases.

3.12 Components

(1) Detector. Transforms sensor data into activated percept monads within the model.
(2) Tier Activator. Activates the monads of structural schemata.
(3) Effector. Transforms actions into environmental commands, receives feedback on the execution status of the commands, and activates the corresponding effect monads within the model.
(4) Context Matcher. Matches behavioral schemata contexts with activated monads in the model. A context is satisfied when all enabling monads are active and no impeding monads are active.
(5) Prediction Matcher. Matches expectations (i.e., expiring predictions) to activated monads in the model, and when satisfied, activates the monads representing the behavioral schemata.
(6) Concurrence Associator. Creates "Cases" based on the concurrently activated monads in a lower tier.
(7) Sequence Associator. Creates "Events" based on the sequentially activated monads in a lower tier.
(8) Inductor. Aggregates "Cases" into "Types" and "Events" into "Plans".
(9) Reasoner. Infers new relationships using multiple strategies.
(10) Equilibrator. Revises behaviors according to failure and evolves behaviors according to success.
(11) Predictor. Sets an expiration time for a behavior's prediction (thereby creating an "expectation") based on actions the system has committed to undertake.
(12) Action Requestor. Bids for actions to be performed based on goals (inactive predicted monads), and satisfied behavior contexts.
(13) Action Selector. Decides which action to schedule for execution based on multiple biases [22].
(14) Executor. Invokes an action.
(15) Consolidator. Removes useless items and combines useful items.
(16) Administrative User Interface. Provides a system control dashboard and allows parameter adjustment.

4 EXPERIMENTS

Two experimental domains are proposed for this phase. A foraging domain (based on the Pioneer 3 DX robot simulation environment as described in Chaput [14]) , and a robot play domain (similar to Kaplan et. al. [22]) where a wireless mobile robot with audio and visual sensors can interact with various objects.

5 IMPLEMENTATION STATUS

The prototype is in the detailed design phase. It will be implemented using an agent platform and a database (either conventional SQL, high performance SQL, or NO-SQL).

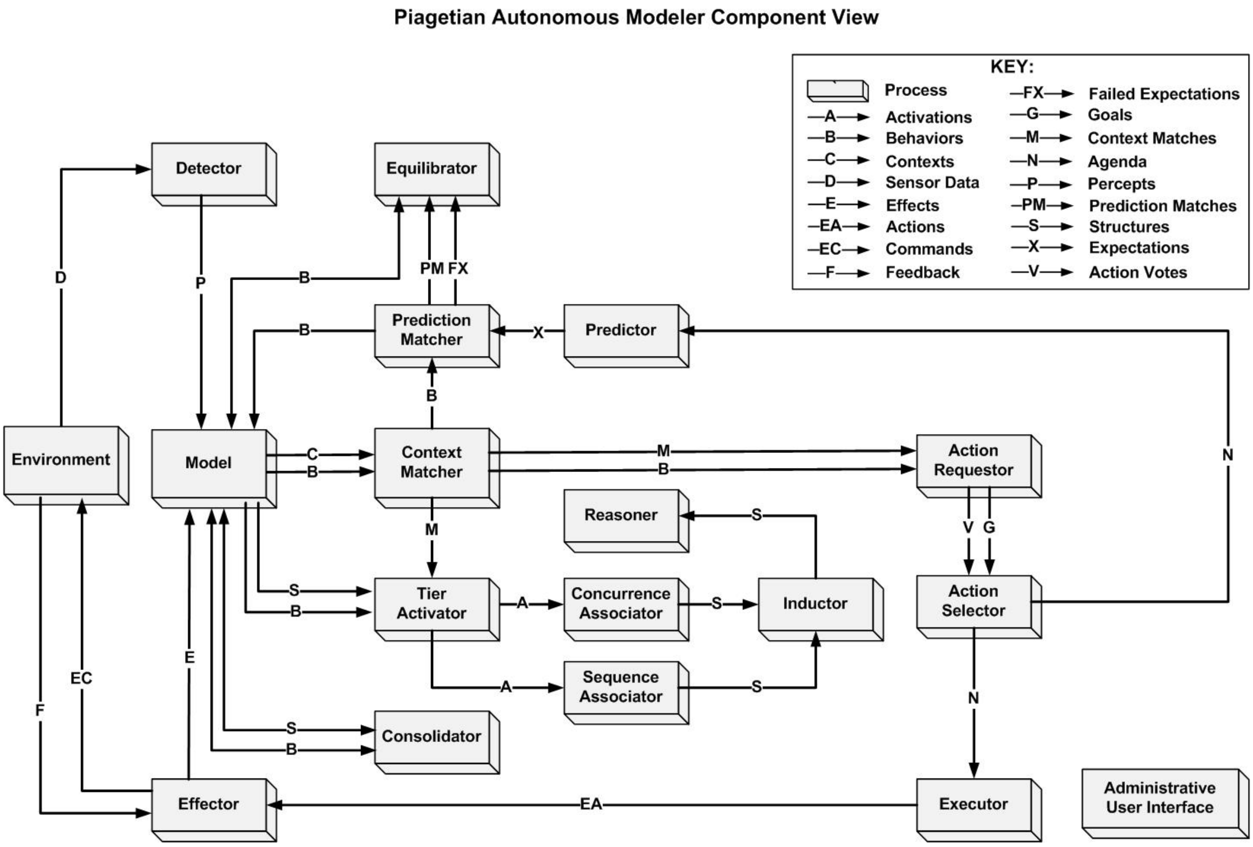

Figure 7. PAM Components.

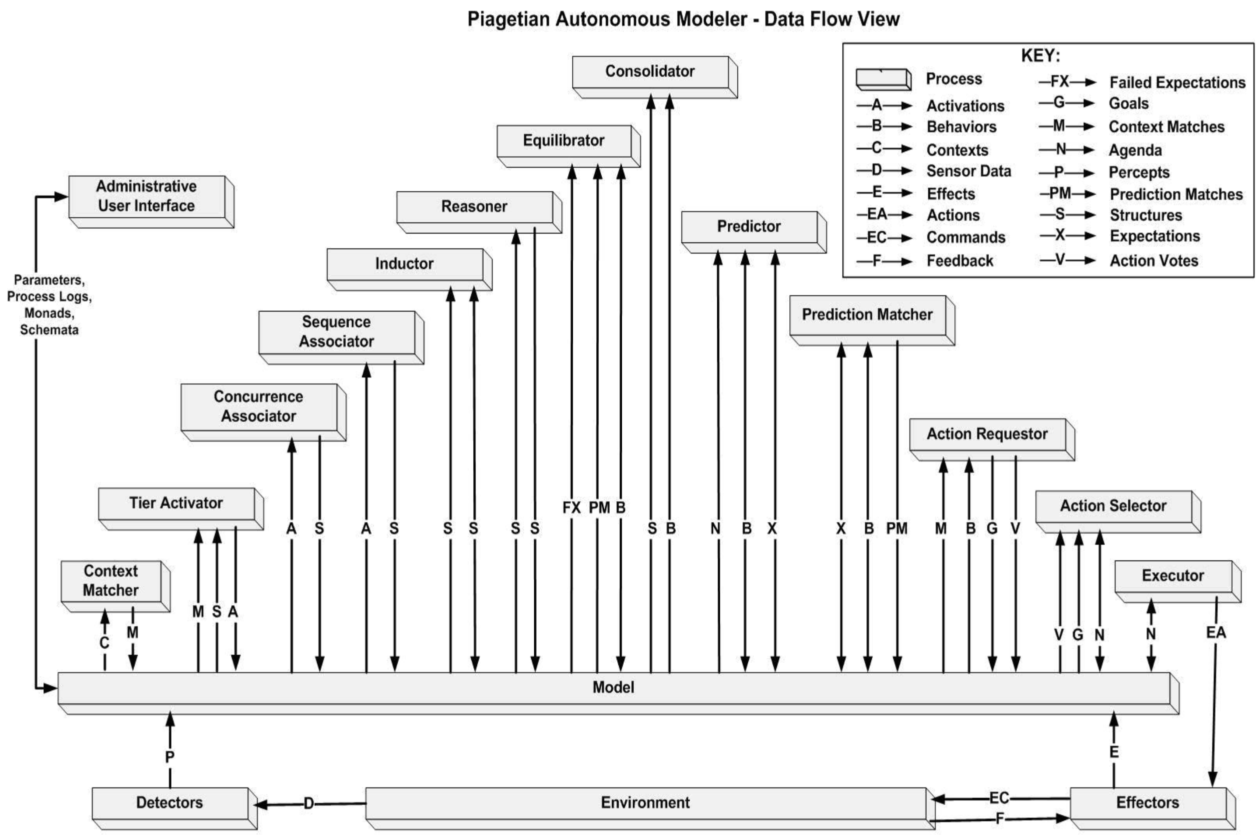

Figure 8. PAM Data Flow.

6 CONCLUSIONS & FUTURE WORK

The PAM architecture promises to be an exciting direction for experimentation in early developmental AI. In contrast to systems such as Chaput's CLA [14] which uses self organizing maps (SOMs), the PAM architecture seeks to exploit structural schemata, multi-strategy inference, cases, events and novel time based interconnections between percepts, action effects, and synthesized concepts.

7 ACKNOWLEDGEMENTS

This author would like to thank Frank Guerin for his broad knowledge of the field and copious references; Marc Pickett and Tim Oates for references and code for the Cruncher; and Roland Hausser for sharing his thoughts on learning and inference.

8 REFERENCES

[1] Drescher, Gary L. Made Up Minds: A Constructivist Approach to Artificial Intelligence. (1991)

[2] Heib, Michael R. & Michalski, Ryszard S. *Multitype Inference in Multistrategy Task Adaptive Learning: Dynamic Interlaced Hierarchies*. Proceedings of the Second International Workshop on Multistrategy Learning, Harpers Ferry WV. (1993)

[3] Tecuci , Gheorghe & Michalski, Ryszard S. *Inferential Theory of Learning*. Machine Learning, A Multistrategy Approach, Volume IV. (1993)

[4] Holland,, John H., Holyoak, Keith J., Nisbett, Richard E, Thagard, Paul R. Induction: Processes of Inference, Learning, and Discovery. (1989)

[5] Goldberg, David E. Genetic Algorithms in Search, Optimization, and Machine Learning. (1989)

[6] Riesbeck, Christopher K. & Schank, Roger C. Inside Case-Based Reasoning. (1989)

[7] Piaget, Jean & Rosin, Arnold The Development of Thought: The Equilibration of Cognitive Structures. (1978)

[8] Piaget, Jean, Brown, Terry, Thampy, Kishore J. Equilibration of Cognitive Structures: The Central Problem of Intellectual Development (1985)

[9] Soros, George. The Alchemy of Finance: Reading the Mind of the Market. (1993)

[10] Kuhn, Thomas. The Structure of Scientific Revolutions. (1962)

[11] Indukhya, Bipin. Metaphor and Cognition: An Interactionist Approach. (1992)

[12] Alkharouf , Nabil W. & Michalski, Ryszard S. *Multi-Strategy Task Adaptive Learning Using Dynamic Interlaced Hierarchies*Proceedings of the Third International Conference on Multistrategy Learning. (1996)

[13] Hausser, Roland. A Computational Model of Natural Language Communication: Interpretation, Inference and Production in Database Semantics. (2010)

[14] Chaput, H. *The Constructivist Learning Architecture: A Model of Cognitive Development for Robust Autonomous Robots.* (2004)

[15] Pickett, Marc, Miner, Don, Oates, Tim. *Essential Phenomena of General Intelligence.* (2005)

[16] Lenat, Douglas B. & Guha, Ramanathan V. Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project. (1990)

[17] Guerin, Frank . *Learning Like Baby: A Survey of AI Approaches.* Accepted for Knowledge Engineering Review (to appear)

[18] Michalski, Ryszard S. & Stepp, Robert E. *Learning from Observation: Conceptual Clustering*Machine Learning An Artificial Intelligence Approach, Michalski, Carbonell, Mitchell (Eds.) (1983)

[19] Morrison, Clayton T., Oates, Tim, King, Gary W. *Grounding the Unobservable in the Observable: The Role and Representation of Hidden State in Concept Formation and Refinement* (2001)

[20] Pickett, Marc & Oates, Tim. *The Cruncher: Automatic Concept Formation Using Minimum Description Length*Abstraction Reformulation and Approximation, 6th International Symposium, SARA 2005. (2005)

[21] Miller, Michael S. P. *The Piagetian Autonomous Modeler Architecture, (Unpublished)* (2010)

[22] Kaplan, Frederic, Oudeyer, Pierre-Yves, Bergen, Benjamin. *Computational Models in the debate over language learnability.* Infant and Child Development 17(1). (2008)