# A Piagetian Model of Early Sensorimotor Development

**Frank Guerin** **Daniel McKenzie**
Department of Computing Science
University of Aberdeen
Aberdeen, AB24 3UE, Scotland
{f.guerin,u14dm4}@abdn.ac.uk

## Abstract

We are interested in developing a computational model of Piaget's theory of sensorimotor intelligence. The main difficulty is that Piaget's theory is quite vague. We analyse existing approaches and find that they model some of the major features of his theory, but that these are not adequate to account for the stage transitions which Piaget described. Instead of copying features of his theory we advocate developing a cognitive model to account for the infant development paths which he described. As a first step in this direction we present a computational model of Piagetian development in the early sensorimotor stages. In order to extend our computational model to later sensorimotor stages, we believe that further psychological studies are required to clarify issues left open in Piaget's account.

## 1. Introduction

We are interested in building AI systems which can learn their own world knowledge autonomously. This is a motivation shared by a number of researchers in Artificial Intelligence (Drescher, 1991; Chang et al., 2006; Mugan and Kuipers, 2007; Stojanov et al., 1997). Piaget's theory of constructivism gives an account of how humans build up their world knowledge through their interactions with the environment. Unfortunately Piaget's theory does not give the level of detail which would be necessary to inform a computer implementation. From an AI point of view we could choose to proceed with an implementation loosely based on his theory, with the details filled in by intelligent engineering. This we call the *engineering approach*. The pitfall here is that researchers are likely to come up with solutions which are not the same, and not as good as those which evolution has found for the biological system. One of the first striking observations which Piaget made is how different the child's mind is compared to the adult's. Most adults are very surprised to discover the way in which young children reason about the world, and the logically inconsistent models which they are happy to entertain. Given this evidence it seems highly unlikely that an intelligent programmer (without input from cognitive science) will come up with representations which are anything like those which are being used by successful biological implementations of intelligence. Furthermore, the history of AI suggests that the representations which the engineering approach comes up with suffer severe limitations when attempting to build systems with domain general competences; i.e., when we make it up ourselves we do it the wrong way (this point was well made by Brooks (1991)).

Thus we believe that it is necessary to do the science (flesh out the details of a theory of cognitive development) before the engineering (build an intelligent system which can autonomously learn). Our goal then is to come up with a precise account of cognitive development. Computers give us a benchmark for precision which was not available when Piaget did his main work on infancy (Piaget, 1936, 1937).

We foresee that the path towards achieving our goal will require alternation between studying both computational and cognitive systems. We can begin with a rough account of a particular development path (i.e., a sequence of behaviours), as described by Piaget's longitudinal studies on his infants. We then attempt to build a computer implementation of the path. This inevitably throws up new questions about details of the development path, for example: what training episodes are necessary or sufficient, what aspects of those episodes are necessary, what kind of competence is present at intermediate stages? Psychological studies with infants will be necessary to answer these questions. The computer implementation can then check if the answers are plausible, and so on. This paper describes the first steps in this research endeavour. We have an implementation which models some of the first stages of development as described by Piaget, taking the model further will require psychological studies to clarify a number of questions not addressed in the literature.

In Section 2 we review AI works inspired by Piaget and identify what we believe to be missing from these efforts. In Section 3 we briefly describe Piaget's sensorimotor stage 2, which it is our goal to model. In Section 4 we describe our own simulated baby. Section 5 concludes and outlines future directions.

## 2. Motivation

In this section we motivate our research by first taking a look at Piaget's theory, and highlighting what we believe to be the most exciting feature of his theory from an AI point of view (i.e. stage transition). We then review some existing work in AI which is inspired by Piaget's theory, and show that this feature has not been captured. Finally we outline a possible path towards addressing this deficiency.

The idea of *stages of development* is one of the central tenets of Piaget's theory[1]. According to Piaget these stages do not unfold due to any preprogrammed maturational process, but due to the operation of a learning mechanism and its interaction with the environment. It is Piaget's thesis that this learning mechanism is innate in the infant and is invariant, while the knowledge structures which it builds are variable and become increasingly sophisticated. Piaget refers to this learning mechanism as a "functional nucleus" (Piaget, 1936). The essential message of Piaget's theory could be phrased as follows: the continuous operation of a single invariant learning mechanism, in its interactions with the environment, leads the infant to exhibit qualitatively different forms of behaviour, of increasing sophistication. When looking at infants' behaviours in detail, as Piaget has done, there is a continuity apparent, with each new behaviour being due to a small adjustment of existing behaviours. Yet, when looking at the behaviours exhibited throughout the whole of infancy, qualitatively different behaviours can be identified, and boundaries can be drawn (though Piaget was at pains to point out that the boundaries are by no means clear cut and that every intermediate behaviour is also present[2]).

The essential element we would like to bring across to AI then is the idea of a mechanism which can develop qualitatively different forms of behaviour. This holds the promise of allowing a program to build its own knowledge structures, rather than requiring them to be handcoded. However Piaget's description of how the learning mechanism operates lacks the precision which would be required to create a computational model. We have the following from Piaget then: (1)a detailed description of infant behaviours, which describes how behaviours build on each other and give rise to qualitatively different forms; (2)a vague theory of how the learning mechanism works in the abstract. Both of these contributions are valuable. The longitudinal studies which Piaget did on his own three children provide us with sequences of observations which show how an infant is gradually extending his/her abilities, and more importantly, they identify the experiences which have led to the development of a particular new behaviour. We will call these sequences *development paths*. There is also considerable value in the abstract theory; he gives us a high level overview of the "big picture" of cognitive development, but we lack detail. There are broadly speaking two ways for AI to go from here: (1) Try to model the observed development paths, with the theory as a general guide, and flesh out a more detailed account of a learning mechanism which could account for the development; this would likely require significant input from further psychological studies. (2) Try to work from the abstract theory, taking the main aspects of the theory which make sense, and using our own creativity as AI programmers to fill in the remaining details required to implement a complete AI system. It is our argument that the first approach is to be preferred because we simply do not have an adequate high level theory which accounts for what we believe to be the essential aspect of Piagetian development: the transition between stages of qualitatively different behaviour. In reviewing the existing works in Piagetian AI below (which all follow the second approach) we will be making the argument that they are embodying aspects of Piaget's abstract theory, but that those aspects are in no way adequate to account for the stage transitions described in Piaget's observed development paths.

Central to Piaget's learning mechanism is the *Schema*, which is a unit of knowledge. A schema gathers together an ensemble of perceptions and associated motor actions. A typical example could be the schema of pulling a string to shake a rattle. This schema includes visual and tactile perceptions of the string, the action of grasping and pulling, and the expectation of hearing and seeing the rattle shake. Schemas always seek opportunities to repeat themselves, and in doing so they generalise to new situations, differentiate into new schemas, combine with other schemas, etc. These aspects of Piaget's theory are captured by Drescher's *Schema Mechanism* (Drescher, 1991). Drescher's schemas were 3-part structures consisting of a context, action, and result. A schema is a prediction about the world: if its action is taken in the context specified, then the result

[1]We are aware that there is considerable controversy over Piaget's theory, with studies contradicting it, and counterclaims in his favour. This paper will not delve into the controversy, interested readers could consult Cohen and Cashon (2003) and Haith (1998) as a starting point. For the purposes of this paper it is sufficient to know that there are leading psychologists who still stand by his theory.

[2]See for example Piaget (1936, p.157) "Of course, all the intermediaries are produced between the primary circular reactions and the secondary reactions".

is predicted. Drescher's system interacts in a *microworld*, which models a baby, with hand, mouth and eye, and thus is close to modelling Piagetian developments. Drescher also introduced the idea of a *synthetic item* which acts as a higher level unit of knowledge, representing the conditions of success of a schema. He used this idea to capture the Piagetian notion of a persistent object concept; i.e., the baby's belief that an object continues to exist even when it is no longer directly perceived. The synthetic item has been very influential and has been cited and used in a number of the subsequent works. Recent work (Holmes and Isbell, 2005) has brought Drescher's ideas more into the mainstream machine learning community.

However, Drescher's work has not led to systems which exhibit cognitive development, with stages of qualitatively different behaviours. For this reason Drescher's work is somewhat disappointing in that it fails to capture what we believe to be the essence of Piaget's contribution. Drescher has made use of some of the principles from Piaget's theory, and has added his own creativity (in inventing the synthetic item), but has not followed the development paths described by Piaget. For example, the notion of object permanence has a long development path in Piaget's theory. One manifestation of the beginnings of an object concept is when the infant removes an occluding object to reveal a recently hidden object (Piaget, 1936, Obs. 126). Piaget, is quite clear about the precursor to this behaviour: it is the behaviour of removing an obstacle to grabbing a desired visible object. The principles embodied in Drescher's system are not sufficient to account for the development of object concept along the path Piaget described, and in fact Drescher does not attempt to model the obstacle removal behaviour.

It is perhaps unfair to criticise Drescher's work in this way, given that it was the first to make a serious attempt to bring Piaget's theory to AI. However, this criticism is equally applicable to subsequent works. Attempts to model (or draw inspiration from) Piaget's theory in the sensorimotor stages have a reasonably long history by now, with Drescher having commenced his work in the mid 80's. Subsequent works have followed the same pattern of copying aspects of the theory but not accounting for Piaget's development paths. We argue that the ongoing research in this direction is not likely to get full value from Piaget's work. We think it important to highlight exactly what is lacking in order to avoid repeating the omission.

The remaining works will be reviewed summarily in the interests of brevity and because our comments on them are mostly a reiteration of the comments on Drescher. The Petitagé system (Stojanov et al., 1997) has *expectancy triplets* which are similar to Drescher's schemas. The world in which it was trialled had walls and obstacles, and thus was not like Drescher's microworld, so a comparison with infant development pathways is not possible. Again, it is inspired by principles from Piaget's theory. Similarly for the CALM system (Perotto et al., 2007) which has schemas similar to Drescher's and runs in an abstract domain. The work of Mugan and Kuipers (2007) does experiment in a domain where a baby is simulated, however it is only loosely inspired by the Piagetian idea of acquiring knowledge autonomously, and does not attempt to copy Piaget closely.

Chang et al. (2006) provide a quite different approach to coding Piagetian schemas. Their system can learn "gists" which are compositions of schemas for certain tasks. This has been successfully applied to learn behaviours in a simulated world, for example a creature learns to sneak up on, and catch, a cat. The schemas learnt have also been transferred to similar situations in slightly different scenarios. It is only loosely based on Piaget's theory, and does not attempt to recapitulate infant development.

A general theme that emerges from the above works is that there is an effort to engineer systems which exhibit certain hallmarks of Piagetian development. The problem with copying features of Piaget's theory, such as hierarchical construction of knowledge, or schema mechanisms, is that the vagueness of his description means that there are too many possible learning mechanisms which can embody these features; most of them are unlikely to lead to systems which learn like humans. Indeed Parisi and Schlesinger (2002) convincingly show that neural networks (i.e. pretty much any neural network) possess properties of Piaget's theories of assimilation and accommodation, and are also a good model of Piagetian schemas. The vagueness of Piaget's description means that his theory is essentially an unfinished theory. We need a complete and precise theory before we can use the elements of that theory in AI systems. This is why we advocate going back to the behaviours Piaget described, to model them, and make our own detailed theory to account for what is happening there. What we are advocating then is a model like Drescher's, in that it simulates a baby, but one that closely follows infant development. After such studies are done we could abstract away and come up with a mathematical theory of the essential features of constructivist processing, for implementation in intelligent systems, but this can be expected to take a very long time. The work described in the next section is a very small step along this path.

## 3. Piaget's Stage 2 Infant

Here we briefly describe the five substages which Piaget identified within his second stage of infancy, and which it is our goal to model. We will number

these stage 2.1 . . . 2.5. Note that stage 1 of infancy is mostly the exercising of reflexes, especially sucking and searching with the lips; we skip this stage in our model. The substages of stage 2 are as follows:

(2.1) The infant learns to suck his thumb, or hand. The infant also does some reflex grabbing of objects which touch his hand. (2.2) The infant learns to look at objects, and to look at his hand with interest. (2.3) The infant learns to take a grabbed object to his mouth, for sucking. (The hand finds the object without the aid of vision.) (2.4) The infant can grab any object so long as the object and his hand are both visible. (2.5) The infant can grab any visible object even if his hand is not in view.

## 4. The Simulation

Our simulated baby lives in a simple 2D world with a few rigid square blocks. A rigid body physics engine simulates the physics of the world, including friction and collisions between blocks; gravity has been disabled. The baby (see Figure 1) has a single movable arm, consisting of two rigid rectangular blocks: an upper arm and a lower arm. The upper arm can rotate at the shoulder, and the lower arm rotates from the elbow. There is a hand at the end of the lower arm (represented by a square). When the hand overlaps with a block, a touching sensation is generated, and if the grab action is then taken, the block will then move together with the hand, until released. The baby's mouth is the square at the left of the figure. The baby also has a field of vision, bounded by the two lines emanating from the mouth. The point of intersection of the two lines is the "eye". The field of vision can be rotated about the eye. As in Drescher's simulation, the centre of visual attention has a fovea; in our system the fovea is capable of moving along a path between the two lines (and equidistant from them). The fovea is shown as an outlined box in the figure.

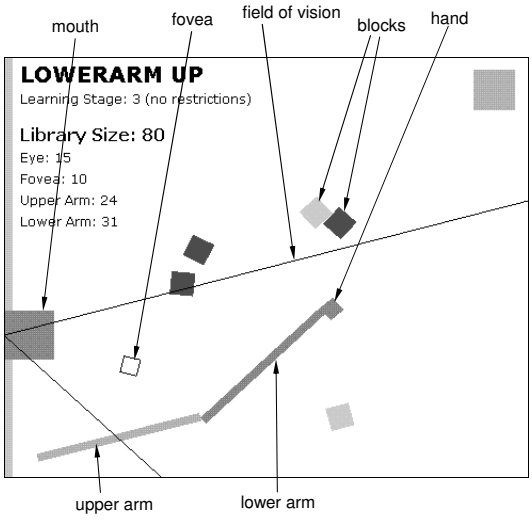


Figure 1: The simulation.

The baby has a number of sensors as follows. Firstly there are five sensors to record interactions with objects: touching or grabbing by the hand, touching or sucking by the mouth, and seeing with the fovea. An additional "seen_objects" sensor returns a set of objects that are in view (i.e., between the lines bounding the field of vision); each object in the set is described by a triple: object type, angular displacement from centre of visual field, distance from fovea. Note that the baby can see his hand (it is an object). Therefore in the scenario depicted in Figure 1, the "seen_objects" sensor returns a set of two objects, one is a block, and one is the hand. Finally the baby also has four sensors for the positions of: upper arm, lower arm, eye angle, and fovea.

The following are the twelve actions which the baby is capable of. There are four arm actions: the upper arm and lower arm can both (independently) move up and down. There are four actions relating to the eye: the angle of vision can move up and down, and the fovea can move forward and backward. Finally there are four non-movement actions: the hand can grab; the mouth can suck; the eye can stare; the fovea can fixate. There is no action for the hand to release what is grabbed; a grabbed object is simply released after a random time interval.

### *4.1 Schemas*

The baby takes actions by executing *schemas*. The schema is modelled on Drescher's (Drescher, 1991). A schema is a 4-tuple $\langle S1, R, S2, \mathit{TargetValue}\rangle$. Where $S1$ is the initial world state sensed before the schema is executed, $R$ is the response (action) taken, and $S2$ is the prediction for what will be sensed after completion. $\mathit{TargetValue}$ is the addition over Drescher's schema, and it records reinforcement learning values, indicating how useful this schema is to achieve other schemas. $\mathit{TargetValue}$ is in fact a list of pairs $\langle \mathit{Schema\_id}, \mathit{Value}\rangle$ where each $\mathit{Schema\_id}$ identifies another schema which this schema has a value towards, and each $\mathit{Value}$ is the value of the action. A list is used because a schema may be useful for achieving more than one goal, and hence has a value towards each of those goal schemas. For example, a hand movement schema may move the hand closer to contact with the mouth, and hence will have a value towards the mouth sucking schema, but it may also move the hand closer to the fovea, and also have a value towards this goal (which we call a target).

There are three types of schemas: *bare* schemas, *target* schemas, and *normal* schemas; these will be described now.

The baby starts with twelve initial schemas in a library, and thereafter generates his own new schemas and adds them to the library. The twelve

initial schemas correspond to the twelve actions listed above. There are eight *bare* action schemas corresponding to the movement of arm, eye angle and fovea. These are bare in the sense that they have only an $R$ value, and have empty values for $S1, S2, TargetValue$. For example the bare schema for moving the lower arm up is represented as $\langle -, lowerarm\_up, -, - \rangle$. These schemas correspond to random actions which the baby may take; they are not particularly interesting to the baby unless they lead to some interesting result. The remaining four schemas are special *target* schemas; these correspond to actions which the baby finds interesting, and which we would like to rediscover after he achieves them. They include the actions: mouth_suck, hand_grab, eye_stare, fovea_fixate. The innate schema for the mouth sucking specifies (in its $S1$ part) that the mouth should be touched; thus touching the mouth effectively acts as a trigger for sucking. Similarly for the hand grabbing; the hand should be touched. The eye stare schema is triggered when any object falls on the centreline of the field of vision; stare will then be activated to hold the eye at that position. Similarly, if an object is present at the fovea, this will trigger the fovea fixate action to hold the gaze there. These initial schemas are intended to model the innate movement schemas of the infant, including reflexes such as sucking, grabbing and gazing, which the infant performs with interest.

If a target schema achieves a new unexpected result (i.e., a new $S2$), then it is differentiated into two new schemas, one for the old expected result, and one for the new one. This is necessary to distinguish the difference between sensing and sucking the hand or a block, or touching and grabbing different blocks for example.

When a bare schema is selected for execution (e.g. $\langle -, lowerarm\_up, -, - \rangle$), the information for its $S1$ part is filled in based on the current state of the environment (as recorded by the sensors). After completion of the action, $S2$ is filled in with information about any aspects of the environment which have changed. The resulting schema is called a *normal* schema. There are no normal schemas in the library when the program starts, but they quickly become the most common type of schema after learning.

## *4.2 The Basic Learning System*

The main loop of the program randomly selects a schema from the schema library, executes it (if possible), and updates the library appropriately based on the results (this often results in the addition of new schemas to the library). As mentioned above, there are three types of schemas: $^{(1)}$target schemas, $^{(2)}$normal schemas, and $^{(3)}$bare action schemas. The program must treat each differently if it is selected for execution, as described now:

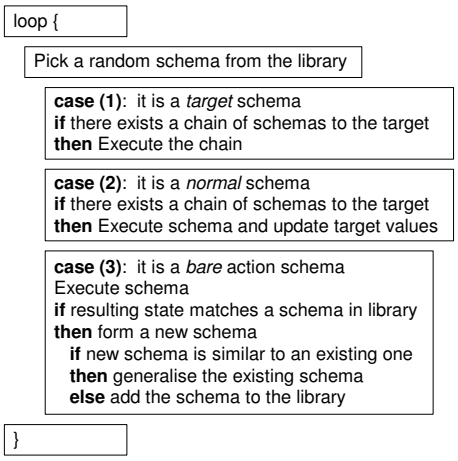


Figure 2: Rough description of learning algorithm.

(1) A target schema can be directly executed if the current environment matches its $S1$ (for example the mouth_suck schema's $S1$ requires that the mouth is touched). Alternatively, there may exist a chain of normal schemas which can take the baby from where he is to a state matching the target's $S1$ (for example if the baby's hand is at $P$ the chain of schemas $sch4, sch2$ from Figure 3 can be executed). These schemas are found in the library by searching for schemas whose $S1$ matches with the current environment and whose $TargetValue$ contain this target schema. If available, this chain will be executed; otherwise the target schema fails to execute.

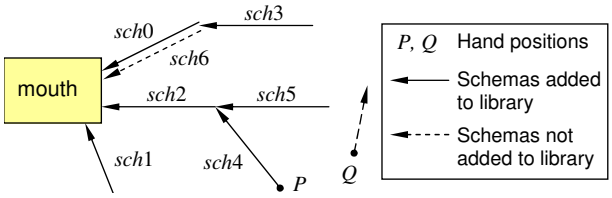


Figure 3: Learning Normal Schemas. The schemas indicated show arm movements converging on the mouth.

(2) If a normal schema is selected for execution, then it can be executed only if its $S1$ matches the current environment. If it is executed its reinforcement learning values will be updated.

(3) If a bare action schema $\langle -, R, -, - \rangle$ is selected for execution, then as mentioned above, a new normal schema $\langle S1, R, S2, - \rangle$ is created based on the environment. Let us call this schema $sch_i$. To determine if this new schema $sch_i$ should be added to the library, we need to know if it is useful to achieve any target. Therefore we attempt to perform a *partial match* with some existing normal schema $sch_j$ which has value; i.e., we need to compare the $S2$ of $sch_i$ with the $S1$ of some existing $sch_j$. A partial match means that it is sufficient for a subset of the sensors in the schemas' $S2$s and $S1$s to match, potential matches are scored based on the closeness of

the match. If there is no match then we discard $sch_i$ because it is not useful to achieve any target (see for example $Q$ in Figure 3). If there is a match, then we need to follow through a chain of schemas until a target which $sch_j$ has value towards is achieved, or not achieved. If the target cannot be achieved, then $sch_i$ is discarded, otherwise $sch_i$ is eligible for addition to the library, with a value that is discounted from $sch_j$. The next step is to remove the sensors from its $S1$ and $S2$ which were not matched by the partial match (because they have been proved to be irrelevant). This is one of the methods by which schemas are generalised. The reason for testing the whole chain (all the way until the target) before adding $sch_i$ is because the technique of partial matching may have made a match based on irrelevant sensors, so the target might not be successfully achieved.

A further test is performed before adding a new normal schema to the library. We check if the new candidate is similar to any existing schema in the library. Similarity here means that the $R$ parts of the schemas are identical as well as one of the target schemas which they have value towards, and both the $S1$s and $S2$s have at least one identical sensor. If they are similar we will *generalise* the existing schema to account for the differences in the new schema. Generalising removes any different sensors in the $S1$s and $S2$s of the two schemas, thus removing irrelevant aspects. This leads to schemas which capture only the relevant parts of the environment in their $S1$ and $S2$. For example, in Figure 3 $sch0$ and $sch6$ are similar, therefore $sch6$ will not be added, but $sch0$ will be generalised as a result of the discovery of $sch6$.

This completes the description of the basic learning routine. By learning a collection of schemas which can for example bring the arm to the mouth from any position (as in Figure 3), we model the acquisition of thumb-sucking in the infant. In this way our learning algorithm fits within the Q-learning framework of reinforcement learning; each schema records the value of a particular action from a particular state. It is more limited however, because there are no values defined for much of the space from which actions could be taken; schemas (and values) are only added as they are found to be interesting.

The mechanism described so far was successful to learn a number of individual behaviours, for example to (1) suck the hand from any position, (2) centre the visual field on an object to stare, (3) fixate the fovea on an object. The program also learnt a number of positions of eye and hand from which the hand could touch a seen object, and grab it.

### *4.3 Collecting Schemas as Options*

The acquisitions achieved thus far take us to sensorimotor stage 2.2. However it proved difficult for the baby to go further and learn grabbing and sucking of objects, for the following reason. When the hand grabs an object, and then takes it to the mouth, this creates a new sensation for the mouth (i.e., it is sucking that object rather than the hand). The sucking of the new object differentiates a new target schema, and the baby then has to learn which schemas have value to achieve this target. The baby already knows how to bring hand to mouth from any position; however, with the basic learning mechanism as described above it is necessary for the program to learn an entirely new set of schemas for bringing the hand, with object, from any position to the mouth.

This can be overcome by the introduction of *superschemas*. A superschema collects together a group of normal schemas and regards them as a unit, listed in the library as a single schema. When called it acts like a target schema, in that it attempts to call any normal schema which has a value towards it. The superschema idea fits within the *options* framework of reinforcement learning (Sutton et al., 1999). A superschema is a triple: $\langle a\ list\ of\ S1s, TargetSchema\_Id, TargetValue\rangle$. The $TargetSchema\_Id$ identifies the target schema which the superschema achieves. The $a\ list\ of\ S1s$ is a collection of all the $S1$s from all the normal schemas that were learned with a value towards this target. The superschema can be executed from any of these $S1$ situations. The $TargetValue$ is the value of the superschema to achieve some other superschemas.

At the initialisation of the program a superschema is generated for each target schema, these superschemas grow as the baby learns. When a superschema $I$ is selected from the library and executed, if the resulting situation matches the $S1$ of another superschema $J$ with some value towards a target, then superschema $I$ takes on a discounted value towards $J$. This is how superschemas can link up in a chain. In this way the baby first learns individual superschemas for each of the targets: sucking, orienting the gaze, fixating the fovea, and grabbing. Having acquired a few schemas in each superschema collection, the baby starts linking up the superschemas: orienting the gaze takes a value towards the superschema of fovea fixating; fovea fixating takes a value towards grabbing the object. This addition made it possible for a reliable grabbing superschema to develop. The grabbing superschema records the position of the seen hand relative to the fovea, when the fovea fixates on an object, hence it is reliable regardless of objects' positions. This ability was not learned reliably when fixating was not a superschema. The difficulty was that each normal schema, which moved the fovea to fixate on an object, needed to take on value towards the target schema of grabbing, and this had to be repeated for each position of the fovea.

Finally we return to the problem encountered in learning to suck an object, given that sucking the

hand has been learnt. The target of sucking the hand has a superschema which groups together all of the normal schemas which have a value towards hand sucking. This superschema is first learned with initial conditions ($S1$s) which specify an empty hand. Because of the technique of partial matching, the superschema may get called when the hand is in an appropriate position, but happens to be holding an object. In this case the held object is taken to the mouth, and the resulting sensation is different to expected (i.e., sucking object instead of hand). Therefore the superschema is differentiated. The differentiation is done on the value of the hand sensor, because it is this value that was not matched by the partial matching. The existing superschema is generalised into two new superschemas, one for hand holding object, and one for hand empty; i.e., their $S1$ specifies this condition, and the target they achieve is the specific (empty or holding object) target schema. To complete the differentiation, all of the subschemas which served the original hand sucking superschema are generalised so that their $S1$ does not specify any value for the hand sensor; finally they are all given target values which point to both of the new superschemas.

This improvement allows objects which are grabbed to be sucked (stage 2.3), and also allows a reliable coordination of vision and grabbing, taking the baby to stage 2.4. This is as far as our implementation has progressed so far, however we have worked out a minor adjustment which should take the baby to stage 2.5. We require that the hand can come and grab a seen object even if the hand is currently outside the field of vision. Piaget describes the way in which this particular behaviour was acquired in detail in the case of his daughter Lucienne (Piaget, 1936, Obs. 80). Upon seeing an object her desire to suck it was excited, but her hands were not visible, and she was not at the stage where she could bring them into view. However, the desire to suck motivated her to bring her hands towards her mouth, and then immediately upon seeing a hand she directs it towards the object (she knew how to do this stage 2.4). Thus the new behaviour is clearly a fortuitous combination of previously known behaviours, at its first performance. This can be modelled by introducing the idea of *interruption* to options (Sutton et al., 1999). When the infant is in a scenario where an object is in view, but the hand is not, the superschema of attempting to suck the hand can be triggered. Now the superschema can be interrupted after any of the normal schemas in the chain towards the mouth is executed. As soon as the hand comes into view this superschema can be interrupted in favour of triggering the superschema of moving the hand toward the object. This then gives the superschema of hand-sucking a value towards the superschema of grabbing a seen object. To consolidate the behaviour this will need to be compiled into a new superschema once it has been acquired.

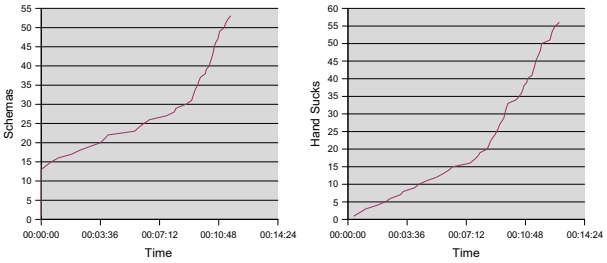


Figure 4: The number of schemas added to the library, and the number of successful hand sucks, over time.

Figure 4 shows some results form the simulation; the two graphs indicating (left) how many schemas have been learnt over time, and (right) the number of successful hand sucks. One can see that the graphs get steeper as time passes; this is because after a certain amount of time there are more schemas in the library; these correspond to various movements which will achieve some of the target schemas; this means that when a random movement is taken, it is more likely to connect with an existing schema, and result in the successful completion of a target, and the addition of a new schema to the library.

## 5. Conclusions and Future Work

We have presented a simple computational model of Piaget's theory in the early sensorimotor stages; despite its simplicity, it is arguably the most detailed existing model of these sub-stages. This simply goes to show how little work has been done in this area so far. Though our implementation only covers the beginnings of Piaget's sensorimotor stages (and in a rather simple way), it has already revealed some interesting insights into how relatively recent AI techniques (reinforcement learning with options) can effectively model some of the development described by Piaget. To compare our modelling approach with Drescher (1991), we have introduced new techniques in our model whenever it was required to allow the model to achieve the next acquisition on the development path described by Piaget. In contrast, Drescher introduced new techniques (such as the synthetic item) when he wanted his model to display a particular competence which Piaget's infants' had displayed, but Drescher did not attempt to get his model to follow the sequence of acquisitions which Piaget described as having led to that competence. Our rationale for pursuing our approach has been primarily our interest in modelling the development mechanism, rather than a particular behaviour or competence. Since Piaget's theory does not provide sufficient constraints for modelling, we want to use the development paths he described to constrain the

possible models.

To take our model further, to sensorimotor stages 3 and 4, we need to firstly enhance the representation used for schemas. Schemas at stage 3 are more advanced because the infant begins taking an interest in phenomena in the external world (such as swinging objects), rather than simply taking things to his mouth to suck. Currently we record static sensor values in a schema's $S1$, this we will need to extend to be able to represent trajectories of motion, and shapes of objects. In order for schemas to *assimilate*, it will be necessary to see analogies between similar trajectories and shapes. Whatever decisions we make about the appropriate representations must be made with a view to accounting for the transition from stage 3 to 4. Piaget's description of the development path here is sparse, so that we can foresee a variety of ways of fleshing out the details. Rather than choosing one that seems reasonable to the adult mind, we would like to test some hypotheses with studies of infants. For example, for a stage 4 behaviour such as "removing the obstacle" Piaget's account describes how the stage 3 "striking" schema is borrowed as a means; however, when striking doesn't work the infant is able to borrow other actions to displace the obstacle, and the origin of these is not clear. Clarifying these issues may require intense infancy studies. In general, in future work we hope to get a more precise account of cognitive development by alternating between computational studies and psychological studies of infants.

This research endeavour is different from much work in AI in that it is not attempting to achieve or model a particular competence, but rather to model a development path. Our motivation for doing this is our acceptance of Piaget's main thesis: i.e., that there is an invariant functional component to intelligence, which is at work at all stages, at all ages. Thus by attempting to model a development path we are forcing ourselves to model this invariant constructivist mechanism, and to gain insight into that mechanism. If we accept the constructivist viewpoint, then the learning mechanism is most important, and the necessary structures for a particular competence at a particular stage can easily be built by the mechanism. By focusing on modelling a development path we hope not to fall in the trap which much of AI has succumbed to; that is the trap of coming up with a system which performs very well on a very specific task, but does not generalise to other tasks, and seems to give little insight on the generic algorithms and representations which would be required to tackle generic tasks.

**Acknowledgements:** Thanks to Ruth Woods, Nir Oren, Joey Lam and the anonymous reviewers for comments.

## References

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47:139–159.

Chang, Y.-H., Cohen, P. R., Morrison, C. T., Amant, R. S., and Beal, C. R. (2006). Piagetian adaptation meets image schemas: The jean system. In *SAB*, pages 369–380.

Cohen, L. B. and Cashon, C. H. (2003). Infant perception and cognition. In Lerner, R., Easterbrooks, A., and Mistry, J., editors, *Comprehensive handbook of psychology. Volume 6, Developmental Psychology. II. Infancy*, pages 65–89. New York: Wiley and Sons.

Drescher, G. L. (1991). *Made-Up Minds, A Constructivist Approach to Artificial Intelligence*. MIT Press.

Haith, M. M. (1998). Who put the cog in infant cognition: Is rich interpretation too costly? *Infant Behavior and Development*, 21:167–179.

Holmes, M. and Isbell, C. (2005). Schema Learning: Experience-based Construction of Predictive Action Models. In *Advances in Neural Information Processing Systems (NIPS) 17*, pages 585–562.

Mugan and Kuipers (2007). Learning distinctions and rules in a continuous world through active exploration. In *Proceedings of the Seventh International Conference on Epigenetic Robotics*.

Parisi, D. and Schlesinger, M. (2002). Artificial life and piaget. *Cognitive Development*, 17:1301–1321.

Perotto, F., Buisson, J., and Alvares, L. (2007). Constructivist anticipatory learning mechanism (CALM): Dealing with partially deterministic and partially observable environments. In *Proceedings of the Seventh International Conference on Epigenetic Robotics*.

Piaget, J. (1936). *The Origins of Intelligence in Children*. London: Routledge & Kegan Paul. (French version published in 1936, translation by Margaret Cook published 1952).

Piaget, J. (1937). *The Construction of Reality in the Child*. London: Routledge & Kegan Paul. (French version published in 1937, translation by Margaret Cook published 1955).

Stojanov, G., Bozinovski, S., and Trajkovski, G. (1997). Interactionist-expectative view on agency and learning. *Math. Comp. Sim.*, 44(3):295–310.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.