# Muddy Tasks and the Necessity of Autonomous Mental Development

**Juyang Weng**
Embodied Intelligence Laboratory
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824 USA
weng@cse.msu.edu

In Proc. 2005 AAAI Spring Symposiums, Developmental Robotics Symposium, Stanford University, Stanford, CA, March 21-23.

**Abstract**

Why do we need autonomous mental development (AMD) for intelligent robots? This paper introduces a concept called *task muddiness* which supports AMD as a necessity for higher intelligence. The 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 measured by a number of muddiness axes. The composite muddiness explains why many challenging tasks, such as the DARPA Grand Challenge 2005 (DARPA 2004), are muddy and why the autonomous mental development (AMD) approach is necessary for muddy tasks.

## 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 considered very difficult (by humans), such as playing (simulated) chess games. However, they have done poorly in areas that are commonly considered easy (by humans), such as vision, audition and natural language understanding.

There have been numerous studies on measuring the intelligence of AI systems. The "imitation game," proposed by Alan M. Turing (Turing 1950), now known as the Turing Test, has greatly influenced the ways machine intelligence was studied. The limitation of such a symbolic text-based test has now been better recognized (e.g., see the article by Donald Michie (Michie 1993) and Donald Norman (Norman 1991)). The proposed *Total Turing Test* (Russell & Norvig 2003) includes computer vision to perceive objects and robotics to manipulate objects and move about. The National Institute of Standards and Technology (NIST) has been hosting Workshops on Measuring the Performance and Intelligence of Systems, known as PerMIS, held annually since 2000 (Meystel & Messina 2000), where the proposed metrics are largely 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 (see, e.g., Cohen & Howe 1988 (Cohen & Howe 1988)) but not tasks.



Intelligent Quotient (IQ) and Emotional Quotient (EQ) (Goleman 1995) have been proposed to measure human intelligence. The tests in the field of psychometrics concentrate on the differentiation of human individuals in a human age group. They are not designed for measuring intelligence between humans and machines, or between humans and animals. Howard Gardner (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 & Gerald Edelman proposed to use mutual information to measure the complexity of integrated biological neural systems (Tononi & Edelman 1998). Nevertheless, these studies do not provide a mechanism for evaluating how muddy tasks are across many AI tasks.

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 impractical 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 very 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. Based on the muddiness proposed here, this paper outlines three categories of tasks Categories 1 to 3 and explains why the current machines perform the tasks in Category 1 well, but not Category 2 and worse in Category 3. Finally we discuss the composite muddiness as a performance metric of intelligence, as a candidate among other alternatives that have been proposed (see, e.g., an excellent survey by Russell and Norvig (Russell & Norvig 2003)).

This paper does not directly address how to construct an intelligent machine. However, the work proposed here is important to examining an AI task. Albert Einstein said: "The mere formulation of a problem is far more essential than its solution." Some tasks to which we have applied our AMD approach have demonstrated that the approach has a high potential for dealing with very muddy tasks (Weng

2004).

Section 2 discusses the basic principles that motivated the muddiness concept. Section 3 gives an example about muddiness factors, which naturally calls for the multiple muddiness presented in Section 4. Section 5 introduces the composite muddiness. Section 6 uses the muddiness concept to introduce three categories of AI tasks. Section 7 discusses the composite muddiness as a metrics for intelligence and Section 8 concludes.

## Principles of Task Muddiness

We need to understand tasks using a generally applicable muddiness measure.

### Characteristics of muddiness

The concept of muddiness was motivated by the following considerations.

1. Across task domains. Muddiness can incorporate any task. E.g., a computer chess playing task can be compared with a face recognition task, in an intuitive way.
2. Independent of technology level. A task that is muddy today remains muddy in the future, no matter how advanced computer technology becomes.
3. Independent of the performer. A task that is muddy for machines is also for humans as well. However, humans are good at performing muddy tasks.
4. Quantifiable. It helps us to understand why an AI task is difficult in a quantitative way.
5. Amenable to evaluating state-of-the-art intelligent machines. It objectively measures the overall capacity requirement of AI tasks and, consequently, the machines that execute these tasks.
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 a system as an agent.

### A system as an agent

A system for performing one or more tasks is an agent. By standard definition, an agent is something that senses and acts (see, e.g., an excellent textbook by Russell & Norvig (Russell & Norvig 2003) and an excellent survey by Franklin (Franklin & Graesser 1997)).

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.

### A human as an agent constructor

Depending how a task is specified, a task can be a subtask of another more complex task. For example, playing a chess game 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 to a machine directly or to a human developer who fabricates the machine and writes programs for it?

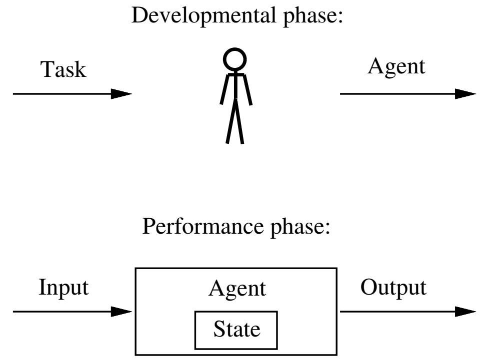


Figure 1: The *manual* developmental phase and the *automatic* performance phase. The developmental phase is not automated. The performance phase is partially or fully automated. The agent may have its internal state when it performs a task in the performance phase.

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. 1.

In the developmental phase, a human being accepts a task that the machine is supposed to perform. He understands and analyzes the task and then constructs an agent which is the 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 operation. It accepts an input and produces an output. Through this process, the agent performs an instance of the task. It may accept more input and produce an output for each instance. This way, the agent can perform more instances of the same task.

## Muddiness Frame Examples

It is not very 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 about 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 task has to consider. 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

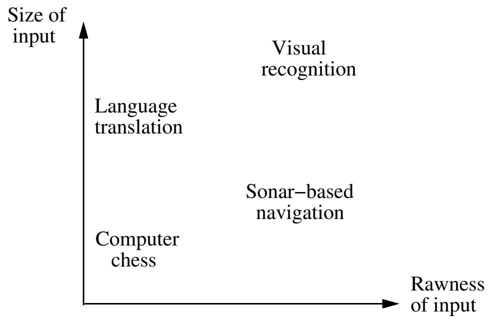


Figure 2: A muddiness frame for two muddiness factors, rawness of input and size of input. This diagram is for conceptual visualization only.

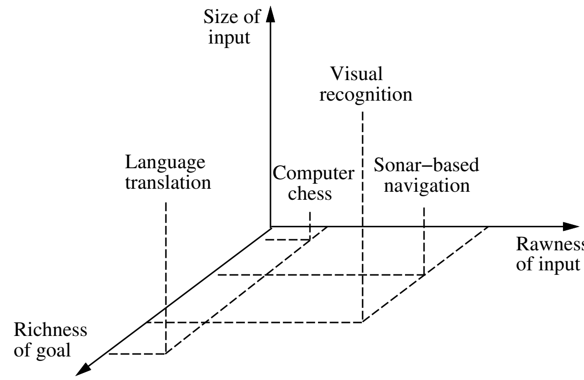


Figure 3: A muddiness frame for three muddiness factors.

$d = 240 \times 320 = 76800$ 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. 2. 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. It means how difficult it is to describe the goal of the task in a 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 without much ambiguity. Conversely, a program can always be written in mathematical terminology. We also insist that the description must be in terms of the input of the system since information available to the agent is 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 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? ... 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 well in terms of image input.

Take language translation as another example. It is almost impossible to write down mathematically the goal of translation for an article based on the text input. What do you mean by translating well? What do you mean by "meaning" in mathematical terminology? ... Thus, it is extremely difficult to express the goal of translation in mathematical terminology.

Augmenting the previous muddiness frame, by adding the richness of the goal, gives the muddiness frame shown in Fig. 3. We should not consider such positions for these tasks as absolute. A particular task arising in actual application can vary tremendously in all the three muddiness factors. For instance, technical language translation from 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 works. The input is the information that the agent receives from the external environment. The internal environment is the internal representation of the agent, i.e., its "brain" (does not include its body). The output is the output from the agent. The goal is the objectives of the tasks that the agent performs.

It can be proved that given a task, the five categories constitute a partition of the world:

**Theorem 1** *Suppose that $U$ is the universe that consists of everything in the world, real and imaginary. Given a task $T$ and an agent' brain $B$ required to perform the task, let $S_{ee}$, $S_{ip}$, $S_{ie}$, $S_{op}$ and $S_{gl}$ denote the five subsets of $U$: the external environment of $B$, input into $B$, internal environment of $B$, output from $B$ and the goal of $T$, respectively. $S_{ee}$, $S_{ip}$, $S_{ie}$, $S_{op}$ and $S_{gl}$, with a formal definition, give a partition of $U$. That is, these sets are non-overlapping (intersection is an empty set), and the union of these sets equals the universe $U$:*

$$U = S_{ee} \cup S_{ip} \cup S_{ie} \cup S_{op} \cup S_{gl}. \quad (1)$$

The proof is omitted due to the space limit. The above theorem indicates that the five sets contain everything in the universe, nothing has been neglected. In other words, any additional muddiness factor has a category to which it belongs. Of course, there are many other ways to partition the

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 |

universe. Other partitions may not necessarily be as intuitive and concise as the one defined here for our purpose.

Table 1 gives some major muddiness factors, grouped into the above five categories. Let us examine the additional factors of muddiness.

## External environment

*Awareness* means whether the programmer knows about the external environment in which the agent works when he does programming. *Complexity* measures how complex the external environment is. *Controlledness* refers to whether the environment is controlled. *Variation* indicates whether the environment is changing. *Foreseeability* means whether the future environment is foreseeable or not.

## Input

The *background of input* indicates whether the input includes information that is not related to the task at all. And, if it does include background, how complex is the background. The *variation of input* refers to the complexity of variation among inputs that require the same output. The *occlusion of input* is another factor of muddiness. Presence of occlusion in input makes a task muddier. The *activeness for input* indicates whether the agent must actively acquire input in order to perform the task and, if it must, how complex the active acquisition actions are. 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.

## Internal environment

It is difficult to understand the requirement 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 on not only the context state, but also the long term memory. For consistency with the 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 a motor control signal vector 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 internal environment* is the measured value of the size of the internal storage space needed. The *representation of internal environment* concerns whether the internal representation is given or not from the task specification to the task execution agent. It also characterizes how much information is given.

For this concept, the following distinctions are 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 turn 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, case (c) is the most prevailing, but we should not rule out case (b) since it is a desirable goal of AMD.

The *observability of internal environment* means whether the internal representation of the agent is observable by the outside world. Closely related to the observability of internal environment is the *imposability of internal environment*. The imposition here means that the human teacher directly sets the value of the internal representation of the agent. The representation of a 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 is the right thing to do. However, the parent cannot set what a child actually thinks about.

The *time coverage* of internal state characterizes how complex the required temporal coverage pattern is for the context when the task is performed.

## Output

The agent outputs its actions to its effectors. The *terminalness of output* reflects how the output can be used directly without human processing. 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. The *multi-modality of output* indicates how many distinct effector modalities are used.

## Goal

Each task has a goal. The *variability of goal* indicates whether the goal of a task may change, and the degree of change. The *availability of goal* means whether or not the goal is given at the time of machine construction. 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?

Finally, we have finished our examination for the repository of our muddiness factors. The list of muddiness factors in Table 1 is not meant to be exclusive. It is meant to provide enough detail for discussion.

# Composite muddiness

If we use $n$ muddiness factors, we can construct an $n$-dimensional muddiness frame, similar to what is in Fig. 2 for a 2-D case and Fig. 3 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 their coordinates. We should only use the muddiness frame in an intuitive and qualitative sense. 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. We denote the muddiness coordinate of the $i$th row in Table 1 as $m_i$. The value of $m_i$ should never be smaller than 1: $m_i \geq 1$. The composite muddiness of a task can be modeled by the product of all the coordinates:

$$m = m_1 m_2 ... m_n = \prod_{i=1}^{n} m_i, \qquad (2)$$

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 $i = 1, 2, ..., n$. 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. Fig. 4 plots three iso-muddiness curves in a 2-D muddiness frame using two mud-

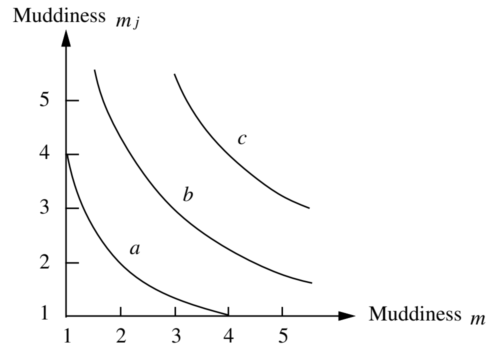


Figure 4: Iso-muddiness curves in a 2-D muddiness frame.

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

# Three Task Categories

Mastering the muddiness frame as a tool, we are ready to examine whether a given task is muddy. To facilitate our discussion, we divide 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. If we locate these tasks in our 20-dimensional muddiness frame, 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 artificial intelligence. 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. As an example, the following table summarizes the visual navigation task in the DARPA Grand Challenge 2005 (DARPA 2004), and the corresponding characteristics of the AMD approach (Weng 2004). Of

Table 2: Muddiness factors of DARPA Grand Challenge

| Cat. | Factor | Muddiness | AMD |
|---|---|---|---|
| Ext'l Env. | Awareness | Unknown | Env. open |
| | Complexity | Complex | Env. open |
| | Controlledness | Partially controlled | Env. open |
| | Variation | Changing | Allowed |
| | Foreseeability | Nonforeseeable | Env. open |
| Input | Rawness | Real sensor | Suited |
| | Size | Large | Suited |
| | Background | Complex | Suited |
| | Variation | Complex | Suited |
| | Occlusion | Present | Exp. dependent |
| | Activeness | Moderate | Suited |
| | Modality | Complex (video) | Suited |
| | Multimodality | Multiple | Suited |
| Int'l Env. | Size | Large | Suited |
| | Representation | Approach depend't | Not given |
| | Observability | Approach depend't | Unobservable |
| | Imposability | Approach depend't | Not imposable |
| | Time coverage | Simple | Exp. dependent |
| Output | Terminalness | High | Suited |
| | Size | Moderate | Suited |
| | Modality | Complex | Suited |
| | Multimodality | Multiple | Suited |
| Goal | Richness | Low | S.T.T.P. |
| | Variability | Fixed | S.T.T.P. |
| | Availability | Given | S.T.T.P. |
| | Convey-mode | Simple | S.T.T.P. |

S.T.T.P.: Short-term (a few months) of training possible
Exp. dependent: Dependent on training experience

course, whether an example system of the AMD approach will actually meet the Grand Challenge depends also on many other factors, such as funding, design, implementation, computational resource, training experience, system reliability, etc.

### Category 3: Very muddy tasks

The tasks in this category are so muddy that little has been done for AI tasks in this category. Category 3 consists of mostly tasks for which humans have not yet built a machine to try. This category is of fundamental importance, since a solution to the tasks in Category 3 probably holds the key to the solutions to the tasks in Category 2. Some tasks in Category 3 are: (1) Learn about new muddy subjects — autonomously learn *any possible muddy subjects* including those the machine maker does not know about. (2) Create new knowledge of a high value — discover new facts about science and produce creative works on *any possible subjects* including those the machine maker does not know about, and furthermore, those that we humans do not know about. The term "any possible subjects" includes all the subjects that a normal human can potentially learn in his lifetime, although he may not necessarily actually learn all of them.

## Intelligence Metrics

Based on the muddiness introduced above, 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 can perform. Collectively, the intelligence of an agent is measured in terms of the variety of muddy tasks it can carry out and the muddiness of these tasks.*

As somewhat expected, humans and higher animals (such as dogs and cats) are much more intelligent than modern computers, if we use the composite muddiness introduced here as the measure. Using this measure, intelligence is not something that is easy to demonstrate by current machines, but the measure demonstrates the importance of the AI field and directions for potential breakthroughs.

## Conclusions

The composite muddiness of a task introduced here is a measure of the intelligence of the performer. With many muddiness factors in a muddy task, it seems that AMD is necessary to handle highly muddy tasks, due to its *task-nonspecificity* (Weng 2004). A manually designed *task-specific* representation in a traditional approach restricts its capability to deal with highly muddy tasks.

## References


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

DARPA. 2004. DARPA grand challenge 2005: Rules. Technical report, DARPA.

Franklin, S., and 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*, 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., and Messina, E. R., eds. 2000. *Measuring the Performance and Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop*. Gaithersburg, MD: National Institute of Standards and Technology.

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.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall, 2nd edition.

Tononi, G., and 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.