PITFALLS IN THE MODELING OF DEVELOPMENTAL SYSTEMS[a]

FREDERICK SHIC[†] and BRIAN SCASSELLATI[‡]
Yale Social Robotics Laboratory, Yale University, 51 Prospect Street,
New Haven, CT 06511, US
frederick.shic@yale.edu[†]
scaz@cs.yale.edu[‡]

Computational models of development aim to describe the mechanisms that underlie the acquisition of new skills or the emergence of new capabilities. The strength of a model is judged by both its ability to explain the phenomena in question as well as its ability to generate new hypotheses, generalize to new situations, and provide a unifying conceptual framework. Although often constructed using traditional engineering methodologies, evaluating the performance of a computational model of development in terms of traditional perspectives is a flawed approach. This paper addresses the fundamental issues that confound quantitative analysis of computational models of developmental systems. In particular we focus on the following recommendations: 1) don't equate the success of a developmental model with its peak performance at some task; 2) don't employ purely subjective or vague measures of model fitness; and 3) don't hide or reject variation as found in the computational model. Along the way, we discuss the aspects of computational models of development that lead to the requirements for specialized methods of analysis.

Keywords: autonomous mental development, metrics, developmental systems, computational modeling

1. Introduction

Recently, many researchers in robotics have begun moving away from constrained systems tuned for specific domains and towards more general systems capable of operating under a wide range of conditions and environments. This trend has been motivated largely by a growing body of scientific work that links the flexible nature of biological systems with the interactions between developmental processes and environmental concerns. Notably, the emerging field of autonomous mental development[2] seeks to bring ideas from basic research in neuroscience and developmental psychology to robotics by situating robots in a dynamic real-world environment while equipping these robots with only the most parsimonious set of self-scaffolding guiding principles. It is hoped that robots reared under such conditions will develop the capacity to operate generally, without the need for task specification and without the need for environmental constraints. It is this promise of open-ended incremental improvement that has lead to a great deal of interest in developmental robotics[3,4,5].

[a] This paper is a revised version of a paper originally published and presented at the International Joint Conference on Neural Networks in Vancouver, B.C. 2006[1].

The mapping from biological and psychological research to robotics, however, is often difficult. The extraction of key concepts and basic principles from biology and the translation of these ideas into software is a process that is not wholly natural, partly due to the discrepancies between cells and silicon, and partly due to the ambiguity and multi-faceted nature of the biological developmental process. The conversion process is aided, however, by the use of modeling. The computational modeling of developmental processes treats aspects of the development system as a metaphor, and, within this metaphor, finds relations and links between the interacting components. By employing computational models in the investigation of developmental processes, it is thus possible to gain insight into the underlying principles that govern the process, and it is this insight that can form the basis for autonomous mental development. Given that there exists utility in the creation and analysis of these developmental models, however, one is faced with the task of evaluating these models.

At first glance, it seems that those who employ computational models to describe developmental phenomena belong in the same camp as those who want to build better computational systems using inspiration from developmental biology. In both cases, the goal is the creation of a computational framework, simulation, or mathematical formulation. In both groups, researchers are intimately concerned with the fundamental mechanisms driving developmental processes. However, the aims of these two groups could not be more dissimilar: one group uses biological themes to develop better engineering; the other uses computational techniques to formulate better descriptions of biological development. This paper focuses on those who seek to illuminate biological development through the use of computational modeling, and discusses the issues that arise when these systems are evaluated in traditional engineering terms.

Computational models are employed in all areas where human developmental progression can be tracked. For instance, computational models of development are used to model observed patterns of word pronunciation[6], the development of auto-associative memory[7], the advent of numerical perception[8,9], the incorporation of objects into categories[10], and the progression of motor skills necessary for coordinated reaching and pointing[11]. These models of development are not only found in many different domains, but are also employed in a wide range of applications including models of the learning of sets of skills[12], investigations into the mechanisms underlying the development of cognitive capabilities[13], and comparisons of populations involving atypical developmental progression[14,15]. These models employ subsystems and components that range from the most elementary, as is found in studies that examine the interaction of simulated neurons in human memory[16], to the exceedingly complex, as is found in models of social behavior and reasoning[17].

The diversity of domain, application, and technique confounds the formulation of a single, unified, globally applicable method for evaluating the fitness of a computational model of development. Like metrics for measuring the intelligence of artificial computational systems[18,19], the characterization of a developmental model must take into account multiple dimensions that often reflect very specific aspects of the model's purpose and role. Despite their apparent heterogeneity, however, computational models of developmental systems share several commonalities. First and foremost, developmental models generate multiple results that must be matched in a temporal progression. That is, these models do not operate along a simple binary axis of “correct” or “not correct”, but rather progress from milestone to milestone.

Also, like most models, the utility of a computational model of development is connected with its ability to accurately represent the phenomena in question. In contrast to models found in most pattern-recognition applications, however, the peak performance of developmental models is not of prime importance. This is not to say performance is unimportant, as a model that is intended to represent the development of some learned skill should ultimately produce measurable improvements in performance. However, it is the relationship of learning, the progression of developmental trends, and the interactions of the components of the model that are of primary interest.

Given that pure performance is not the measure by which a computational developmental model should be evaluated, the question arises: what should the measure be? As mentioned earlier, due to the diversity of developmental modeling, this is not a question that can be answered simply and neatly. In many ways, it makes more sense to address the question in the negative: what are some common errors and misconceptions in the evaluation of computational models of development? or, more simply, how should we not evaluate developmental systems?

This paper seeks to provide guidelines as to how developmental mechanisms may be compared and quantified, and how common pitfalls may be avoided. In order to properly phrase our goals, in Section II, we first discuss the difficulties that arise when we try to take a conventional engineering metric of performance, such as a task-based performance measure, and attempt to apply it as the sole measure of the worthiness of a developmental model. We continue, in Section III, with a discussion on how computational models of development require rigorous methods for analysis as compared to subjective, vague, or superficial measures of success. In Section IV, we show how the variance of computational models of development, in terms of its emergent behavior, is actually an advantage, and not a problematic deficit that needs to be hidden or rejected. We then conclude with a summary of the main points of this paper, together with a discussion on the utility, purpose, and ultimate role of computational models of development in practical application as well as in theoretical investigation. As we proceed, we will give simple toy examples that are illustrative of our major points.

2. The Problem with Peak Task Performance

In many pattern recognition and machine learning applications we are interested in training a computational model to best perform some specific task. For instance, a face recognition system used in biometric authentication can be evaluated solely on its ability to accurately recognize specific faces. However, the developmental time course of face recognition, as recorded by psychological and psychophysical experiments in neonates, infants, children, and adults, is much more complex than the mere fact that recognition can occur at some maximal accuracy. Evaluating a computational model of a developmental system in terms of peak task performance misses all the complexities of the underlying developmental process.

This brings us to our first problem with employing peak task performance as the single measure of how well a developmental model performs: it neglects the time-varying aspects of development. For instance, for a developmental model of syllable phoneme segmentation, we are not interested in how many phonemes a computational model can discriminate after being trained with a corpus of examples, but whether the number of phonemes recognizable as a function of training time resembles, say, a logistic growth function, thereby having the capacity to mimic the developmental function as found in human children[20].

A second problem is that choosing a specific task as representative of a developmental process neglects other related milestones and events that may be of even greater interest. In the human developmental time course of face recognition, for example, the increasing accuracy of recognition as a function of exposure to faces is only one aspect of the phenomenon. A computational developmental model of face recognition should explain not only how accuracy improves with age, but also should remark upon how face processing skills progress from a general sensitivity to face-like configurations found shortly after birth[21,22], to a preference for the mother's face at 1 month of age[23], to the ability to discriminate between familiar and unfamiliar individuals by 3 months[24], and so on, up towards adult levels of face recognition performance. Choosing one particular measure of performance binds us to one particular interpretation of success; this, in turn, blinds us to deficits of omission.

A third problem is that the assumption of a specific task neglects the complexity of the real-world environment. This leads to at least three problems: 1) a defined task measure fails to represent domains where the task is generated internally or is defined implicitly; 2) it provides no motivation for learning and no grounding for development; 3) it drastically underestimates the difficulty of the domain. By framing the problem within traditional machine learning paradigms, we assume that the problem is self-contained, or contained within a small, compact, controllable domain. Nothing could be farther from the truth: human development occurs far from a vacuum and a positive trend in the analysis of models of development is the use of computational agents that are explicitly embedded in complex environments[25]. If we limit ourselves to performance measures that are tightly coupled to a particular representation of the problem, we limit the generalizability of our results and the power of our implications. If we instead link the task to some representation of the world, we ground the developmental process in question to some concrete foundation. This, in turn, allows for a direct investigation of the interplay that occurs between an individual and his environment. In addition, in traditional applications, functional considerations, such as performance degradation in the presence of noise, under environmental variations, and under increasing demands are typically secondary to the question: how well does it work? In developmental models, however, such consideration are vital, as they describe how the emergence of new skills can arise in a robust fashion—a requirement for computational models of development that are biologically-relevant, as opposed to those that are simply biologically-inspired.

2.1. *Example – Face recognition*

As a simple toy example highlighting the aforementioned problems we train a small face recognition system to show how developmental milestones within a computational trend may be isolated. We are interested in this system because the developmental progression of infant face recognition is particular well studied. Our network learns to recognize faces specifically (i.e. identifying the individuals as a particular individual), faces generally (i.e. as belonging to the general class of faces, but not corresponding to a known individual (e.g. a stranger)), and non-faces (drawn from various locations in a scene containing no people).

We take from the UMIST face database[26] a selection of 6 cropped faces in black-and-white, with each face presented from 19 different viewpoints. These face images are filtered with a Laplacian-of-Gaussian filter (σ=5 pixels), cropped to square dimensions, and downsampled by nearest neighbor interpolation to a 10x10 grid of intensities. Similarly, we take one scene from the Caltech Office Database[27] and extract 100 random square regions within this scene (side length randomly drawn from a range 16 pixels to 160 pixels). These regions are converted to black and white, and then filtered and downsampled in the same manner as the faces are.

Three individuals are selected as faces to be recognized specifically, three faces are selected to comprise the general face class, and the non-face class is populated by random sampling from the office scene. A simple two-layer neural network (one input layer, one hidden layer, and one output layer) is created which takes the downsampled image pixels as input, in a manner similar to [28]. The input layer is fully connected to a hidden layer consisting of 3 hidden tansig nodes. This hidden layer is connected to the output layer which consists of 4 logsig nodes, one for each of three face targets, and one for the general class of faces.

The network is trained by adaptive gradient descent (repeated batch random presentation of face images and non-face images) and the resultant learning curve, as a function of mean squared error of target outputs, is shown in Fig. 1. Fig. 1. also highlights some developmental milestones in the course of network learning. These milestones are judged to have occurred when network performance on the corresponding test dataset has a sensitivity and specificity of over 70%, and does not drop below this bar for the remainder of the learning.

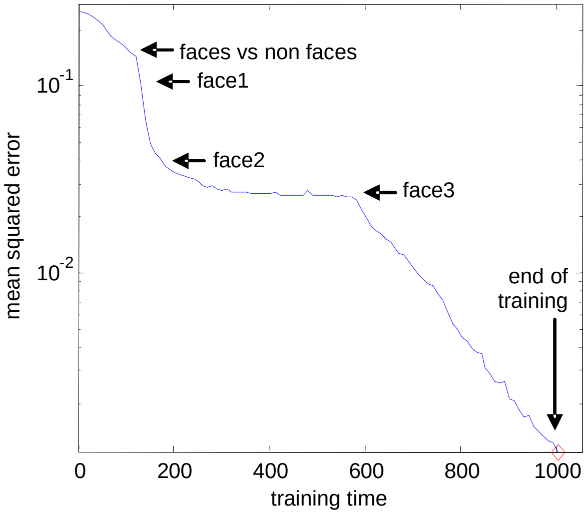


Fig. 1. Training curve of a neural network learning to simultaneously recognize the class of faces (faces vs non faces) as well as 3 specific individual faces (face1, face2, and face3). The x-axis is the number of training rounds, whereas the y-axis is the mean-squared error of the recognition tasks. Milestones, as evidenced by behavior on a separate testing dataset, occur at various points on the recognition learning curve. Whereas a traditional engineering task would only be interested in the end point of development (diamond, marked "end of training"), a developmental model is interested in the entire curve, including milestones along the developmental path.

If we were to view this task as a pure engineering problem, i.e. a recognition problem with a clearly defined accuracy metric, we would in essence be following the route of large scale studies such as [29,30]. We would only be interested in the end point of development. With a developmental model we are interested not only in the shape of the learning process, but also with milestones that correspond to activity on the learning curve. Furthermore, the developmental systems approach can ask additional questions regarding the nature of face representation: does there exist some specific pattern of feature or pattern recognition within the hidden layer of the network?; is there a pattern of learning consistent with N-shaped or U-shaped development as noted in developing infants[31]?; does this developmental approach towards face learning offer advantages in generalizability, such as was shown in Nagai et al[32] for joint attention representation? A purely engineering approach misses the underlying complexity of the developmental process.

3. Choosing The Right Quantitative Measure

In order to be able to compare a computational model with the developmental reality, we need to find some appropriate metric for measuring the distance between the predicted phenomena and the biological progression. In many cases, this entails characterizing behavior. This is often difficult because behavior is itself often an emergent property of the underlying physical or neurophysiological scaffolding: it does not lie on the measurable axes of the system, it lies on top. Because of this, specific measures that pinpoint the phenomena in question must be developed, tested, and deployed. This process of finding the best set of measurements and metrics is plagued with many potential pitfalls. These pitfalls include an over-attachment to trivial computational effects and the over-reliance upon subjective or superficial measures.

When building a computational model of development, it is often far too easy to develop a simple representation of some particular phenomenon, grab the quickest and nearest pattern recognition system, train the system, and present the resultant learning curves as evidence for a developmental trend. The fundamental problem caused by building developmental models this way is that it transforms the rich, complex tapestry of human development into a dull, one-dimensional string that is almost trivial.

The most basic aspect of all machine learning systems is that they learn, adapt, and specialize. If we are interested in how humans learn to discriminate between images of oranges from images of apples, and we frame this problem as an error-minimization task, we should not be surprised that our ability to identify oranges increases as we apply non-linear gradient descent. Similarly, if we frame object recognition as a constrained clustering problem operating over silhouette histogram statistics, we should not be surprised that objects that cast the same shadow also gravitate towards the same clusters. A temporal progression leading to greater efficiency is the most basic aspect of developmental processes; its presence in a computational model of development is the lowest bar that must be reached for suitable discourse to begin. In other words, computational learning as a mechanism leading to a temporal progression mimicking development is only interesting in its own right when the observed effects are non-trivial.

A related trap that often turns a computational model of a developmental system into a trivial experiment is caused by under-constraining the parameters of the developmental model. For example, the complexity of even the simplest models of cognitive processes can be staggering: a basic model of visual attention (e.g. [33]) uses hundreds of potential parameters. In these cases, it is easy to step back and explain any resultant discrepancies between a computational model and physiological reality by a hand-waving argument involving the adjustment of any number of possible parameters. Unfortunately, the same argument that makes a model theoretically match a particular observed effect is also the same argument that makes a model theoretically match any observed effect, or nothing at all[34]. It is far more productive to start a model with reasonable parameters, adjust these parameters sparingly, and discuss how the parameters interact.

In addition, it is possible to over-constrain a simple computational model by linking together modules that are tightly coupled in the interface between effect and prerequisite. Such systems are in effect large scale cause-and-effect chains, which only serve to highlight the inevitable conclusion unless the mechanisms underlying the predicate expansions are transparent and become the true subject of investigation. For example, consider the ability for a 2 year old child to point to objects when they are named (e.g. a child may point to a toy boat when his mother says, “boat”). A simple robotic model of this would require 1) word recognition; 2) object recognition; and 3) the ability to map visual scenes to motor coordinates. If we trained a system to map spoken words to visual signatures of objects, then trained a system to map locations in the visual scene to motor coordinates, we should expect that, given a proper reward structure, the robot can learn to point to an object when the object is named. Here the scaffolding may be of developmental interest; so too may the on-line interactions and adaptations of cognitive intermediates be worthy of attention. The fact that learning is achieved should not be the primary result.

Sweeping generalizations and overly simplistic or superficial analyses are, of course, not limited to computational models of development: they can infect any computational model used for behavioral analysis or reproduction of biological action. As we are well-prepared to anthropomorphize even simple geometric shapes[35], our minds also readily ascribe a label of “biologically related” to a large class of complex stimuli. In other words, it is all too easy to assign life-like attributes to a computational system that generates behavior of sufficient complexity. However, the mere impression of biological relevancy does us little good. For a computational model to be useful, it must be able to generate predictions or further some particular hypothesis. For a developmental computational model to find relevancy, there must exist some quantifiable means of comparing the model to behavioral reality.

It is a natural reaction to believe that a computational model matches biological reality when some measurable surface characteristic of the computational model behaves in some biologically plausible fashion. The difficulty, however, is that many such comparisons can be highly superficial. For instance, consider the development of fine motor skills necessary to accomplish some task. A computational hypothesis on the development of these skills could be that a child is basically an adult with sub-adult accuracy. One approach to building a computational model of increasingly accurate motor control would be reinforcement learning. Suppose we are able to train such a model, obtain a desired motor behavior, and subsequently are interested in presenting our reinforcement strategy as a good model of motor development. First, we show that the final task performance is good; however this, as mentioned in the previous section, is expected. Next, we show that performance increases over time; but this too is also only a minimum requirement and not in itself sufficient to warrant adjudicating a developmental model successful. We are left, then, with comparing our computational motion trajectories with the motion trajectories of human subjects. However, a simple measure based on the distances between joints or end-effectors ends up with a definition of proximity that is too strict: it overestimates the “distance” between two trajectories when the Euclidean distance at some point in time is large, but the distance in terms of intent, mechanism, or encapsulated behavior, is small.

A measure of the applicability of some developmental or biological model to reality should factor in the key components and factors that affect the process in question. This is necessary because otherwise any trends that do match physical evidence will match only phenomenologically. If we are only interested in the surface characteristics, we don't actually need a computational model at all. It is more useful to incorporate the forces that are known to be biologically or psychologically significant, and to thereby be able to investigate the relationship of these forces in framing the actual development or behavior, than it is to have a model that accurately characterizes a trend over a time-frame, but has no basis in deeper mechanism.

3.1. *Example – Comparing eye fixations*

As an example, consider the comparison of real and artificial eye movements in Fig. 2. The left image of Fig. 2 is the actual eye trajectory of a human subject viewing some scene. We can take this eye trajectory and use it to build a probability map by placing a Gaussian at every fixation point and normalizing. This probability map can form the basis of a simple computational model for synthesizing a sequence of artificial eye movements. At each time step, a random location of fixation is chosen from the underlying probability distribution (right image of Fig. 2). However, this model suffers from several flaws. First, the total Euclidean distance between the human gaze trajectory and the trajectory generated computationally can be arbitrarily large. Second, the computational model is also terrible at representing many characteristics of human eye trajectories, and notably lacks fixations and saccades.

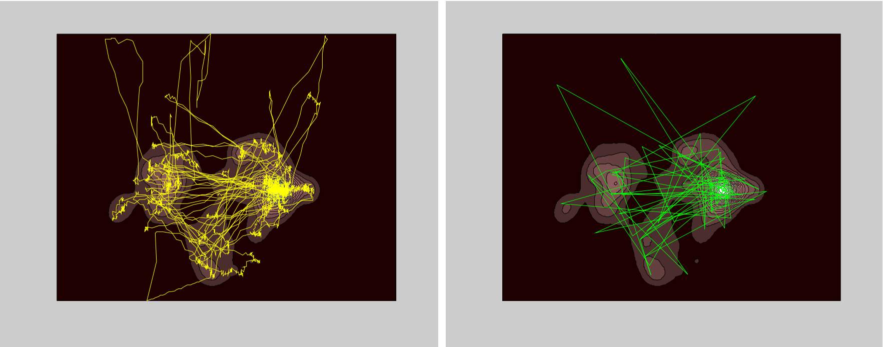


Fig. 2. Human gaze data (left) and a trajectory drawn probabilistically from an approximation to the underlying density (right). Note that whereas saccades and fixations are identifiable in the left image, these properties do not exist in the right image.

The major weakness of the above model, however, is not that the distance between computed fixation points and real fixation points is large, nor is the major problem that the computational model so blatantly does not produce coherent time-varying action. The major weakness is that it ignores the fact that the fixations of an individual will depend highly on the scene itself (see [36]). Employing Euclidean distance as a sole measure of the similarity of two gaze trajectories completely misses this point. Similarly, the use of a probability density function is also inappropriate, as it implicitly incorporates Euclidean distance as the basis for its comparison. Fig. 3 illustrates this point further. In very simple cases where there is only one single salient region, distance makes sense as a measure of how close a fixation is with another fixation (Fig. 3, left). However, in a more realistic case, employing Euclidean distance as the basis of fixation similarity would fail completely (Fig. 3, right). That is, if the implicit goal of the observer is to look at the eyes of individuals in the scene, focusing on the eye of the left face or the right face is equally valid, yet the distance between the eyes of the two faces could be arbitrarily distant.

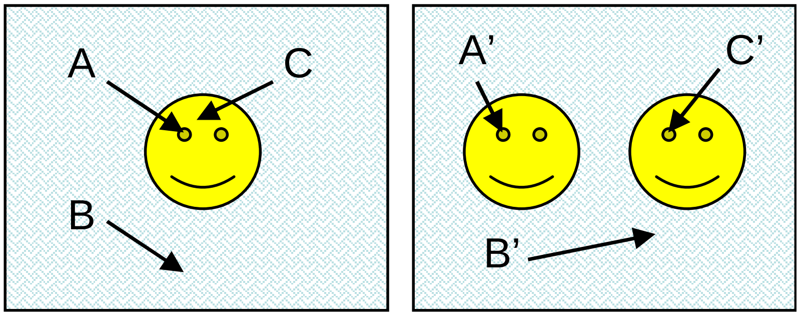


Fig. 3. Problems with Fixation Distance Metrics for Measuring Similarity. In the case on the left, if some model picks point A and another model picks point B, we could safely say that these two models are dissimilar. Conversely, if one model picks A and another model picks C, we could say that the models are similar. In this case, a distance metric based on distance between fixations makes sense. In the case on the right, the Euclidean distance between A' and B' is roughly equivalent to the distance between A' and C'. A distance measure of similarity would gauge a model choosing C' and a model choosing B' as being equally similar to a model choosing A'. Based on underlying features, however, a model choosing C' would be much more similar to a model choosing A'. In this case, using a fixation distance metric does not make sense. By employing distance metrics between points of fixation, we ignore the underlying substrate of visual attention: that of features of the scene itself.

The myriad ways of looking at the similarity between a computational model and the physiological gold standard (for eye gaze: as Euclidean distances, as a density function, as a series of saccades and fixations, and as an operation over underlying features) point towards the need to obtain a relevant measure of how a computational model is performing. If some aspect of the developmental model is of interest, a measure must be explicitly assigned. On the bright side, if the aspect is able to be tracked accurately, and if the measures are chosen appropriately, we are able to obtain statistics regarding the performance of a model that are meaningful to development (e.g. see [37]).

For the eye gaze comparison example, we cannot measure our success by the end-point of behavior by calculating Euclidean distances between fixations. This measure is a reflection of a process, and it is this process, not its realization, in which we are interested. Likewise, we cannot measure our success by the input to our system, the entirety of the visual scene, because this interpretation precedes any interesting processing. Our metric for similarity must lie someplace between these two extremes. For example, we could assume that the proper level of comparison is at the level of the local features centered at the points corresponding to fixations. Our measures can neither be too loose, where they degenerate to hand-waving, nor can they be too tight, where they bring us the inevitably expected goal: in either case we are doomed to succeed.

Similarly, our models cannot be so vague that they can incorporate any effect, leading us to the unsatisfying but inevitable statement that the desired trend could be shown if only some variables were constrained. Such models can, in fact, be fit with some effort; they perform randomly when presented with new data. Our models cannot be so specific, so completely integrated with the environment, that all the conclusions fall down like a line of dominos. Often in these situations it is instructive to return to basics and ask: can this experiment fail? If the answer is no, then there is no experiment.

Judging whether a developmental model has failed, of course, is its own problem. Just as individuals are found along a wide range of physical and personal characteristics, so too can their developmental progression vary. However, in contrast to the almost useless fact that developmental systems improve, the notion that developmental systems vary turns out to be crucial in the evaluation of a computational model of a developmental system.

4. Retaining Variation

In most applications, variation of performance is seen as a negative factor, often being viewed as system unreliability or instability. But for computational models of developmental systems, retaining and employing variation is both positive and practical. Developmental milestones rarely happen with clockwork precision. For example, children begin babbling between 3 and 8 months of age, begin using single words at around 12 months, develop a small vocabulary of single words between the ages of 15 months and 18 months, and begin using simple phrases between 18 and 24 months[38]. Children who do not follow this progression are at risk for problems such as developmental or language delay. However, a great deal of variation occurs in practice, since, as language acquisition is a combination of both innate capability and environmental exposure[39], its emergence as a distinct capability in children reflects the interaction of multiple cognitive and muscular subsystems[40].

The key point is that, for the progression of skills on the time-course of human development, some variation is expected. This leads to some greater flexibility in computational modeling, as events are not bound by some strict schedule. However, this also leads to greater demands, as the source of the variations must be explained in a way that is rigorous. For developmental milestones we are not as interested in exact times of appearance as we are in preserving a certain order of skill emergence. While some skills are bounded naturally (for instance the ability to speak simple phrases cannot occur before the ability to speak simple words), other skills are not, especially when comparing across modalities (e.g. speech capability versus motor coordination). We can then explore the interaction of previously developed capabilities in providing a scaffold for new abilities.

One particular aspect of the developmental progression that should be defined and quantified, however, is the sources of developmental variation. Factors leading to variability in skill onset should be phrased in terms of intrinsic stochastic mechanisms in cognitive development, such as neurogenesis, the growth of dendritic branches, or inherent cortical plasticity, or in terms of extrinsic environmental variability, such as limited exposure to, say English vocabulary words in a Spanish-speaking household. In addition, the interactions between these factors, which are often the cornerstone of a particular computational investigation, should be formulated in such a way that the cascading effects of variability of intrinsic and extrinsic factors on the stochastic schedule of skill emergence can be examined.

In line with having a variable basis, computational models of development are not always expected to work. This is not to say that a model should produce gibberish or nonsensical results, but that the failure of a model to maintain some typically developing structure is useful. One of the most fruitful uses of computational modeling in developmental psychology is in the analysis of developmental pathology or atypical development. This is why apparent failures in a computational model of a developmental system should neither be ignored nor swept under the rug: an apparent failure signals either a true flaw in the model, which must be addressed, or a possible mechanism for arrested or abnormal development. Similarly, it is a failure of the model when the model succeeds where humans would fail (for example a module for tracking the development of face recognition through adolescence should exhibit well-known visual-system peculiarities, such as decreased ability to recognize inverted faces). For these reasons, the parameters on which a computational model of development is built should be varied, the systems involved should be stressed, and the failures of the models should be carefully examined.

Another form of variation that may appear is the incidental milestone: sometimes a developmental effect emerges as the result of the particular computational model employed. Typically employed as an interesting aside, the emergence of behaviors that are unexpected from a computational model, and not related to any explicit encoding, are the best evidence that a particular implementation or developmental simulation achieves a level of performance exceeding expectation. Unexpected behaviors that do not correlate with biologically observed phenomena should also be reported. These are indications that remaining work needs to be accomplished, either in the form of the reassessment of assumptions and formulations, or in the form of further investigations and experiments.

Finally, one of advantages of having a computational model is that the model should be executable multiple times. The aggregation of a series of simulations should be able to lead to statistics regarding the frequency, ordering, and distributions of emergent skills. By integrating across multiple runs, we can examine the variability in onset of one particular skill versus another, and all computational attributes against the true biological ground truth. In this manner, the aspects of behavioral comparisons seen previously as confounds to analysis and quantification, can be brought into line with rigorous metrics.

4.1. *Example – Locomotive development*

Conventional wisdom holds fast to the cliché: you must crawl before you walk. However, this is not actually true: roughly five percent of infants begin walking without any previous crawling[41]. A computational model that aims to describe the developmental progression of locomotion from birth should be able to characterize this variation as well as the general time course of emergent behavior. As an example we will consider a simple dynamical systems model of locomotion. Note that this example serves purely in a didactic capacity and is not necessarily intended to be representative of any serious investigation (a serious investigation should at least incorporate some aspects of the work by Thelen and Smith[42]).

We begin by assuming that the development of locomotion can be described by two variables: arm locomotion (a) and leg locomotion (g). These two motor capabilities represent the abstract concept of a maturing musculoskeletal system and a developing neurological motor coordinating capability. For this example, we hand-constructed a set of equations that generate a vector field which produces interesting results (in other words, these equations are not intended to be illustrative only, and not based on any empirical observation or published data). The equations for that field are:

$$\begin{aligned} \frac{da}{dt} &= 4a - g - \frac{28}{3}ag + 1 \\ \frac{dg}{dt} &= \frac{1}{3}(40g^3 - 66g^2 + 23g + 3) \end{aligned} \qquad (1)$$

The developmental progression of a single individual is a discrete random walk (100 steps) over this field. da/dt and dg/dt together define the center angle αc and corresponding magnitude rc. The actual movement at each time step is a step of size 0.018rc in a random direction drawn from a normal distribution centered at αc with standard deviation equal to 45˚/rc. This gives us a trajectory across arm and leg locomotion space (Fig. 4).

Since locomotive development of each individual is determined by probability, each run of the model will generate slightly different results. As in the true biological reality, variation occurs. This variation, however, is stable: all trajectories start at the point where locomotive capability is completely undeveloped and stop in a basin of attraction where development ends. This global behavior is reliable and reproducible, even though the individual path is not. Consistent with evolutionary theory, those behaviors and biological components that are vital to the fitness of an organism must contain stabilizing machinery.

Furthermore, we can take this model and apply to it a second level of analysis. We add a single Boolean variable, s, which governs the expression of locomotion. No locomotion is observable unless s is true. s begins as false, and, once it switches on, it remains in that state for the remainder of the experiment. The decision to switch on is determined stochastically with a probability of 7% at each time step. This process represents the idea held by some that though developmental progress is being made internally, and that though the capability to move exists, the first initiation of motion is essentially a stochastic decision process. In Fig. 4, the dotted points on the trajectory represent the points where locomotion is expressed.

We can further divide the space of locomotion into subregions, with each subregion corresponding to a particular behavior. For example, in Fig. 4, the bottom left region marked in hatching represents the area of precrawling, which includes crawling where the belly remains on the floor; the surrounding left part of the space represents the area where arm motion and leg motion contribute equally to locomotive efforts, leading to true crawling (with the belly off the floor); the remaining right side of the space is when leg action begins to dominate and walking occurs.

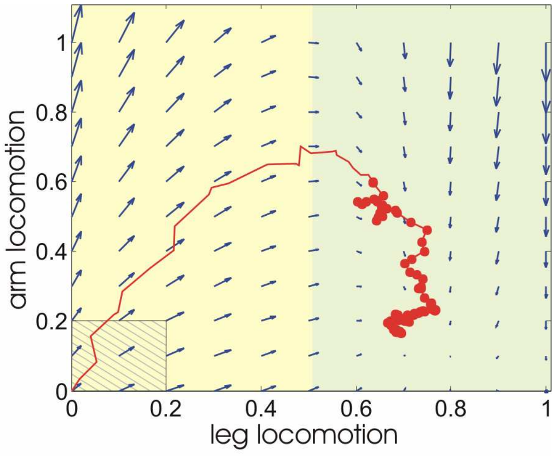


Fig. 4. A dynamical system as a developmental model of locomotion. Current development, in terms of legs and arms, are points in the figure (e.g. (0,0) corresponds to the point where no part of locomotion is developed, (1,0) corresponds to the point where only the legs are developed). An individual's developmental progression is represented as a curve emanating from the origin. Dots on the trajectory represent locations where expression of locomotion occurs (see text). The arrows represent the most likely direction in which development will proceed at any given time, with the magnitude corresponding to the relative speed of the transition. The locomotion space is separated into discrete behaviors, with the small hatched region in the lower left the precrawling behavior, the left side of the space the crawling behavior, and the right side of the space walking.

By aggregating the results of 1,000,000 independent trials, we can examine the statistics regarding when certain behaviors arise. These results are shown in Fig. 5A for true crawling (25.9±5.8 weeks) and Fig. 5B for walking (45.1±5.7 weeks). These values are in line with evidence from child development and psychology[38]. However, much more information can be extracted from the developmental model beyond simple means and standard deviations. We can determine, for instance, that the incidence of walking without prior crawling is 7.7%; for these individuals the mean age for walking is 49.1±13.4 weeks (Fig. 5C). We can note that the associated distribution for these individuals is highly non-normal and skewed to the right, implying that the vast majority of these walkers who never crawl are simply late expressers for motility (a notion that would, naturally, need to be verified by further experimentation). By examining the delay between walking and crawling for individuals that do both crawl and walk, we can calculate an average delay of 18.8±6.9 weeks (Fig. 5D) between crawling and walking based on distributional differences. Though this example is obviously a very simple toy example, the analyses performed highlight the potential types of information that can be collected in a computational model of a developmental system, turning the inherent variance of the system into both additional statistics for analyses and novel hypotheses for further investigation. In this particular example, we could continue by examining the rate of regression from walking back to crawling, the final distribution of walking expertise, capacity to walk versus expression of walking, and so on. Not only does the variation of the model paint a richer portrait of the developmental scene, it also provides additional measures that can be independently evaluated and compared against empirical results.

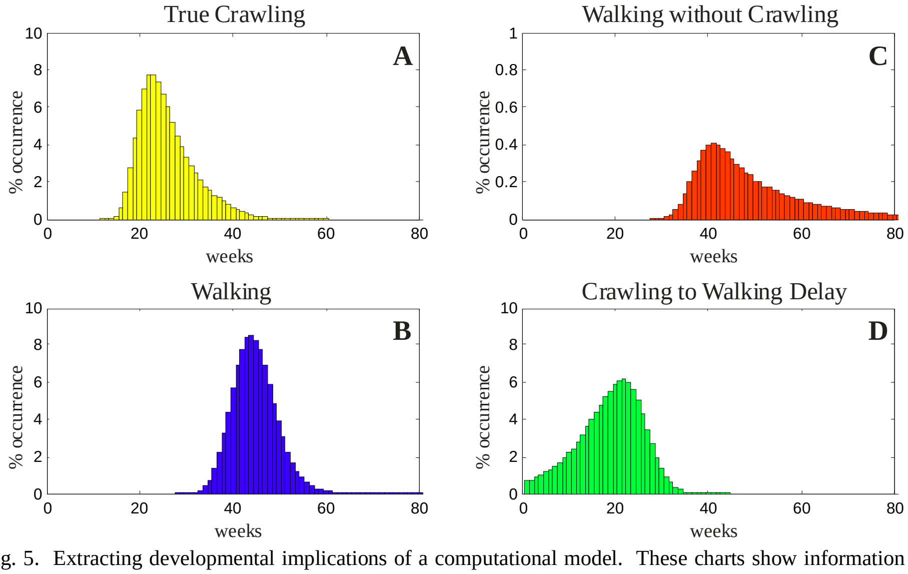


Fig. 5. Extracting developmental implications of a computational model. These charts show information that can be extracted from a computational model of walking and crawling development. Plot A shows the times at which the crawling behavior emerged (25.9±5.8 weeks) in a population; Plot B shows the distribution for walking (45.1±5.7 weeks). In Plot C, which focuses on the 7.7% of the population that walked before crawling, the mean age for walking is 49.1±13.4 weeks. The distribution of times between walking and crawling is

shown in Plot D (18.8±6.9 weeks). The variation of the system provides a rich description of the progression of developmental milestones.

5. Discussion

Computational models of developmental systems are powerful tools in the analysis of not only typical cycles of maturation, but also of alternate pathways leading to abnormal development. The specific purposes of models of development also lead to specific prerequisites for interpretable results. In this paper we have discussed how traditional methods for evaluating machine learning systems are not wholly applicable to developmental systems in three ways (Fig. 6). (1) Pure task-based measure of performance fail to capture the hallmark of a developmental process: the developmental time-course. We have also shown how binding our performance evaluation to a specific task leads to ignoring other interesting subtasks and related developmental milestones, as well as the interaction between the subject being modeled and enclosing context of the environment. (2) Finding the correct quantitative measures for the evaluating the performance of a developmental model is critical for generating relevant results. We have shown that employing the wrong measure leads to subjective, overly simplistic, or superficial measures of performance and that these types of measures are ultimately harmful to the power of conclusions that can be drawn for a computational model. Furthermore, we have shown that in order to avoid triviality, a computational model must provide greater insight than the zero-order measure of task performance (i.e. the model learns to perform the task reasonably well), but also the first-order measure of learning (i.e. performance changes during training). (3) Variation is the strength of a developmental model. The variations can be collected into distributions for measures of interest, many of which can be independently compared and verified against empirical results. The underlying uncertainties of the model, the unreliability of obtaining convergent results, and the errors and quirks of simulation are disadvantages for a system measured strictly by performance, but are advantageous for systems investigating developmental trends.

In traditional engineering…	**In models of development…**
…we optimize for peak task performance	…the journey there matters.
…the metrics for evaluation are often obvious	…the obvious metrics are often wrong.
…consistency and reproducibility are ideals.	…instability, variation, and failures are more interesting than success.

Fig. 6. Comparison of traditional engineering and modeling of development.

The techniques and applications discussed within this paper are especially applicable to robotic models of development and robotic systems that employ developmental components. Robotic implementations with developmental aspects have additional advantages in ecological validity: robots operating in the real world can incorporate real-world patterns of noise and unpredictability, can host practical factors such as transmission delay and signal degradation, and can explore interactions with humans in interfaces ranging from real-time socially-contingent response to proprioceptive reflex. Robotic systems offer a platform by which the full complexities of the natural human environment can be brought to bear on a developmental question. We should note, however, that, for robots intended to perform engineering tasks, engineering measures of performance are still the most critical measures, even if these tasks rely on developmental components. Indeed, engineering techniques and developmental techniques are not mutually exclusive, as good engineering takes into account the context of the solution, and developmental systems require excellent engineering tools in design and formulation. It is the over-reliance or the sole interpretation of developmental systems through an engineering mindset that proves insufficient.

Though no clear, universal metrics for the evaluation of computational models of development exist, we hope that this paper has brought light to some of the issues relevant to the evaluation of developmental systems. And though this paper has been presented in the negative, as methods and techniques and approaches that should not be employed in building computational models of developmental systems, we hope that it has illuminated also the way these systems should be examined.

6. References

1. F. Shic and B. Scassellati, How Not to Evaluate a Developmental System, in *International Joint Conference on Neural Networks (IJCNN)*, Vancouver, B.C., (July 2006), pp. 10674-10681.
2. J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur and E. Thelen, Autonomous mental development by robots and animals, *Science* **291**(5504) (2001) 599–600.
3. M. Lungarella, G. Metta, R. Pfeifer and G. Sandini, Developmental robotics: a survey, *Connection Science* **15**(4) (2003) 151-190.
4. M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, Cognitive Developmental Robotics As a New Paradigm for the Design of Humanoid Robots, *Robotics and Autonomous Systems* (**37**) (2001) 185-193.
5. M. Asada, K. Hosoda, Y. Kunioshi, H. Ishiguro, T. Inui, and K. Nishimura, Synergistic Intelligence: Approach to Human Intelligence Through Understanding and Design of Cognitive Development, in Fifth International Conference on Developmental and Learning (ICDL), (May 2006).
6. M. S. Seidenberg and J. L. McClelland, A distributed, developmental model of word recognition and naming, *Psychological Review* **96**(4) (1989) 523-568.
7. D. S. Rizzuto and M. J. Kahana, An Autoassociative Neural Network Model of Paired-Associate Learning, *Neural Computation* **13**(9) (2001) 2075-2092.

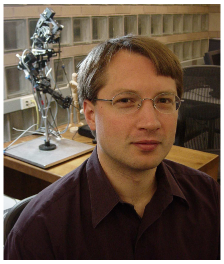
8. S. A. Peterson and T. J. Simon, Computational Evidence for the Subitizing Phenomenon as an Emergent Property of the Human Cognitive Architecture, *Cognitive Science: A Multidisciplinary Journal* **24**(1) (2000) 93-122.
9. T. J. Simon, Computational evidence for the foundations of numerical competence, *Developmental Science* **1**(1) (1998) 71-78.
10. D. Mareschal, R. M. French, and P. C. Quinn, A connectionist account of asymmetric category learning in early infancy, *Developmental Psychology*, **36**(5) (2000) 635-45.
11. M. Schlesinger, D. Parisi, and J. Langer, Learning to reach by constraining the movement search space, *Developmental Science* **3**(1) (2000) 67-80.
12. R. Sun, E. Merrill, and T. Peterson, From implicit skills to explicit knowledge: a bottom-up model of skill learning, *Cognitive Science: A Multidisciplinary Journal* **25**(2) (2001) 203-244.
13. Y. Munakata, Computational cognitive neuroscience of early memory development, *Developmental Review* **24**(1) (2004) 133-153.
14. C. O'Laughlin and P. Thagard, Autism and Coherence: A Computational Model, *Mind and Language*, **15**(4) (2000) 375-392.
15. E. Carlson and J. Triesch, A Computational Model of the Emergence of Gaze Following, *Progress in Neural Processing*, **15** (2004) 105-114.
16. J. J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *PNAS* **79** (1982) 2554-2558.
17. N. J. Vriend, An Illustration of the Essential Difference between Individual and Social Learning, and its Consequences for Computational Analyses, *Journal of economic dynamics & control*, **24**(1) (2000) 1-19.
18. S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edn. (Prentice-Hall, New Jersey, 2002).
19. J. Weng, Muddy Tasks and the Necessity of Autonomous Mental Development, in *Proc. 2005 AAAI Spring Symposiums, Developmental Robotics Symposium*, Stanford University, Palo Alto, CA, (2005).
20. B. Fox and D. K. Routh, Analyzing spoken language into words, syllables, and phonomes: A developmental study," *Journal of Psycholinguistic Research* **4**(4) (1975) 331 – 342.
21. E. Valenza, F. Simion, V. M. Cassia, and C. Umilta, Face Preference at Birth, *Journal of Experimental Psychology Human Perception and Performance*, **22**(4) (1996) 892-904.
22. F. Simion, V. M. Cassia, C. Turati, and E. Valenza, The Origins of Face Perception: Specific versus Non-Specific Mechanisms, *Infant and Child Development*, **10**(1) (2001) 59-66.
23. J. Bartrip, J. Morton, and S. De Schonen, Responses to mother's face in 3-week to 5-month-old infants, *British Journal of Developmental Psychology*, **19**(2) (2001) 219-232.
24. M. de Haan, M. H. Johnson, D. Maurer, and D. I. Perrett, Recognition of individual faces and average face prototypes by 1- and 3-month-old infants, *Cognitive Development* **16**(2) (2001) 659-678.
25. M. Schlesinger and D. Parisi, The agent-based approach: A new direction for computational models of development, *Developmental Review* **21** (2001) 121-146.
26. D. B. Graham and N. M. Allinson, "Characterizing Virtual Eigensignatures for General Purpose Face Recognition," in H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie and T. S. Huang (eds), *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, **163** (1998) pp. 446-456.
27. M. Fink, The Full Images for Natural Knowledge Caltech Office DB, *Technical Report Caltech [CSTR:2003.008a]*, California Institute of Technology, Pasadena, CA (2003)

28. H. A. Rowley, S. Baluja, and T. Kanade, Neural Network-Based Face Detection, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1996), pp. 203-208.
29. P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, Face Recognition Vendor Test 2002: Evaluation Report, in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures* (2003), p. 44.
30. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, Overview of the Face Recognition Grand Challenge, in *IEEE Conference on Computer Vision and Pattern Recognition* (2005).
31. C. Cashon and L. B. Cohen, Beyond U-Shaped Development in Infants' Processing of Faces: An Information-Processing Account, *Journal of Cognition and Development*, **5**(1) (2004), pp. 59-80.
32. Y. Nagai, M. Asada, and K. Hosoda, Developmental Learning Model for Joint Attention, In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), pp. 932-937.
33. L. Itti, C. Koch, and E. Niebur, Model of Saliency-Based Visual Attention for Rapid Scene Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(11) (1998) 1254-1259.
34. K. R. Popper, *The Logic of Scientific Discovery* (Basic Books, New York, 1959).
35. F. Heider and M. Simmel, An Experimental Study of Apparent Behavior, *The American Journal of Psychology*, **57**(2) (1944) 243-259.
36. F. Shic and B. Scassellati, A Behavioral Analysis of Computational Models of Visual Attention, *International Journal of Computer Vision* (in press).
37. F. Shic, W. Jones, A. Klin, and B. Scassellati, Swimming in the Underlying Stream: Computational Models of Gaze in a Comparative Behavioral Analysis of Autism, in *28th Annual Conference of the Cognitive Science Society*, Vancouver, B.C., (July 2006), pp. 780-785.
38. S. P. Shelov and R. E. Hannemann (eds). *Caring for Your Baby and Young Child: Birth to Age 5* (Bantam, New York, 1993).
39. E. Bates, P. Dale, and D. Thal, Individual differences and their implications for theories of language development, in P. Fletcher & B. MacWhinney, Eds., *Handbook of Child Language*, (Basil Blackwell, Oxford, 1995), pp. 96–151.
40. E. Bates, Plasticity, localization and language development, in S. Broman & J.M. Fletcher, Eds., *The changing nervous system: Neuro-behavioral consequences of early brain disorders*, (Oxford University Press, New York, 1999), pp. 214-253.
41. "Crawling", [Online document], [2006 Feb 15], Available at HTTP: http://www.healthofchildren.com/C/Crawling.html
42. E. Thelen and L. B. Smith, *A Dynamic Systems Approach to the Development of Cognition and Action*, (MIT Press, Cambridge, MA, 1994).

Acknowledgements

The authors would like to thank the anonymous reviewers whose comments helped to clarify and improve this paper. Support for this work was provided by a National Science Foundation CAREER award (#0238334) and award #0534610 (Quantitative Measures of social response in autism). Some parts of the architecture used in this work was constructed under NSF grants #0205542 (ITR: A Framework for Rapid Development of Reliable Robotics Software) and #0209122 (ITR: Dance, a Programming Language for the Control of Humanoid Robots). This research was supported in part by a grant of computer software from QNX Software Systems Ltd.

Frederick Shic received his B.S. degree in Engineering and Applied Science from the California Institute of Technology in 1996 and his M.S. degree in Computer Science from Yale University in 2004. From 1996 to 1999 Mr. Shic was a software engineer with the Sony Interactive Studios of America. From 2000 to 2002 he was an active researcher with the Huntington Medical Research Institutes and the Rudi Schulte Research Institutes, focusing on the application of magnetic resonance spectroscopy to human neurodegenerative diseases. Currently, Mr. Shic is a Ph.D. candidate with the Yale Social Robotics Laboratory at Yale University in New Haven, CT. His interests include computational models of visual attention, the modeling of developmental systems, and autism spectrum disorders.



Brian Scassellati is an Associate Professor of Computer Science at Yale University. He received his Ph.D. in Computer Science and Electrical Engineering from the Massachusetts Institute of Technology in 2001. His research focuses on using social robots to evaluate models of how infants acquire social skills and to assist in the diagnosis and quantification of disorders of social development (such as autism).