Minds & Machines (2009) 19:93–115
DOI 10.1007/s11023-008-9127-1

Task Muddiness, Intelligence Metrics, and the Necessity of Autonomous Mental Development

Juyang Weng

Received: 28 April 2007 / Accepted: 4 November 2008 / Published online: 11 December 2008

Abstract This paper introduces a concept called *task muddiness* as a metric for higher intelligence. Task muddiness is meant to be inclusive and expendable in nature. The intelligence required to execute a task is measured by the composite muddiness of the task described by multiple muddiness factors. The composite muddiness explains why many challenging tasks are muddy and why autonomous mental development is necessary for muddy tasks. It facilitates better understanding of intelligence, what the human adult mind can do, and how to build a machine to acquire higher intelligence. The task-muddiness indicates a major reason why a higher biological mind is autonomously developed from autonomous, simple-to-complex experience. The paper also discusses some key concepts that are necessary for understanding the mind and intelligence, such as intelligence metrics, the mode a task is conveyed to the task executor, a human and a machine being a joint task performer in the traditional artificial intelligence (AI), a developmental agent (human or machine) being a sole task performer, and the need for autonomy in task-nonexplicit learning.

Keywords Intelligence metrics · Human intelligence · Turing test · Artificial intelligence methodology · Physical development · Mental development · Sensors · Effectors · Internal representation · Vision · Audition · Languages · Actions

Introduction

Despite the power of modern computers, we have seen a paradoxical picture of artificial intelligence (AI): Computers have done very well in areas that are typically

J. Weng (✉)
Michigan State University, East Lansing, MI 48824, USA
e-mail: weng@msu.edu

considered very difficult (by humans), such as playing simulated chess games. However, they have done poorly in areas that are commonly considered easy for humans, such as vision, audition, and natural language understanding. On the other hand, there is a lack of appreciation of what the human mind does on a daily basis, from one task to the next, partially due to a lack of general framework for measuring tasks.

There have been numerous studies on the measurement of the intelligence of AI systems. The "imitation game," proposed by Turing (1950), now known as the Turing Test, greatly influenced the ways machine intelligence was studied. The limitation of such a symbolic text-based test has now been better recognized, e.g., Michie (1993) and Norman (1991). The proposed *Total Turing Test* (Russell and Norvig 2003) includes computer vision to perceive objects and robotics to manipulate objects and move about. The National Institute of Standards and Technology has been sponsoring the Workshops on Measuring the Performance and Intelligence of Systems, known as PerMIS, held annually since 2000 (Meystel and Messina 2000), where many proposed metrics are application-specific and, thus, lack the applicability to a wide variety of tasks. There have been some studies on the procedures for evaluating research articles in AI, e.g., (Cohen and Howe 1988), but not for evaluating tasks.

Intelligence quotient (IQ) and Emotional Intelligence (Goleman 1995) have been proposed to measure human intelligence. The typical tests in the field of psychometrics concentrate on the differentiation of human individuals in a human age group, which consists of normally developing, physically healthy human individuals. These tests are very useful for not only measuring the normal development of a normal child, in contrast with, e.g., an autistic child, but also for understanding what traditional robots lack. However, tests in psychometrics are not designed for measuring tasks. Gardner (1993) proposed the concept of "multiple intelligences," in the sense that human intelligence is displayed not only in logical-mathematical reasoning or emotional aspects, but also through other aspects such as bodily kinesthetic and spatial skills. Giulio Tononi and Gerald Edelman proposed to use mutual information to measure the complexity of integrated biological neural systems (Tononi and Edelman 1998). Nevertheless, these studies do not provide a mechanism for evaluating how tasks are across many tasks.

A task performer is called agent, regardless whether it is biological, artificial, or mixed. The latter mixed type is becoming more and more important because of a growing field called bioengineering or biomedical engineering. Biological networks have been grown in Petri dishes and humans have been assisted by various types of artificial organs, limbs, and aids (Enderle et al. 2005; Brooks 2002).

The term "muddy" is used to refer to tasks that are not "clean." In this paper, a composite muddiness is proposed, which contains an open number of axes of muddiness (factors), each measuring a different characteristic of a given task. It is overly simplistic to enforce independence among these axes, but each axis should measure a different muddiness characteristic. We do not require each axis to represent the same "level" of information, since this "simple-minded" requirement is counter productive. Some tasks are somewhat general—they can be also called "problems," but we use the term "task" for consistency.

Five muddiness categories have been identified in this paper so that all of the muddiness factors fall into these five categories. Alternative to other alternatives that have been proposed (e.g., see an excellent survey by Russell and Norvig (2003)) we discuss the composite muddiness as a performance metric for intelligence, both natural and artificial. Based on the muddiness discussed here, this paper outlines three categories of tasks, Category 1 Clean Tasks; Category 2 Muddy Tasks, and Category 3 Very Muddy Tasks. The task muddiness theory explains why the traditional artificial machines perform the tasks in Category 1 well, but not for Category 2, and perform worse for tasks in Category 3.

The implications of the task muddiness discussed here are likely to be multifold. (1) The task muddiness concept facilitates the appreciation and understanding of the end-to-end nature of biological intelligence: from the raw sensors end all the way to the motors end. Why is this end-to-end nature, namely, sensory and motor experience, important to intelligence? Is there any totally disembodied intelligence in nature? (2) When a task-specific software executes a task (e.g., the Microsoft Word program executes a word processing task), is the program the only executor of the task or the human programmer and the program act together as a joint task executor? When a human programmer designs a task-specific representation (e.g., the column width and line spacing), into a program, should such representational intelligence attribute to the human designer's intelligence or the machine's? Does the machine understand this task-specific representation? When a human hand-writes a paragraph on a piece of paper, does he understand the concepts such as column width and line spacing? (3) What are the typical modes of communication through which a task is conveyed to the task executor? This question directly points the sharp contrast between how a task-specific machine learns a task-specific skill with programmed-in task representation and how humans acquire mental skills through a autonomous developmental process. All these and other questions boil down to the issue of mental development by humans and animals.

Computational autonomous mental development (CAMD) is a relatively new approach to intelligence (biological and artificial) that has attracted an increasing amount of research activities. Although developmental psychology and developmental neuroscience have been well recognized scientific discipline, computational modeling of mental development was not raised explicitly until recently (Elman et al. 1997; Weng et al. 2001). Computational models of mental development, especially those that cross a wide variety of space scales (cellular, cortical, corticocortical and brain scales) and time scales, are necessary for understanding how the mind works.

A fundamental characteristic of CAMD is the computational modeling for agents to learn tasks that are unknown during the conception time (i.e., formation of zygote for animals and the programming time for robots) (Weng et al. 2001). Therefore, the developmental learning agent (humans or robots) must internally (i.e., within the brain) generate *representations* guided by a new kind of program called a *developmental program*. For biological organisms, this developmental program[1]

[1] Biologists and neuroscientists have no problem with calling it a program. Read, e.g., Reik and Dean (2002) and Sur and Rubersein (2005).

is represented by the genome in the zygote. For artificial machines, this developmental program can be directly designed (Weng et al. 2001; Weng and Hwang 2006) or artificially evolved.[2]

In biological development, two types of development are involved, physical and mental. Through the process of physical development, a single-cell zygote is developed into a fetus, a newborn baby, and an adult through interactions with the environment, which provides the necessary physical conditions (e.g., temperature), nutrition, sensory and motor experiences, etc. The older theory of preformation holds that the gametes containing small but perfectly formed bodies waiting to grow. This primitive theory has long been passed by the epigenetic theory of development, which is supported by much recent evidence in developmental biology. The theory of epigenesis explains that differences in cells and tissues arise in development because gene-expression programs change as cells differentiate and, furthermore, the actually occurred change depends on not only the gene-expression, but also the cell's environment, which includes the physical environment and other cells. For example, during the growth of neurons, electrical activities can trigger molecular or developmental programs that create connections, shape particular connections and modify the connection strength (e.g., see a review by Sur and Rubenstein (2005)).

The brain develops along with the body development. The mind is what the brain does. Therefore, the mental development characterizes the functional development of the brain. It takes place in parallel with, and is greatly shaped by, the physical development (including the development of the brain), the activities of the body (including the brain), and the environment of the agent. Through mental development, the agent incrementally learns to perform increasingly more sophisticated tasks through interactions with environments, using the agent's sensors and effectors and the mental skills that it has learned earlier in similar, but typically different settings.

The field of developmental robotics aims to simulate biological mental development, not necessarily including the biological physical development because designing and fabricating a robot body by human engineers has long become practical. The various modes of learning for a developmental machine are very similar to that of human developmental learning (Weng et al. 2006). Therefore, neural science, psychology, AI, and robotics face many common research issues under the subject of autonomous mental development. Their advances can benefit greatly through multidisciplinary communications and collaborations. The material in the following sections may help to understand why the mind develops—many human daily tasks seem to be too muddy for nature to choose the route outlined by the preformation theory.

Section 2 discusses the basic principles that motivated the muddiness concept. Section 3 provides examples of muddiness factors, which naturally calls for the framework of multiple muddiness presented in Sect. 4. Section 5 introduces the

[2] The reader is referred to a special issue on autonomous mental development in the vol. 11, no. 2 issue of the IEEE Transactions on Evolutionary Computation, guest-edited by Jay McClelland, Kim Plunkett and Juyang Weng.

composite muddiness. Section 6 uses the muddiness concept to introduce three categories of AI tasks. Section 5 discusses the composite muddiness as a metric for intelligence. Section 8 contains some concluding remarks.

Principles of Task Muddiness

Designing a metric for evaluation of tasks is challenging. As discussed above, no comprehensive metric previously existed. However, the issue of task evaluation is important to understanding the mind as well as the tasks that the mind can process.

In order to understand the proposed muddiness measure for tasks, we need to look into some fundamental principles that are related to evaluating a task.

Characteristics of Muddiness

The concept of the proposed muddiness was motivated by the following considerations.

1. Across task domains. Muddiness can incorporate any task. For example, a computer chess-playing task can be compared with a face recognition task, in an intuitive way.
2. Independent of species or technology level. A task that is muddy for dog is also muddy for humans. A task that is muddy for today's computer technology remains muddy for future technology, no matter how advanced computer technology becomes.
3. Independent of the performer. A task that is muddy for machines is also muddy for humans and vise versa. However, humans are good at performing muddy tasks.
4. Quantifiable intuitively. It helps us to understand why a task is intrinsically difficult in a quantitative way. It is impractical to expect that a measure of this grand scale is totally quantifiable so that every task can be compared in concrete numbers. This is because tasks that are interesting to consider for our purpose are not those that give detailed specifications (e.g., the number of receptors in a human's retinas or the number of pixels in a robot's cameras). It is more useful to examine task muddiness intuitively and conceptually without requiring mapping each muddiness factor to a concrete number. That is, we use the intuition provided by simple algebra, without forcing us down to the calculation detail of arithmetics.
5. Amenable to evaluating state-of-the-art intelligent machines and to appreciate what humans can do. It objectively measures the overall capacity requirement of tasks faced by humans and machines.
6. Indicative of human intelligence. It enables us to fully appreciate human intelligence along multiple dimensions.

Before we are able to discuss muddiness, we first consider an animal or a robot as an agent.

An Animal or Robot as an Agent

Systematically modeling any intelligent being as an agent was an important conceptual advance of the theory of AI. By definition, an agent is something that senses and acts.[3] An illustration is shown in Fig. 1. The input to the agent is what it senses from its *external environment* and the output from the agent is the action that it applies to the external environment. The device that the agent uses to sense input from the environment is called a sensor. The device that the agent uses to deliver the output to the external environment is called an effector. For example, our eyes are visual sensors used for sensing visual information from the external environment (e.g., watching a movie). Our hands are manipulatory effectors that deliver our manipulatory actions to the external environment (e.g., picking up a pen from a table).

The above agent senses only the external environment and acts on only the external environment. By *external*, we mean the environment outside the "brain." The body of the agent is considered external. The brain (or the central nervous system) senses its internal environment (the brain itself) and acts on it. For example, a human has a new idea and he acts on the internal environment so that he concentrates on the new idea instead of the music that is playing in his external environment. The skill of controlling internal actions is an emergent capability acquired from autonomous mental development. Weng (2004) formulated what is called a Self-Aware Self-Effecting (SASE) agent model, which senses and acts on its internal (brain) environment in addition to sensing and acting on the external environment. The SASE model indicates that autonomous mental development develops complex representations in the internal environment in order to perform muddy tasks.

A Human Engineer as an Agent Constructor

An agent is used to perform single, multiple, or an open number of tasks. Depending on how a task is specified, a task can be a subtask of another more complex task. For example, making a move is a subtask of playing a game of chess, and playing a game of chess is a subtask of participating in a chess tournament.

Suppose that we are given a task to be performed by a machine. Here, we need to distinguish to whom the task is given. Is it given directly to a machine or to a human engineer who fabricates the machine and writes the programs for it? We consider that a task is given to a human being who constructs and writes programs for the machine, which executes the task. Therefore, two phases are involved: the developmental phase and the performance phase, as illustrated in Fig. 2. In the developmental phase, a human engineer accepts a task that the machine is supposed to perform. He understands and analyzes the task before constructing an agent (machine) that is supposed to perform the task. Therefore, the product of the developmental phase is an agent. In the performance phase, the agent is put into

[3] See, e.g., an excellent textbook by Russell and Norvig (2003) and an excellent survey by Franklin and Graesser (1997).

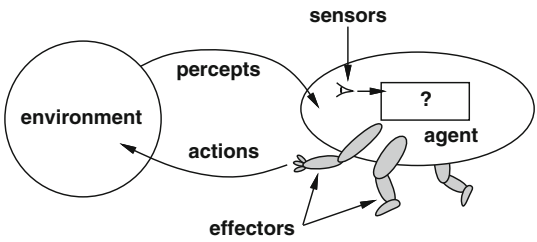

Fig. 1 The abstract model of an agent, which perceives the external environment and acts on it (adapted from Russell and Norvig 1995)

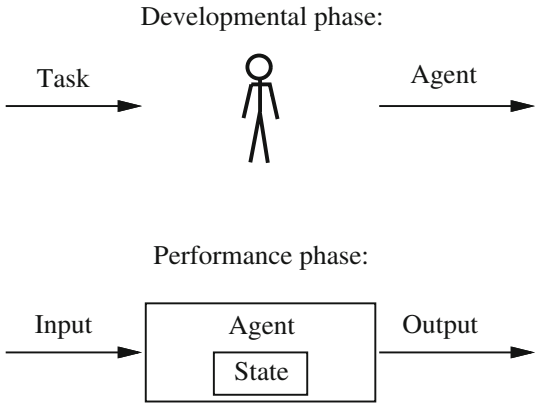

Fig. 2 The *manual* developmental phase and the *automatic* performance phase. The developmental phase is not automated. The performance phase is partially or fully automated. The agent may have its internal state (e.g., firing patterns of neurons) when it performs a task in the performance phase

operation. It accepts an input and produces an output. Through this process, the agent performs an instance of the task. It may accept another input and produce an output for each instance. This way, the agent can perform more instances of the same task. This is the traditional *approach of manual development*.

In contrast, according to the approach of autonomous development, called the developmental approach (Weng et al. 2001), the tasks that a machine is supposed to perform are unknown during the time of programming. Therefore, understanding the tasks to be performed is not the responsibility of the human engineer, but the developmental machine itself. This is consistent with human learning: it is the human learner who is responsible for understanding the tasks being learned, not his parents.

Whether a human can produce a successful agent for a given task depends on (1) how muddy the given task is and (2) how he constructs the agent. We will devote much of the remaining part of this paper to the first issue as the second issue is addressed elsewhere, e.g., (Weng et al. 2001).

The Constructors of a Human Agent

Then, who is the constructor of a human agent? One may say, well, his parents.

In a sense, it is true. The two parents gave birth to their child. However, modern biology has provided deeper insight into this important issue: biologically, each of the two parents provided only a cell called a gamete (egg or sperm). The two gametes are combined to result in a single cell—a fertilized egg—whose entire set

of genes (i.e., genome) contains the complete information necessary for development from the single cell to a normal human adult having about 60 trillion cells of different types in a working configuration. We will call this genome *the developmental program*. As we know it, the constructor of this developmental program is mainly evolution, where not only the parents, but also many ancestors and the environments in which many generations have lived in have played compounding roles.

However, this miracle of development, from a single cell to a functional normal adult, can take place successfully only if the environment provides necessary biological conditions and information. The biological conditions include all the post-conception biological conditions necessary for development, such as nutrients and temperature. The information includes all necessary environmental information for normal mental development, such as parenting, school teaching, and social interactions.

In summary, at least three constructors are responsible for the construction of a human agent: the biological parents, the human genome, and the human environment in which the agent grows up. These three constructors are not necessarily disjoint. For example, the biological parents may be part of the environment for training the child, but this does not have to be the case without exception (e.g., children raised in an adopted home).

Constructing a human agent is not that difficult for his parents, as the evolution has got all the mechanisms ready, biological and environmental. The same is not true for a machine agent. Even with modern computers, constructing a machine agent for many muddy tasks is extremely difficult. In the following sections, we study task muddiness to see why.

In the following, the term "programmer" for a machine agent means a computer programmer. The term "programmer" for a human agent means the process of human evolution, including the two human parents of the human agent. Whenever a factor of task muddiness is discussed, both human and machine agents are applicable.

Muddiness Frame Examples

It is not beneficial to put the muddiness of a task into a single abstract measure that is arbitrarily defined. Any task can be positioned in a muddiness frame to allow a visualization of how muddy this task is compared with other tasks. The muddiness frame is like a coordinate system that we use to specify a point. Each axis represents a factor of muddiness.

Let us first consider two such muddiness factors: *the rawness of input* and *the size of input*.

If the input to a machine is edited by a human being, the input rawness is low (e.g., computer chess playing and text-based language processing). If the input is directly from a sensor without human editing, the rawness is high (e.g., visual recognition and sonar-based navigation).

The input space is a space that contains all of the possible inputs. The size of the input space, or the *size of input* for short, indicates the number of possible different values that the agent has to consider while performing the task. For a symbolic input where each frame is an alphanumeric input from A to Z followed by 0 to 9, its input size is $26 + 10 = 36$. For a vector input of dimension d whose each component takes m different values, the size of input is m^d. For example, an image of $d = 240 \times 320 = 76800$ pixels, with each pixel taking a byte from 0 to 255, the size of input is 256^{76800}, an astronomical number.

The muddiness frame using only these two muddiness factors is shown in Fig. 3. Some typical tasks are positioned in this frame. The direction of each axis denotes the direction of increase in the corresponding muddiness factor. Since the meaning of each muddiness factor is not simple, it is not useful to assign a concrete number to each class of tasks. Thus, we should interpret the coordinates of these tasks qualitatively instead of quantitatively.

The next factor to introduce is the richness of the goal of a given task, or *richness of goal* for short. This refers to how difficult it is to describe the goal of the task in well-established mathematical terminology. We insist on mathematical terminology since it is a concise and precise way of expressing programs. A task that can be fully described in mathematical terms can be converted into an algorithm with little ambiguity. Conversely, a program can always be written in mathematical terminology. We also insist that the description of the goal of the task must be in terms of the input of the system since information available to the agent is only from its input when it performs the task.

Consider playing a computer chess game. The goal of the task is to checkmate your opponent's king. One can use mathematical terminology to describe this condition. Thus, the richness of this goal is low.

Next, consider identification of humans from video images, a task of visual recognition. A series of questions are raised if you attempt to describe this task in terms of input to the system. What do you mean by humans? How do you describe an image that contains a human and one that does not? More questions need to be asked before one can construct a machine to perform the task. You will quickly realize that it is almost impossible to describe this task in mathematical terminology based on only image input. You probably can describe a human face well in terms of common sense, but you cannot precisely describe a human face in terms of image input.

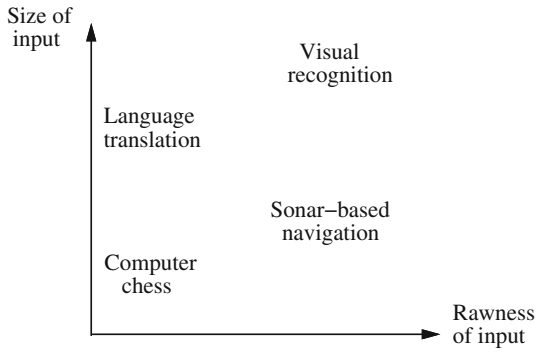

Fig. 3 A muddiness frame for two muddiness factors: rawness of input and size of input. This diagram is for conceptual visualization only

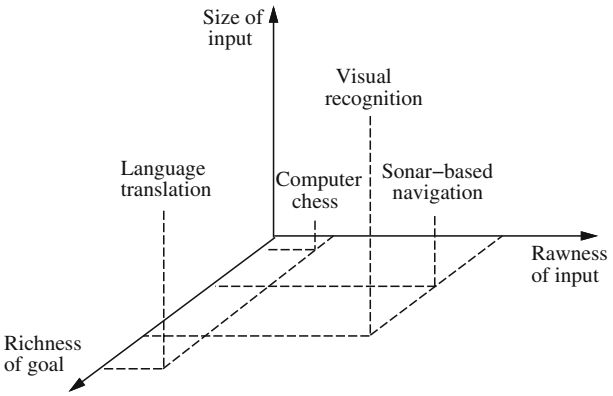

Fig. 4 A muddiness frame for three muddiness factors

Take language translation as another example. It is almost impossible to write down mathematically the goal of translation of an article based on the text input. What do you mean by translating well? What do you mean by "meaning" in mathematical terminology? Many more questions must be asked before one can construct a translation machine. Thus, it is extremely difficult to express the goal of translation in mathematical terminology. Thus, the richness of this goal is high.

Augmenting the previous muddiness frame by adding the richness of the goal, we have the muddiness frame shown in Fig. 4. We should not consider such positions for these tasks as absolute. A particular task arising in actual application can vary tremendously in all three muddiness factors. For instance, technical language translation from a text in a very specific domain with a very small vocabulary may not have a very high measure in richness of goal.

Muddiness Frames

In order to acquire a more complete view about the muddiness of any given task arising from the real world, we need to introduce more factors for muddiness. We divide the factors into five categories: external environment, input, internal environment, output, and goal. The external environment is the world in which the agent's brain works, including the agent's body. The input is the information that the agent's brain receives from the external environment. The internal environment is the agent's brain, which does not include the body. The output is the (motor) actions from the agent's brain. The goal is the objectives of the tasks that the agent performs, which is considered to be separated from the external environment for our purpose.

It can be seen that given a task, the five categories constitute a partition of the universe. In other words, any additional muddiness factor has a category to which it belongs. Of course, there are many other ways to partition the universe. Other partitions may not necessarily be as intuitive and concise as the one defined here for our purpose.

Table 1 A list of muddiness factors for a task

Category	Factor	Clean	Muddy
External env.	Awareness	Known	Unknown
	Complexity	Simple	Complex
	Controlledness	Controlled	Uncontrolled
	Variation	Fixed	Changing
	Foreseeability	Foreseeable	Nonforeseeable
Input	Rawness	Symbolic	Real sensor
	Size	Small	Large
	Background	None	Complex
	Variation	Simple	Complex
	Occlusion	None	Severe
	Activeness	Passive	Active
	Modality	Simple	Complex
	Multi-modality	Single	Multiple
Internal env.	Size	Small	Large
	Representation	Given	Not given
	Observability	Observable	Unobservable
	Imposability	Imposable	Nonimposable
	Time coverage	Simple	Complex
Output	Terminalness	Low	High
	Size	Small	Large
	Modality	Simple	Complex
	Multimodality	Single	Multiple
Goal	Richness	Low	High
	Variability	Fixed	Variable
	Availability	Given	Unknown
	Conveying-mode	Simple	Complex

Table 1 gives some major muddiness factors grouped into the above five categories. Although these muddiness factors superficially look ad hoc, they are not since they are natural things for a programmer to consider before constructing a machine. They are meant to be explanatory, not exhaustive.

Let us examine the additional factors of muddiness.

External Environment

Awareness. refers to the degree to which the programmer knows about the external environment in which the agent works. If he knows the environment, he can define features that the machine can use. Otherwise, recognizing objects in an unknown external environment is a much harder task to accomplish.

Complexity. measures how complex the external environment is. If the environment contains exclusively cubic blocks, the task of working in the environment is cleaner than an environment of human daily living, where very complex objects are present. Just think about what you can see from a busy street.

Controlledness. gauges the degree in which the environment is controlled. If an environment is not controlled, then the complexity of the environment is not bounded. Although a human infant does not necessarily recognize his parents, brothers, or sisters, he is exposed to their faces early on, as well as complex interactions which include their conversations, smiles, and tickles. In a controlled environment, some objects and activities are disallowed.

Variation. indicates whether the environment is changing. A fixed environment, such as a static office setting, is cleaner than a dynamic one where pages of a book can be turned, people can move around, the lighting can change, and all the furniture can be rearranged.

Foreseeability. reflects whether the future environment is foreseeable or not. A static environment is not necessarily foreseeable, if no information is given about the environment. If it is known that a car driving environment is the Mojave Desert between Barstow, California, and Primm, Nevada, the environment is partially foreseeable, but many details of the actual environment are still unknown such as other cars.

Input

We have already discussed the rawness of input and the size of input. The *background of input* indicates whether the input includes information that is not related to the task (e.g., background in a face recognition task). Also, if the input does include background, how complex is the background (e.g., uniform gray or a busy street)? The *variation of input* refers to the complexity of variation among inputs that require the same output (e.g., a static horse or a running horse, in a horse recognition task). The *occlusion of input* is another factor of muddiness. Presence of occlusion in input makes a task muddier (e.g., an occluded face in a face recognition task). The *activeness for input* indicates whether the agent must actively acquire input in order to perform the task (e.g., the robot must go to libraries to find the required information). The *modality of input* measures the complexity of the input modality. The sensory modality affects how muddy a task is. The task of accomplishing this using a laser range scanner, for example, is less muddy than the one that uses two video cameras based on stereo ranging. The *multi-modality of input* indicates how many distinct sensory modalities are used (e.g., vision alone versus vision and audition).

Internal Environment

It has been a common mistake to neglect that the internal environment is a *necessary* and *important* category for characterizing tasks. For example, few AI researchers have paid sufficient attention to a stark contrast between a human and a traditional AI system: the former autonomously generates internal task-specific

representations, but the latter requires a human programmer to design its internal task-specific representations.

It is difficult to understand the requirements of internal memory without considering a key concept called context. The need of an internal environment is determined by the need of representing a distinguishable *context state*, or often simply called *state*. It is important to note, however, that the true state of the agent is represented not only by the context state (which uses short-term memory) but also the entire memory (which includes the long-term memory). The behavior generated by the agent depends not only on the context state (e.g., hear an insult), but also the long-term memory (e.g., how he has been educated in the past). For consistency with the AI literature, we call the context state simply state.

An AMD agent is a sequential processing agent. It processes one input frame at a time and then produces one frame of control vector for its motors at a time. A sequential processing agent needs a state to identify and distinguish context. It corresponds to that part of memory that is recalled and kept active for the current step. A state indicates the current cognitive situation of the agent.

The *size of an internal environment* is the measured value of the minimally required size of the internal storage space. With all other muddiness factors fixed, translation of a page of technical writing requires less memory than translation of a page of a well-composed essay. Although the actual required internal memory size depends on the approach adopted for the agent, there is a minimally required memory size from the viewpoint of information. It is typically difficult to estimate such a minimal size. The larger the space that is minimally required to perform a task, the muddier the task. As mentioned above, each state in the state space is used to record the context that the agent must currently attend to. The space of the state must contain the information about all the contexts that must be distinguished for the task, whether directly or through interpolation or generalization. Thus, the size of state (space) indicates how many possible contexts are required and how large those contexts are for a given task. Further, the long-term memory space, in addition to the context space, indicates how much space is minimally needed to carry out the task.

The *representation of internal environment* concerns whether the internal representation is given or from the task specification to the task execution agent. It also characterizes how much information is given. The less information about internal representation is given, the muddier the task is. For this concept, the following distinction is important:

(a) A human is the sole task executor.
(b) A machine is the sole task executor.
(c) A human and a machine are combined as the task executor: The programmer programs the machine, which in term executes the task.

In cases (a) and (b), the task specification is directly conveyed to the sole task executor. In case (c), typically the task specification is conveyed to the human who in turn designs a representation for the machine.

In the current AI field, this distinction has been largely overlooked. For human intelligence, (a) is the case. In the current AI field, case (c) is prevailing since it is

the mode of traditional, manual-development AI. However, with an AMD agent, (b) is the case (Weng et al. 2001; Weng 2004). The distinction between (b) and (c) are philosophically important, and it clarifies the dilemma of the traditional AI: If a traditional non-developmental machine does a muddy task well (hardly any so far), it is the human programmer who is intelligent, not the machine. If a developmental machine does a muddy task well (more and more are expected to be demonstrated), it is truly the machine task-performer that is intelligent in the same sense as a human task-performer who is intelligent.

The *observability of internal environment* means the degree to which the internal representation of the agent is observable by the outside world during the construction (or development) of the agent. This is important since the developmental phase may include a training process during which the agent is trained before it is mature enough to do the job. During this training process, the difficulty level of the training depends, to a large degree, on whether the state of the agent at that time is observable, partially observable, or completely unobservable. Why does it matter? Let us consider how a human mother teaches her child. If the mother can observe what the child is thinking about (i.e., the current state of his brain), then she can teach the child more effectively. For example, if the mother knows that the reason why the child does not want to eat is because he wants to play with a favorite toy, then the mother can give the toy to the child while feeding him. Unfortunately, the state of the human brain is not directly observable. We can only observe the behavior of an individual. But, his behavior does not have to be consistent with what he thinks about. For example, a child can make up various excuses for his poor performance in school without revealing the real reason. If the agent's state is not observable, the human teacher does not have sufficient information about the true state of the agent during the training process and, thus, it is harder to train such an agent.

Closely related to the observability of internal environment is the *imposability of the internal environment*. The imposition here means that the human teacher directly sets the value of the internal representation of the agent. The representation of the human brain is not imposable through direct brain manipulation, assuming that brain surgery is not what we are interested in here. A parent can tell his child what the right thing to do is. However, the parent cannot directly impose what a child actually thinks about (e.g., through electric wires linking into various areas of the brain). If a human teacher can directly impose desired values to the internal representation of the agent under training, it is more effective to conduct the training, but the training is tedious. The observability and imposability of the internal environment are closely related. For example, if the internal environment is both observable and imposable, the imposed value of the state can be determined by a human according to the observed current state.

The *time coverage* of the internal state characterizes how complex the required temporal coverage pattern is for the context when the task is performed. If the temporal context required is short, the task is cleaner. If the extent of the temporal context is fixed, the task is cleaner. If the temporal context is monolithic, meaning that everything sensed in a time window can be used in the same way, the task is cleaner. A non-monolithic context means that attention selection is required to attend to only old events that are related to the task at hand, instead of putting

everything that happened within a time window into the context. In other words, long, time-varying, non-monolithic temporal context makes the task muddy. Many tasks that humans perform routinely require a temporal context that spans several days or even years. However, not everything that happens in this long time window is used in making a decision. Here are some examples. A college student works harder whenever he recalls how his parents sent him to college three years ago. But, how he works now has little to do with what he ate this morning for breakfast. A successful entrepreneur conducts his business more prudently whenever he recalls how his grandparents' business went bankrupt and how he started his new business from scratch. In these examples, the human subjects appropriately select only temporal events that are closely related to the decision they want to make. Therefore, the temporal context used by humans tends to be of a long temporal span, time-varying, and non-monolithic. In Table 1 the term "complex" is used to describe these characteristics.

Output

The agent outputs its actions to its effectors. The *terminalness of output* reflects how the output can be used directly without human processing (e.g., text versus the motor control signals). While raw input means that it does not require preprocessing by humans, terminal output means that it does not require post-processing by humans. The *size of output* is similar to the size of input. The *modality of output* determines how complex the output is (e.g., just the heading direction of a car versus controlling all the muscles in the vocal tract to speak). The *multi-modality of output* indicates how many distinct effector modalities are used (e.g., driving only versus driving and speaking concurrently).

Goal

Each task has a goal. The goal is very much related to task muddiness. We have already discussed the richness of the goal as a factor of muddiness.

The *variability of goal* indicates whether the goal of a task may change, and the degree of change. For a game of chess, the goal is fixed: to checkmate the opponent's king. However, in our daily life, our goal may change. One may change his plan to go to a gymnasium for a work-out after witnessing a hit-and-run accident. For a 3-year-old child, his goals are more or less playing, eating, and sleeping. When he becomes a college student, however, he may have very different goals. His goals may change depending on how well he does in college, the new knowledge he learns while in college, and so on.

The *availability of goal* measures whether the goal is given at the time of machine construction. One might feel puzzled: If you want to construct a machine to do jobs, you must provide the machine with a goal. This is true for simple or fixed tasks. When the tasks are complex, such as in the case of a household service robot, is the goal given? One may say, "well, the goal is to let the robot do what I say." Do you want to tell the robot what to do every time the door bell rings? Do you want to tell it what to do every time a child steps into the lawn area when the robot is

mowing? Therefore, the issue is how much detail has to be specified for a robot that does numerous tasks and whether the goal of every task needs to be given before it is constructed. A human child is born independent of whether his parents have a clear idea of what the child will do in his lifetime. The goals of doing something, including even what to do, are taught after the construction (birth) of a human being, not before. Future advanced developmental robots will be able to self-decide what to do when the detailed goal is not clearly conveyed. They will be able to show autonomous intention.

The *conveying mode* refers to the mode in which the goal is specified to the task executor. Is it explained via a keyboard in a computer language or in a spoken natural language? The former is clean and the latter is muddy. If a human is the sole task executor or a human is the programmer for a machine, conveying the task goal in a spoken natural language is not a major challenge. This means that humans can execute very muddy tasks. If a machine is the sole task executor, the machine must understand the goal of the given task in whatever conveying mode used. The more complex the mode is, the muddier the overall task.

We have finished our examples of muddiness factors. We did not intend to scrutinize every possible factor. We have passed a number of other factors without mention. Some of them can be considered a finer classification within the discussed muddiness factors. For example, noise in the input can be a measurement for input. On the other hand, rawness covers this aspect since a signal from a real sensor contains noise, which reflects the imperfection of a real sensor and its electronic parts. We did not mention the hierarchy of goals since it is considered as belonging to the richness of the goal. Therefore, one can include as many muddiness factors as needed. The list of muddiness factors in Table 1 is not meant to be exclusive. It is meant to provide enough detail for explanation.

Composite Muddiness

If we use n muddiness factors, we can construct an n-dimensional muddiness frame, similar to what is in Fig. 3 for a 2-D case and Fig. 4 for a 3-D case. From the 25 muddiness factors in Table 1, we have a 25-D muddiness frame.

A caveat here is that the muddiness measures along different axes are very different in nature and, thus, it is hard to compare different muddiness factors using the concrete values of their coordinates. We should only use the muddiness frame in an intuitive and conceptual sense. Then, why bother defining a scalar quantity if we do not compare the muddiness of different tasks quantitatively? This is because a conceptual understanding about how different factors are combined in mathematical operations is more general and more useful than comparing two concrete numbers. For a similar reason, algebra is more general than arithmetics. Two different concepts cannot be combined into a single measure until both are converted into a number by an abstract scheme.

Another caveat is that the sense of muddiness created by a muddiness frame depends very much on what kinds of muddiness factors are included in the muddiness frame.

We would like to give a composite measure in terms of how muddy a task really is. Denote the muddiness coordinate of the ith row in Table 1 as m_i. The value of m_i should never be smaller than 1, $m_i \geq 1$. Each original muddiness x_i (e.g., the size of input space or the size of internal memory) is mapped by a nonlinear function f_i, so that $m_i = f_i(x_i)$ is a properly transformed muddiness. For example, for $x_i \geq 0$, we choose

$$m_i = f_i(x_i) = \alpha_i \log_2(x_i + 1) + 1,$$

where α_i, with $0 < \alpha_i < 1$, determines the relative importance of each muddiness m_i in the following composite mudidiness.

The composite muddiness of any task can be modeled by the product of all the property transformed muddiness coordinates m_i:

$$m = m_1 m_2 \ldots m_n = \prod_{i=1}^{n} m_i, \tag{1}$$

where m is the composite muddiness and n is the number of muddiness axes adopted in a muddiness frame. Note $m_i \geq 1$, for $i = 1, 2, \ldots, n$. This way of modeling the muddiness of a task is not meant to compare relative importance of different muddiness factors on different axes of the muddiness frame.

Once the set of muddiness factors is determined, we can visualize the muddiness of a given set of tasks. For example, in a 2-D muddiness frame, we can plot an iso-muddiness curve. It is a curve on which all the tasks in the muddiness frame have the same muddiness. Figure 5 plots three iso-muddiness curves in a 2-D muddiness frame using two muddiness factors m_i and m_j. In a 3-D muddiness frame, all the tasks having the same composite muddiness value form an iso-muddiness surface. In an n-dimensional muddiness frame, they form a hyper-surface: $\prod_{i=1}^{n} m_i = c$, where c is the constant composite muddiness measure. Note that the coordinate 1 can be used for the simplest useful case, since $m_i \geq 1$.

It can be seen clearly why we did not define the composite muddiness as the Euclidean distance from the position of a given task to the origin of the muddiness frame. Our composite muddiness takes into account the composite muddiness of many axes, not just a single axis. A position that is near an axis still corresponds to a

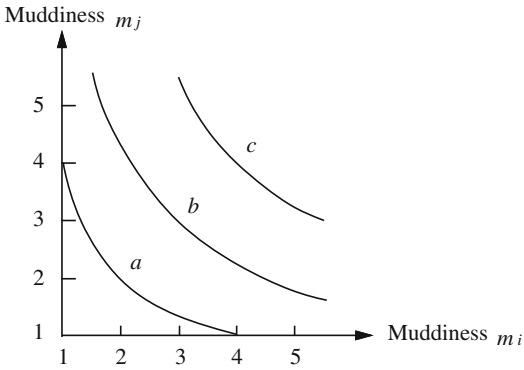

Fig. 5 Iso-muddiness curves in a 2-D muddiness frame

relatively clean task. The Euclidean distance from the origin does not have this property.

Three Task Categories

The composite muddiness and the muddiness factors in Table 1 enable us to appreciate what a developed adult brain can do. The 30,000 genes (i.e., genome) of the human being, residing in a single fertilized egg cell, enables its development from a single cell to a adult human being who handles many muddy tasks. The human genes are a result of evolution, while the human brain is a combined result of the inherited genes and the environments of the human individual while he lives in his society. Clearly, much information in the brain is from the environment, but the genome regulates its development.

Now that we have mastered the muddiness frame as a tool. We are ready to examine whether a given task is muddy. To facilitate our discussion, we divide all possible tasks into three major categories: 1, 2, and 3.

Category 1: Clean Tasks

What kinds of tasks are clean tasks? The list is extremely long. Examples of tasks in Category 1 include: Word processing, industrial control, digital communication, appliance control, digital computation, and playing some (simulated) games (e.g., IBM's Deep Blue). If we locate these tasks in our 26-dimensional muddiness frame as shown in Table 1, they all lie around the origin of our muddiness frame.

Category 2: Muddy Tasks

The tasks in Category 2 are muddy but they are intensively studied by researchers. This category contains tasks that are currently considered as core subjects of AI. Some example tasks in this category are visual object recognition, visual navigation, speech recognition, text-based language translation, sign language recognition, and text-based discourse.

Category 3: Very Muddy Tasks

Category 3 consists of mostly tasks for which humans have not yet built a machine to try. Some tasks in Category 3 are:

1. Learn about new muddy subjects—autonomously learn *any possible subjects* that are of high values, including those the machine maker does not know about. For example, learn about a new disease (e.g., AIDS and SARS when they were first discovered).
2. Create new knowledge—discover new facts about science and produce creative works on *any possible subjects* that are of high values, including those that not only the machine maker but also we humans do no know about. For example,

when the human conventional energy resources have been nearly exhausted, discover new sources of energy for humans.

The term "any possible subjects" above includes all the subjects that a normal human can potentially learn in his lifetime, although he may not necessarily actually learn all of them.

The tasks in this category are so muddy that little has been done for machines. However, this category is of fundamental importance, since a solution to the tasks in Category 3 holds the key to the solutions to the tasks in Category 2. Further, this category helps us to understand and appreciate human intelligence which in turn motivates us to seriously embark on the developmental route toward machine intelligence for muddy tasks. When robots can perform tasks in Category 3, they should be able to participate in the design, construction and training of robots with inputs from humans.

It is expected that the traditional, non-developmental agents are not going to handle tasks in Category 2 well, depending on how muddy the tasks are. This is mainly because those traditional agents require human programmers to program task-specific representations into the machines, which quickly gets out of hand if the tasks become muddy in the composite muddiness, as shown in Fig. 5.

That is why it is necessary to study and construct autonomous developmental agents. In the brain of such an agent, various task-specific internal representations are emergent, along with the emergent skills for internal (brain) and external (outside brain) actions, while the agent autonomously interacts with its living environments (including humans) (Weng 2004). Task-nonspecificity during the programming time is a fundamental difference between a developmental agent and a machine that can only learn in the sense of traditional machine learning (task-specific programming). Although a human can program a machine to execute a relatively clean task, autonomous mental development is the only practically way a general purpose highly intelligent agent comes into being if it must deal with many muddy tasks.

Intelligence Metrics

Based on the muddiness introduced above and the sample tasks discussed, I propose a measure of intelligence in terms of the capability of performing muddy tasks.

Definition 1 A measure of intelligence for an agent is in terms of the composite muddiness of the tasks that it performs. Collectively, the intelligence of an agent is measured in terms of the variety of muddy tasks it carries out and the muddiness of these tasks.

Thus, the muddier a task, the more intelligence is required on the part of the performer. The larger the variation of such tasks, the more intelligence is required. The proposed measure is not the only measure for intelligence, but it is certainly a candidate.

As is somewhat expected, humans and higher animals (such as dogs and cats) are much more intelligent than modern computers, if we use the muddiness introduced

here as the measure. Using this measure, intelligence is not something that is easy to demonstrate by current machines.

If intelligence is measured as the capability of performing a specific task, different AI tasks are then measured by very different metrics. However, what a special purpose machine can do under a specific setting represents mainly the intelligence of the machine programmer, not necessarily the machine's own intelligence. Further, a special purpose machine does not do well for muddy tasks, as defined here.

On one hand, it is totally justified to construct machines that perform specific tasks that humans do not like to do or cannot do well in specific settings. On the other hand, there are many tasks that are extremely muddy and humans would like machines to perform, such as autonomous driving, house cleaning, and personalized tutoring.

Therefore, the criteria for measuring machine intelligence need careful studies to systematically take into account muddy tasks. Consequently, the metrics to measure the power of intelligent machines should emphasize the capabilities of autonomous development in muddy human environments. As reaching a desired end needs means, testing the means is critical for identifying the potential to reach the end.

This is indeed the case with well-accepted test scales used by clinical psychologists for measuring mental and motor scales of human children. Two such well-known scales are The Bayley Scales of Infant Development (for 1–42 months old) (Bayley 1993), which has been widely used for operational clinical tests, and The Leither International Performance Scale (for 2–12 years old), which has several modern adaptations for operational clinical tests (e.g., (Arthur 1952)). These scales have a systematic methodology for the administration of tests and scoring. The reliability and calibration of these scales have been supported by a series of validity studies, including construct validity, predictive validity, and discriminant validity that cover a large number of test subjects and different age groups across wide geographic, social, and ethnic populations.

Let us take a look at an example of tests in the Leither International Performance Scale for a two-year old. The name of the test is Matching Color. The test setup is a row of 5 stalls, as shown in Fig. 6. Each stall is marked with a color card that indicates a certain color, black, red, yellow, blue, and green, respectively.

During the test, color blocks are presented, one at a time, in the order of: black, red, green, blue, and yellow. The examiner places the black block in the first stall and tries to get the subject to put the red block in place by putting it on the table before him, then in the appropriate stall, then on the table again, nodding to him to do it and at the same time pointing to the second or red stall. As soon as the subject begins to take hold of the test, the final trial can be attempted. In this test, the examiner tries to get the subject to imitate his procedure. The test is scored as passed if the subject is able to place the four colors (the first one is placed by the examiner) in their respective stalls on his own during any one trial, regardless of the number of demonstrations or the amount of help previously given by the examiner.

As we can see, the test is not concerned about whether the child has learned the abstract concept of color, but rather the capability of autonomous learning during

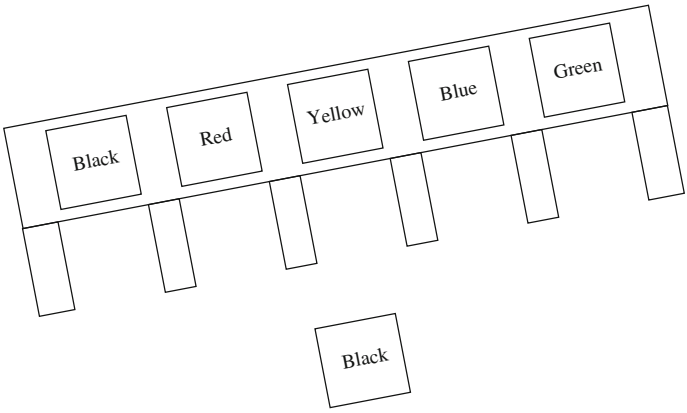

Fig. 6 The setting of Matching Colors in the Leither International Performance Scale for a two-year old. In the original test, actual colors are used. Here, the names of the colors are used instead for illustrating in black and white

which tasks to be learned about are not explicitly given. In particular, the subject must recall a series of desired actions from the corresponding real-time contexts, from imitating the actions of the examiner, to identifying visual color information as a major attended cue, to the coordination of his motor effectors (hands and arms), while the environments contain many irrelevant stimuli. All these desired perceptual and motor behaviors must be invoked in time even though imitation and the keys for success have not been explicitly specified before or during the test. This type of test requires the child to be able to pay attention to the examiner, the stall, the card, the color, etc., respectively, and to autonomously come up with the desired intent for each current context (e.g., what to imitate). Of course, color is just one of the many desired features required by the series of tests in a well designed performance scale. Variants of the matching colors include matching shapes, matching objects, etc.

The mental age that is used for measuring human intelligence in these scales can be used as a measure for general purpose machine intelligence. Currently metrics that have been used for various AI studies mainly measure what a machine can do under a specific setting, instead of the capability of mental development. Such a capability requires an online, interactive and incremental learning capability as the above test demonstrates. For example, an interactive dictionary stores a vast amount of human knowledge and can do remarkable things for humans, but it is not truly intelligent. If a machine can pass the systematic tests like the one shown above, it must have already learned many other skills that no traditional machine has. The performance metrics for measuring intelligent machines should be adapted from those used by clinical psychologists for testing the mental development of human children. Although autonomous mental development is a relatively new direction, its impact on the future of machine intelligence and understanding of human intelligence is far reaching.

Conclusions

The composite muddiness of a task introduced here is proposed as a measure of the intelligence of the performer, a human or a machine. Although mathematic tools are used by the model, the proposed model addresses mainly philosophical and conceptual questions of measuring muddiness of tasks. Of course, it is not the only way to carve up intelligence. However, currently such a metric is missing and it is important to propose one. It is useful for understanding tasks that are dealt with by humans and machines.

A manually designed *task-specific* representation in a traditional AI approach restricts its capability to deal with muddy tasks.

With many muddiness factors in a muddy task, it seems that autonomous development is suited for muddy tasks and is necessary for very muddy tasks, due to the *task-nonspecificity* of autonomous development. That is one of the major reasons why all natural higher intelligent agents go through an extensive process of autonomous mental development. Therefore, a critical metric for measuring general purpose machine intelligence is the capability of autonomous learning while the tasks and subtasks are implicit, as measured by test scales widely used by clinical psychologists. The true answers to how the brain works and how to realize higher machine intelligence lie in understanding how the brain develops. By how the brain develops, I mean not only how the brain's morphology changes, but also the biological and computational mechanisms of cell-centered (Sur and Rubenstein 2005; Weng et al. 2008) dynamic connection, cortical self-organization, adaptation and activity-dependent plasticity scheduling. Higher brain functions, such as perception, cognition, emotion, reasoning, thinking, motor skill perfection, emerge from extensive processes of situated autonomous development, regulated by these biological and computational mechanisms.

References

Arthur, G. (1952). *The Arthur adaptation of the Leither international performance scale*. Washington, DC: The Psychological Service Center Press.

Bayley, N. (1993). *Bayley scales of infant development* (2nd ed.). San Antonio, Texas: Psychological Corp.

Brooks, R. A. (2002). *Flesh and machines: How robots will change us*. New York, NY: Pantheon Books.

Cohen, P. R., & Howe, A. E. (1988). How evaluation guides AI research. *AI Magazine, 9*(4), 35–43.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1997). *Rethinking innateness: A connectionist perspective on development*. Cambridge, Massachusetts: MIT Press.

Enderle, J., Blanchard, S. M., & Bronzino, J. (2005). *Introduction to biomedical engineering* (2nd ed.). Burlington, Massachusetts: Elsevier Academic.

Franklin, S., & Graesser, A. (1997). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent Agents III, Lecture Notes on Artificial Intelligence* (pp. 21–35). Berlin: Springer-Verlag.

Gardner, H. (1993). *Multiple intelligences: The theory in practice*. New York: Basic Books.

Goleman, D. (1995). *Emotional intelligence*. New York: Bantam Books.

Meystel, A. M., & Messina, E. R. (Eds.) *Measuring the Performance and Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop*. Gaithersburg, Maryland: National Institute of Standards and Technology, August 14–16, 2000.

Michie, D. (1993). Turing's test and conscious thought. *Artificial Intelligence, 60*, 1–22.
Norman, D. A. (1991). Approaches to the study of intelligence. *Artificial Intelligence, 47*, 327–346.
Reik, W., & Dean, W. (2002). Epigenetic reprogramming back to the beginning. *Nature, 420*(6912), 127.
Russell, S., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. New Jersey: Prentice-Hall, Upper Saddle River.
Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). New Jersey: Prentice-Hall, Upper Saddle River.
Sur, M., & Rubenstein, J. L. R. (2005). Patterning and plasticity of the cerebral cortex. *Science, 310*, 805–810.
Tononi, G., & Edelman, G. M. (1998). Consciousness and complexity. *Science, 282*(5395), 1846–1851.
Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*, 433–460.
Weng, J. (2004). Developmental robotics: Theory and experiments. *International Journal of Humanoid Robotics, 1*(2), 199–235.
Weng, J., & Hwang, W. (2006). From neural networks to the brain: Autonomous mental development. *IEEE Computational Intelligence Magazine, 1*(3), 15–31.
Weng, J., Luwang, T., Lu, H., & Xue, X. (2008). Multilayer in-place learning networks for modeling functional layers in the laminar cortex. *Neural Networks, 21*, 150–159.
Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., & Thelen, E. (2001). Autonomous mental development by robots and animals. *Science, 291*(5504), 599–600.