Elements of Task Theory

Matteo Belenchia,[1,3] Kristinn R. Thórisson,[1,2]
Leonard M. Eberding[1], and Arash Sheikhlar[1]

[1] Reykjavik University
Menntavegur 1, 102 Reykjavík, Iceland
{matteobel20, thorisson, leonard20, arash19}@ru.is
[2] Icelandic Institute for Intelligent Machines (IIIM)
Reykjavík, Iceland
[3] University of Camerino
Via Andrea D'Accorso 16, Camerino, Italy

Abstract. Tasks are of primary importance for artificial intelligence (AI), yet no theory about their characteristics exists. The kind of task theory we envision is one that allows an objective comparison of tasks, based on measurable physical properties, and that can serve as a foundation for studying, evaluating, and comparing learning controllers of various kinds on a variety of tasks by providing principled ways for constructing, comparing, and changing tasks with particular properties and levels of difficulty. In prior papers we have outlined an approach towards this goal; in this paper we present further principles for its development, including causal relations. We use these principles to expand our prior ideas, with the aim of laying the groundwork for covering levels of detail, prior knowledge of the learner/performer and task difficulty, to name some of the complex issues that must be solved for a useful task theory.

Keywords: Tasks · Environments · Task Theory · Artificial Intelligence · General Machine Intelligence · Evaluation

1 Introduction

Artificial intelligence (AI) systems are built to perform tasks. Whether primarily hand-coded or based on machine learning techniques, intended to perform only a single well-defined task or aiming for general machine intelligence (GMI), tasks are center of stage in the design of all AI systems; tasks also play a key role in their evaluation. In other fields of control engineering systems are evaluated by constructing test batteries to ensure their proper performance, and tune task parameters according to well-understood principles for their effective evaluation.

Despite their importance in AI, no methodology has widely been adopted for the use of tasks in AI research, and no theory about the properties of tasks exists [14]. A lack of a proper task theory in AI has persisted, possibly due to a long-lived constructionist design tradition that relies on hand-crafted solutions[4]

[4] For a discussion on constructionist (allonomic) vs. constructivist (autonomic) design methodologies in AI, see Thórisson et al. [11].

that primarily relies on human domain knowledge. It is therefore not a huge surprise, perhaps, that after 70 years of AI research the only obvious solutions for how to generalize narrow-AI systems are more domain-dependent hand-crafted solutions.

The upshot is that scientific comparison of two AI systems built by two separate teams, or the same team at different times, is currently costly and difficult at best, and impossible at worst. It is thus rarely done; most AI researchers design specific tasks for their specific systems. Without a general theory of tasks, comparing results of AI systems on various tasks is prohibited except through costly experimental procedures. The situation may be survivable in narrow-AI development, but for research in GMI this situation cannot persist if we want to be able to ensure their proper design and safe operation. Domain knowledge and tests (e.g. the Turing test or IQ tests) don't nearly cover the breadth of situations these systems could be facing. As progress towards GMI moves forward, the need to evaluate them in a range of circumstances, tasks, and situations increases. Without a general-purpose methodology that allows comparison along relevant dimensions of variables, tasks, situations and environments, the problem is only going to get worse.

In this paper we extend our prior work on this topic and present further principles for moving towards the kind of task theory envisioned. The focus in this paper is exclusively on the characterization of tasks based on their physical properties, in particular, we aim to resolve some issues that must be addressed to further the research towards e.g. measures of difficulty [2].

2 Related Work

Past research on task composition and analysis focuses quite heavily on performers, rather than the tasks. Task analysis for human tasks has been used since the mid-1900s to make various judgments and design decisions by providing the engineer with a "blueprint" of user involvement, unsurprisingly focusing rather exclusively on it from a 'human-level intelligence' perspective [6]. The GOMS (goals, operators, methods, and selection rules) task-analysis methodology, for instance, is a framework that characterizes a system user's procedural knowledge and can be used to predict human learning and qualitatively describe how a user will use an interface to complete a task [5]. *Cognitive task analysis* (CTA) has a similar purpose and works in a similar way. CTA describes the basis of skilled performance and, unlike GOMS, can explain what accounts for mistakes [1]. Another way to describe tasks is through *hierarchical task networks* (HTN) [4]. HTN are used to decompose high-level tasks into atomic actions to create plans to achieve a goal. Strictly speaking, HTNs do not model the environment but rather produce a list of actions for solving a task [4].

The existence of an intelligent performer is fundamentally assumed a-priori in all these approaches, which makes them rather irrelevant in the design, training, and evaluation of AI systems. For AI the aim must be a dissection and analysis of tasks in such a way that the performer's "IQ" is not given a-priori, and conclusions about the performance of an agent, and thus its design, can result from it, rather than the other way around, without the need for experimentation.

Thórisson et al. [14, 13] have proposed a set of necessary components to describe measurable physical dimensions of tasks that must be adjustable by an evaluator when the goal is to get insight into the different approaches to AI and autonomous learning. The advantage of their proposed approach has been demonstrated in part by Eberding et al. [3], providing experimental results of the possibilities that a well-designed task theory could give developers and evaluators of AI systems. To describe tasks in such a way that they can directly help across a wide range of AI research endeavors, further principles must be developed.

3 Terminology & Background Assumptions

If the world is a closed system with no outside interference, the domains and invariant relations can be implicitly fully determined by the dynamics functions and the initial state. In an open system where changes can be caused exogenously, the explicit definition of domains and invariant relations can restrict the range of possible interactions [14].

An **environment** is a view on a world, typically inside a domain (like your kitchen is one environment within the domain of kitchens)—a domain in this view is thus a family of (related) environments [14]. We also may consider the body of an agent to be part of the task, rather than the agent, because it naturally constrains what the controller can do. Another thing to keep in mind is that the boundary between task and environment isn't always clear.

By **controller** we mean exclusively the mind of an embodied agent: The controller is the complete cognitive architecture of the system, which can receive inputs (observations) and produce outputs (commands) from the environment, and has its own internal state and goals.

The **body** of an agent is the interface between a controller and external world. The body itself, i.e. its transducers, belongs to the environment, following the laws of the environment and interacting with it, directly constraining the controller: Only variables which can be measured by the sensors of the body can be observable at any time and only variables belonging to the body can be directly manipulated by the controller. This body can be generally understood as a set of sensors producing sensory information that is read by the controller and a set of actuators that execute the controller's commands that act as the way the mind can affect the surrounding world. Therefore, different sets of sensors and/or actuators also determine how a task may be affected and how its state (values of variables) may be measured, including what is possible to do in the given execution environment. As shown below, the body of the agent significantly influences foundational principles of a task.

The **variables** in an environment, at any point in time, can be either *observable* (to a degree) or *non-observable*, *manipulatable* (to a degree) or *non-manipulatable*. Assignment of a variable in either pair is mutually exclusive, but either value in one pair can have either value in the other pair (e.g. a manipulatable variable can be non-observable and vice versa). Which variables hold which property can vary in any domain, environment, and through time, depending on their dominant relations at that time in that domain or environment.

A **task** is a problem assigned to an agent, $T = \langle \mathcal{S}_0, \mathcal{G}_t, \mathcal{G}_s, G^-, B, t_{go}, t_{stop} \rangle$, where S_0 is the set of permissible initial states, $\mathcal{G}_t$ is the task's set of top-level goals, $\mathcal{G}_s$ is the set of given sub-goals, G^- is its set of constraints, B is a controller's body, and t refers to the permissible start and stop times of the task [12]. An *assigned* task will have all its variables bound and reference an agency that is to perform it (accepted assignments having their own timestamp t_{assign}). This assignment includes the manner in which the task is communicated to the agent, for example, whether the agent is given a description of the task a priori, receives additional hints, only gets incremental reinforcement signals as certain world sub-states are reached, or some mixture of these. A task is successfully performed when the world's history contains a path of states that matched the task's specification, and thus solved the problem it describes.

The **problem space** S_{prob} of a task describes all valid states of the task-environment which can exist at any time within the temporal boundaries of the task through any action or inaction of the controller. It is constrained by the laws of the task-environment (like the speed of light in the physical world).

The **solution space** S_{sol} of a task is a subset of the problem space, defined by the task's goals $\langle \mathcal{G}_t, \mathcal{G}_s \rangle$ and constraints $\langle G^- \rangle$. For a task T_1, any (partial) state is part of the solution space of T_1 that 1) can be reached from an initial state $S_1 \in \mathcal{S}_0$ without violating the task specification and 2) from which at least one path of states exists which matches the task's specifications, leading to a (partial) state that matches the task's goal(s).

4 Foundational Principles for Task Theory

Based on the above background assumptions we can now turn to some unresolved issues that we consider important for a proper task theory. These range from the relationship between a controller's body and the task-environment, to task decomposition and level of detail (LoD). It should be noted that in the following we take the designer's viewpoint, which differs from the learner's viewpoint in that it assumes a complete overview of the task at hand.[5]

4.1 Causal Relations

A physical 'mechanism,' in our approach, is a directional function that determines the value of some world variables (the *effect*) from the values of other variables (the *causes*). The underlying assumption is that actions produced by a controller, via its body, are *local 'surgeries'* in the space of mechanisms [8], and

[5] In the physical world a complete overview of a task is theoretically prevented, but we can nevertheless assume that critical differences exist between a teacher's view and a pupil's.

those mechanisms are, given certain conditions, invariant and independent of each other.[6] By 'causal knowledge' is meant information that allows an agent to take action that perturbs a mechanism (via it's own body), causing predictable changes in other mechanisms relevant to task goals, i.e. $\{A_{t1} \rightarrow B_{t2}\}$ where $\{A, B\}$ are events at time t and $t2 > t1$. This means that an action affecting a mechanism leaves other mechanisms in their place, and that the effects of such actions can then be predicted using appropriate causal and other relational knowledge (in the form of models). A *chain* of such causal relational knowledge can represent a plan, an explanation, or a re-construction of a particular aspect of a phenomenon. Achieving goals in the context of any phenomenon necessarily requires knowledge of relevant causal relations, in particular, of the causal relations that relate manipulatable and observable variables of the phenomenon to the goals of an assigned task.

§1 Getting things done means making use of (models of) causal relations.

The existence of any causal relations between relevant variables must either be known by a performing agent or discovered by it in the process of performing a task. Such information must be represented in some cognitively manipulatable way, so that the controller can retrieve relevant knowledge in particular circumstances. Elsewhere, we have proposed causal-relational models (CRMs) to represent such information [15, 7]. Whichever representation is used, however, any such representation aimed for tasks in complex domains must be able to represent different levels of detail, since:

§2 Complex domains (like the physical world) contain more than one level of detail (*LoD*).

Consider, for example, how the interior of a house can be seen at the atomic, molecular or interior design level of detail. This means that the causal relations in multi-LoD task-environments (e.g. the physical world) can be thought of as forming (one or more) hypergraphs. One way to conceptualize the process of learning about such a graph is to consider it a modeling process, whereby the models formed need to mirror, in some useful way, these. Following Conant and Ashby [10], a good controller of a system must reference a *model* of that system. For causal relations, this implies that in any multi-LoD domain:

§3 The granularity of domain modeling must match the *LoD* of the causal relations at the lowest *LoD* relevant to a task's goal(s).

If a task involves genetic engineering, the lowest relevant LoD is chemistry, because that's where the task's success or failure will be measured; if a task involves getting some furniture from one office to another office the relevant LoD is object placement measured in centimeters. Let's look further at principles related to LoDs.

[6] While our approach is fundamentally non-axiomatic, cause-effect relationships are probably appropriately considered Platonic. This neither diminishes nor prevents their value or usefulness when dealing on a conceptual level with complex multi-LoD systems like the physical world.

4.2 Levels of Detail

Given that a task can be described at various levels of detail, which level of detail may be most appropriate in a particular case for a particular learner? Considering that the body of the controller, with its sensors and actuators, is *part of the task* (see Section 3 above), and considering that the perception and the interaction of the controller with the task-environment happens through this body, then it is at least possible to set a lower bound to the possible level of task abstraction. The body of the controller constrains the level of detail which is to be used to describe the task: The actuators define the granularity of what can be manipulated while the sensors define the granularity of what can be measured.

§4 A controller's transducers define the finest level of relevant spatio-temporal task detail.

Therefore, the finest possible level of detail for a task depends *necessarily on what the body allows the controller to observe and manipulate*, and tasks described at more fine-grained levels of detail than what the controller's body allows would be experienced by the controller at coarser level of detail, in accordance with what is made possible by its body. For example, if a set of transducers operates at the centimetric level of detail, a description of the molecular or atomic interactions in the task is unnecessary, as they can't be experienced by the controller.

Any phenomenon in the world can be described at different levels of detail, from highly detailed fine-grained descriptions to more abstract, coarse-grained ones. This also applies to tasks: Task specifications can vary and can be made more or less abstract, arbitrarily, ranging from the very general high-level instructions in everyday language to the overly complex descriptions that – at least in theory – can be made at the atomic or even sub-atomic levels. This presents a problem in evaluating an agent A_1 on a task T_1: Let's say that T_1 is to change a spelling mistake in a word in a given electronic document; in one task-environment is to be done by modifying the values of transistors on a CPU, in the other the change is to be made using word processor software. To any human the former will probably always be more difficult[7] than the latter (even experienced CPU designers). One way to address this issue in a task theory is to introduce the idea of a level-of-detail operator that controls for the level of description with respect to the performer's body (sensors and actuators). This has the potential benefit of homogenizing any task relative to its level of detail, for a particular performer. However, how such an operator would produce this result is unclear. Another way is to simply treat the level of detail as part of the task's constraints, $\langle G^- \rangle$. While this is perhaps a less elegant solution, it is exceedingly simple. This gives rise to the following principle:

§5 The level of detail *(LoD)* is part of the task.

In other words, any task is limited to its level of detail, and if the "same" task is presented at another level of detail, it is *not the same task*. For example, an

[7] Producing a useful measure of difficulty is the purview of a proper task theory; this is addressed elsewhere [2].

electronic circuit implementing a logic gates task can be described at the level of its electronic components, at a yet lower level of the chemical reactions in its circuits, or at the higher level of the implemented logic circuit. The task of obtaining some output in such a circuit changes significantly according to the level of description being used. This is because effectively, the variables and the mechanisms changed together with the level of detail. Therefore, variations in the level of detail result in *different* tasks.

4.3 Task Difficulty

The difficulty of executing any particular task is not uniquely determined by the task itself. Some controller might be better or worse suited to perform the task for a plethora of reasons: it could have trained on similar tasks or on tasks which share some of the variables and relationships with this task, it could be quicker (or slower) at learning associations and cause-and-effect relationships, and so on. Controllers, and by controller we mean effectively the mind of the intelligent system, might have either the experience or the architecture that is particularly well-suited (or ill-suited) for the task at hand, or for a type of tasks in general, or for any task at all, for reasons completely independent of the tasks themselves. Difficulty must therefore be a cross product of a task and a controller:

§6 The *difficulty of a task* is a product of the features of the task T and the features of the controller C; i.e. $\{T \times C\}$.

This concept of difficulty includes end-effectors, dexterity, sensors, etc. Note that all end-effectors in nature (extremities, skin, etc.) contain sensors as well that tell about their status; and vice-versa, sensors also are paired with end-effectors (ears on a movable head, rotating eyes, all mounted on a movable body). Thus, it may be said that all effectors are also sensors and vice versa, the difference being merely lie in the direction of information flow amplification. It should be noted that by 'task' we mean task-environment, as variables other than those of the task proper ('task family') may be essential for their completion.

A closely related problem is this: A task becomes easier the more we learn how to do it. This complicates the potential comparison between two controllers that we wish to compare, where one of them knows more about it than the other. How does prior training/ and knowledge affect task analysis/ and task design? This is solved by excluding any part of a task that a controller already *knows how to do*, leaving only the parts that the controller must learn (to whichever extent). This, however, requires separating the task designer's viewpoint from the task learner's viewpoint: From the designer's point of view it is assumed that everything about the task is known and specified. The designer has complete knowledge of the ground truth of the task, including variables, mechanisms, goals, constraints and so on. The learner, on the other hand, has limited knowledge of the task, owing to its limited perception and experience of the task and world. Its knowledge of the task comes with no certainty of correctness: It is defeasible knowledge which could at any time be proven wrong by subsequent experience [9]. The upshot is that we can include prior knowledge in any discussion of difficulty:

§7 Any measure of the *difficulty of a task* must take note of the performer's prior training and knowledge, and thus, prior knowledge is part of C in the equation $\{T \times C\}$.

Given the lower bound of useful granularity to describe a task, the question arises which of the theoretically usable variables of the world influence the task. It can be argued that all variables which are part of the environment necessarily must be part of the task. However, to describe the task in relation to the agent's goals the focus should lie on variables which constrain the solution space of the task. Non-constraining variables are those of no importance to solve the task. Therefore they need not be modeled by a learner of the task.

§8 A task is unchanged by variables which do not constrain its solution space.

While such variables and relations are superfluous to the task when looking at it from the designer's perspective, they can influence the learning of an agent. Especially when such variables are observable to the agent they can lead to wrong or misleading correlations with solution space constraining variables. When taking the agent's perspective these misleading correlations between superfluous and non-superfluous variables becomes an issue of experience. If previous encounters with superfluous variables have lead to a knowledge generalization which excludes these variables they do not influence the performance of the agent on the task. If the agent has not yet learned about these elements they can prolong learning times by making it more difficult to extract relevant causal structures from the observations. For example, the presence of multiple switches does not affect the task of turning on the lights (assuming only one such switch is needed).

5 Discussion

We consider the principles thus outlined still up for discussion, as there are unforeseen implications for any of the suggested commitments. Many questions remain to be answered, in particular with respect to whether some existing paradigms or methodologies might be suited to either appropriately address the issues raised here, or possibly explain them away. As far as we are able to see, no particular theory exists, and no existing paradigm, addresses in a unified manner the issues of level of detail, causal chains, and multi-goal achievement. These, in our opinion, must be included for a proper theory of tasks. Let's take a brief look at some of the approaches mentioned in the Related Work section, to see whether they could possibly challenge, address, or extend, our proposed principles.

In hierarchical task networks (HTN, [4]), causality is considered, but only as a high-level relation between tasks. When two tasks interfere with each other, a causal relationship is recorded indicating the two tasks t_e and t_p and a predicate q which is both an effect of t_e and a pre-condition of t_p [4]. Also, when the goal e of a task t results already achieved, by the effect of another task t', a constraint of the form (t', e, t) is added to the network to record it as a causal relation [4].

In contrast, we consider causality in a more fine-grained way, as a relationship between variables in the same task (**§1**). In this sense our approach to causality is more general, because it can be also applied at coarser levels of detail to trace causal relationships between the tasks themselves (e.g. the effect of executing the task of "walking to the door" is also the pre-condition for starting the task of "opening the door".)

In the related work we surveyed, we found no discussion about the different levels of detail in the physical world and how to deal with them. Usually, the level of detail of the task is given and fixed from the start. Therefore the novelty of our approach consists (1) in realizing that there are always multiple levels of detail to deal with (**§2**), (2) that the level of detail is constrained by both the goals of the task (**§3**) and the body of the controller (**§4**) and (3) that the selected level of detail for the task is an intrinsic characteristic of the task itself, which when changed also determines a change of the task (**§5**).

To the best of our knowledge, this is the first to attempt to formalize the notion that the difficulty of a task depends on both the features of the task and the controller (**§6**), as past approaches have mainly considered the characteristics of the task alone. For example, typical attempts to AI evaluation use games like chess or arcade games, which are considered interesting because of the purported difficulty of the task itself (from the human point of view). The notion that a task's successful execution depends on prior training and knowledge (**§7**), on the other hand, is the main premise of artificial intelligence.

We are not aware of any discussion about the effects of eventual 'superfluous' variables in the task. This is mostly due to the fact that the tasks under consideration were defined a-priori to include only variables that in some way constrain the solution space. Therefore, the intuitive notion that 'superfluous' variables do not change a task is made explicit here (**§8**).

6 Conclusion

We summarized previous findings of our work and described foundational principles of tasks. We expect that such a task-theory can help to understand the pros and cons of different approaches to AI architecture design, help researchers to evaluate their (and other's) systems, compare them, and help developers of GMI-aspiring systems focus on the task at hand: Building systems capable of solving complex tasks in complex environments.

The introduced principles of task theory helps to avoid an anthropomorphic view on tasks and agents, which we hope reduces bias in evaluation and design of agents. We believe that by describing these principles, mistakes of the past might be avoided (e.g. over-amplifying the importance of certain task such as board games, video games, and others). Instead, by identifying task properties, describing them in causal structures and analyzing them thoroughly, the importance of a task being solved can be better understood and classified accordingly.

A future requirement for a task theory is to make changes in the level of detail of a task inherently available to the analyst. While this changes the task (see §5), the level of detail is of importance when analyzing different learners. If a learner is able to group relational models and task variables in order to change the level of detail of interaction by itself (hierarchical learners are the most promising ones for that) it is necessary to represent these changes when analyzing the task. Future work also includes the construction of tasks at multiple different levels of detail.

References

1. Crandall, B., Klein, G., Klein, G.A., Hoffman, R.R., Hoffman, R.R., et al.: Working minds: A practitioner's guide to cognitive task analysis. MIT Press (2006)
2. Eberding, L.M., Belenchia, M., Sheikhlar, A., Thórisson, K.R.: About the intricacy of tasks. In: International Conference on Artificial General Intelligence. Springer (2021, in submission)
3. Eberding, L.M., Sheikhlar, A., Thórisson, K.R., Andrason, S.P.: Sage: Task-environment platform for evaluating a broad range of AI learners. In: International Conference on Artificial General Intelligence. Springer (2020)
4. Georgievski, I., Aiello, M.: An overview of hierarchical task network planning (2014)
5. Kieras, D.: Goms models for task analysis. The handbook of task analysis for human-computer interaction pp. 83–116 (2004)
6. Kirwan, B., Ainsworth, L.K.: A guide to task analysis: the task analysis working group. CRC press (1992)
7. Nivel, E., Thórisson, K.R.: Replicode: A constructivist programming paradigm and language. Technical RUTR-SCS13001, Reykjavik University School of Computer Science (2013)
8. Pearl, J.: Causality. Cambridge University Press, Cambridge, UK, 2 edn. (2009). https://doi.org/10.1017/CBO9780511803161
9. Pollock, J.L.: Defeasible reasoning. Cognitive science **11**(4), 481–518 (1987)
10. Roger C. Conant, W.R.A.: Every good regulator of a system must be a model of that system. Int. J. Systems Sci. **1**(2), 89–97 (1970)
11. Thórisson, K.R.: A new constructivist AI: From manual construction to self-constructive systems. In Theoretical Foundations of Artificial General Intelligence (Pei Wang and Ben Goertzel, eds.) **4**, 145–171 (2012)
12. Thórisson, K.R.: Seed-programmed autonomous general learning. In: Proceedings of Machine Learning Research. pp. 32–70 (2020)
13. Thórisson, K.R., Bieger, J., Schiffel, S., Garrett, D.: Towards Flexible Task Environments for Comprehensive Evaluation of Artificial Intelligent Systems & Automatic Learners. In: Proceedings of the International Conference on Artificial General Intelligence. pp. 187–196 (2015)
14. Thórisson, K.R., Bieger, J., Thorarensen, T., Sigurðardottir, J.S., Steunebrink, B.R.: Why artificial intelligence needs a task theory – and what it might look like. In: Proceedings of the 9th International Conference on Artificial General Intelligence. pp. 118–128. Springer (2016)
15. Thórisson, K.R., Talbot, A.: Cumulative learning with causal-relational models. In: International Conference on Artificial General Intelligence. pp. 227–237. Springer (2018)