Open-Ended Learning: A Conceptual Framework Based on Representational Redescription

Stéphane Doncieux, David Filliat, Natalia Díaz-Rodríguez, Timothy
Hospedales, Richard Duro, Alexandre Coninx, Diederik M. Roijers, Benoît
Girard, Nicolas Perrin, Olivier Sigaud

Open-Ended Learning: A Conceptual Framework Based on Representational Redescription

Stephane Doncieux[1]*, David Filliat[2], Natalia Díaz-Rodríguez[2], Timothy Hospedales[3], Richard Duro[4], Alexandre Coninx[1], Diederik M. Roijers[5], Benoît Girard[1], Nicolas Perrin[1] and Olivier Sigaud[1]

[1] Sorbonne Université, CNRS, ISIR, Paris, France, [2] U2IS, INRIA Flowers, ENSTA ParisTech, Palaiseau, France, [3] Institute of Perception, Action and Behaviour, University of Edinburgh, Edinburgh, United Kingdom, [4] GII, Universidade da Coruña, A Coruña, Spain, [5] Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, Netherlands

Reinforcement learning (RL) aims at building a policy that maximizes a task-related reward within a given domain. When the domain is known, i.e., when its states, actions and reward are defined, Markov Decision Processes (MDPs) provide a convenient theoretical framework to formalize RL. But in an open-ended learning process, an agent or robot must solve an unbounded sequence of tasks that are not known in advance and the corresponding MDPs cannot be built at design time. This defines the main challenges of open-ended learning: how can the agent learn how to behave appropriately when the adequate states, actions and rewards representations are not given? In this paper, we propose a conceptual framework to address this question. We assume an agent endowed with low-level perception and action capabilities. This agent receives an external reward when it faces a task. It must discover the state and action representations that will let it cast the tasks as MDPs in order to solve them by RL. The relevance of the action or state representation is critical for the agent to learn efficiently. Considering that the agent starts with a low level, task-agnostic state and action spaces based on its low-level perception and action capabilities, we describe open-ended learning as the challenge of building the adequate representation of states and actions, i.e., of redescribing available representations. We suggest an iterative approach to this problem based on several successive Representational Redescription processes, and highlight the corresponding challenges in which intrinsic motivations play a key role.

Keywords: developmental robotics, reinforcement learning, state representation learning, representational redescription, actions and goals, skills

1. INTRODUCTION

Robots need world representations in terms of objects, actions, plans, etc. Currently such representations are carefully designed and adapted to the robot's task (Kober et al., 2013). But a general purpose robot capable of solving an unbounded number of tasks cannot rely on representations hardwired at design time, because each may require a different representation. To achieve the vision of a robot that can solve an open-ended series of tasks in an increasingly efficient way, we consider an alternative paradigm: that the robot should discover the appropriate representations required to autonomously learn each task.

Representational redescription is the ability to discover new representations based on existing ones. It is a key ability of human intelligence (Karmiloff-Smith, 1995) that remains a challenge in robotics (Oudeyer, 2015). In this paper, we propose a unifying conceptual framework for addressing it. We assume an agent endowed with low-level perception and action capabilities which receives external rewards when it faces a task. We also assume the agent is endowed with reinforcement learning (RL) capabilities efficient enough to let it learn to solve a task when cast as a Markov Decision Process (MDP). From these assumptions, the main challenge in our framework is determining how an agent can discover the state and action representations that let it cast tasks as MDPs, before solving them by RL (Zimmer and Doncieux, 2018).

In MDPs, states and actions are primitive components considered given, and they are generally defined by the human designer having a particular task and domain in mind (see **Figure 1**). To make a step toward open-ended learning, we propose a conceptual framework for representational redescription processes based on a formal definition of states and actions. Then we highlight the challenges it raises, notably in terms of intrinsic motivations.

2. THE REPRESENTATIONAL REDESCRIPTION APPROACH

Our Representational Redescription approach is depicted in **Figure 2**. We consider an agent endowed with low-level perception and action capabilities, and which faces an open-ended sequence of tasks. The agent receives some external rewards from these tasks. The problem for this agent is to determine how to use this reward to learn the corresponding task. In an MDP, an RL algorithm explores the possible outcomes of an action when executed in a particular state. As pointed out by Kober et al. (2013), there is a need to appropriately define the state and action spaces for an efficient learning process. To do so, the possible alternatives are either to rely on a single generic state and action space or to build them on-the-fly when required. In this work, we do the latter and make the following assumptions:

ASSUMPTION 1. *A* single *state and action space cannot efficiently represent all the decision processes required to solve the tasks an open-ended learning system will be confronted to. To solve the task k defined through a reward value* $r_k(t)$*, the agent needs to build an MDP* M_k*.*

ASSUMPTION 2. *An open-ended learning process needs to build these MDPs on-the-fly.*

ASSUMPTION 3. *The agent is endowed with some RL algorithms to allow it to learn to solve the task, once the underlying MDP has been fully defined.*

3. CONCEPTUAL FRAMEWORK AND BASIC DEFINITIONS

3.1. Markov Decision Processes

Decisions in robotics can be modeled with MDPs using $< S_k, A_k, p_k, R_k >$, where k is a task identifier[1], S_k is the state space, A_k is the action space, $p_k : S_k \times A_k \times S_k \rightarrow \mathbb{R}$ is a transition function, where $p_k(s_t, a_t, s_{t+1})$ gives the probability to reach s_{t+1} from s_t after having applied action a_t and $R_k : S_k \rightarrow \mathbb{R}$ is the reward function. A policy $\pi_k : S_k \rightarrow A_k$ is a process that determines which action to apply in any state.

In the proposed framework, the observed reward $r_k(t)$ is distinguished from the reward function of the MDP $R_k(t)$. The agent may not know to what state the observed reward $r_k(t)$ can be associated. It is actually part of the proposed open-ended learning framework to *interpret* observed rewards and associate them to states in order to build the reward function $R_k(t)$ required to learn how to maximize them.

The notations used here have been intentionally kept as simple as possible. This framework can be easily extended to more complex cases, including semi-MDPs, stochastic policies or other definitions of the reward function.

3.2. States

DEFINITION 1. *A state is a description of a robot context that respects the constraints of its decision process.*

Following (Lesort et al., 2018), a good state representation should be (1) Markovian (i.e., the current state summarizes all the necessary information to choose an action), (2) able to represent the robot context well enough for policy improvement, (3) able to generalize the learned value-function to unseen states with similar features, and (4), low dimensional for efficient estimation (Böhmer et al., 2015). State representation learning approaches learn low dimensional representations without direct supervision, i.e., exploiting sequences of observations, actions, rewards and generic learning objectives (Lesort et al., 2018).

To bootstrap the open-ended learning process, we define S_0 as the state space containing the set of possible sensorimotor values. This first state space may not be low dimensional, Markovian, or structured enough for efficient exploration, thus motivating the search for better adapted state spaces.

3.3. Reward Functions and Goals

A reward function may contain different kinds of information: an indication of success in fulfilling a Human user defined task, or in reaching an autonomously defined action goal (see next section). It may also contain guidance to help reach the goal (reward shaping).

Besides reward functions defined in $\mathbb{R}$, the proposed framework requires, for the description of actions, the definition of boolean reward functions that will be called *goal functions*:

---

[1]A single state and action space can be used for several tasks and a single task could be associated to multiple representations, but we use this notation to highlight the dependency between tasks and MDPs that is central to this framework.
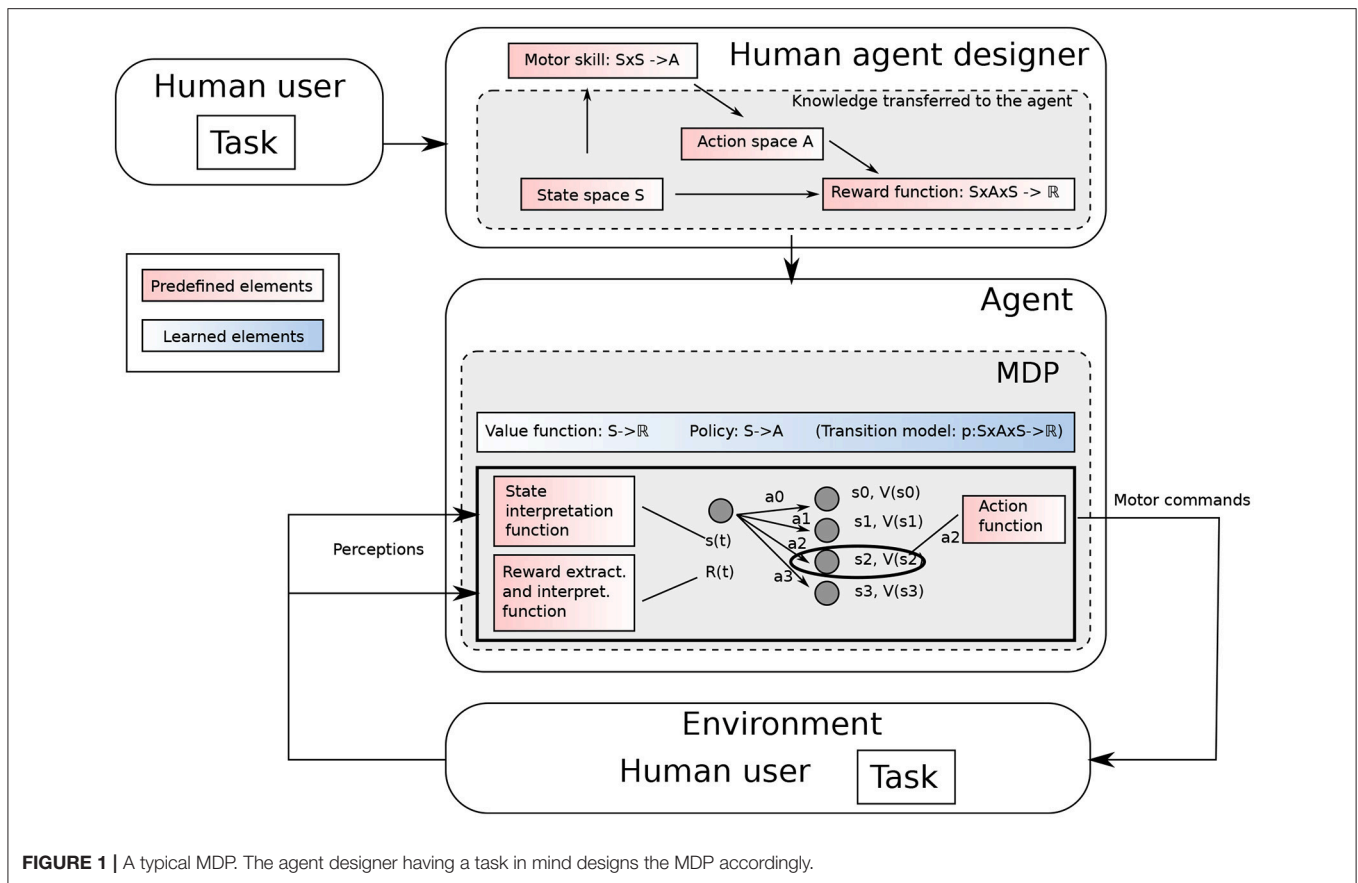
FIGURE 1 | A typical MDP. The agent designer having a task in mind designs the MDP accordingly.

DEFINITION 2. *A goal function, denoted $\hat{R}$, does not contain any shaping term and tells whether the goal associated to this reward function is achieved or not.*

A goal function is a specific reward function aimed at defining the notion of success or failure required for action definition. The task to solve does not need to be described with such a boolean function.

DEFINITION 3. *Goal states are states s for which $\hat{R}(s) = True$.*

3.4. Actions

In the proposed framework, actions are not systematically predefined, but can be built on-the-fly. The design of the corresponding algorithms requires to define what an action actually is. The proposed definition relies on the notion of goal function to add a purpose to a policy. Actions are framed within different abstraction levels depending on the granularity of the policy, as in the options framework (Sutton et al., 1999). Actions are one of the main components of an MDP. An MDP M_k needs an action space A_k. A_k is an action space defined at an abstraction level k. It relies on policies of level $k-1$, defined in an MDP $k-1$. They can be used to build policies at the level k that can, in turn, be used to build new actions for another MDP at the level $k+1$.

DEFINITION 4. *Actions $a \in A_k$ are the primitives of MDP M_k. An action a is a policy π relying on actions available at a lower level*

and built to reach a goal state associated to a goal function $\hat{R}$. The action succeeds if the trajectory of the robot controlled by this policy converges to a goal state of $\hat{R}$; otherwise, it fails. An action is then fully defined by the triplet: $\{\pi, \hat{R}, t_{max}\}$ where t_{max} is the maximum amount of time after which the action is considered failed if no goal state is reached.

If the level on top of which an MDP M_k is built is, itself, an MDP, actions $a \in A_k$ can be considered as macro-actions or options.

The goal state of an action is frequently defined relative to a particular initial state s_{init}, where s_{init} is the state of the agent when the action is triggered, e.g., "Turning $90deg$" or "moving forward $1m$."

The definition of an action is hierarchically recurrent: an action a_k relies on a policy π that also relies on a set of lower level actions $\{a_l, l < k\}$. To stop the recurrence, a specific set of actions A_0 is defined, that corresponds to the lowest level accessible by the robot, i.e., motor commands. They are also actions, as motor commands always have a goal (reaching a particular velocity or position, for instance) that a low-level control process aims at reaching and eventually staying at. As suggested by Harutyunyan (2018, Chapter 5), we assert that it may not be necessary, or even desirable, to have the same time-scale and discounting for lower and higher level actions.
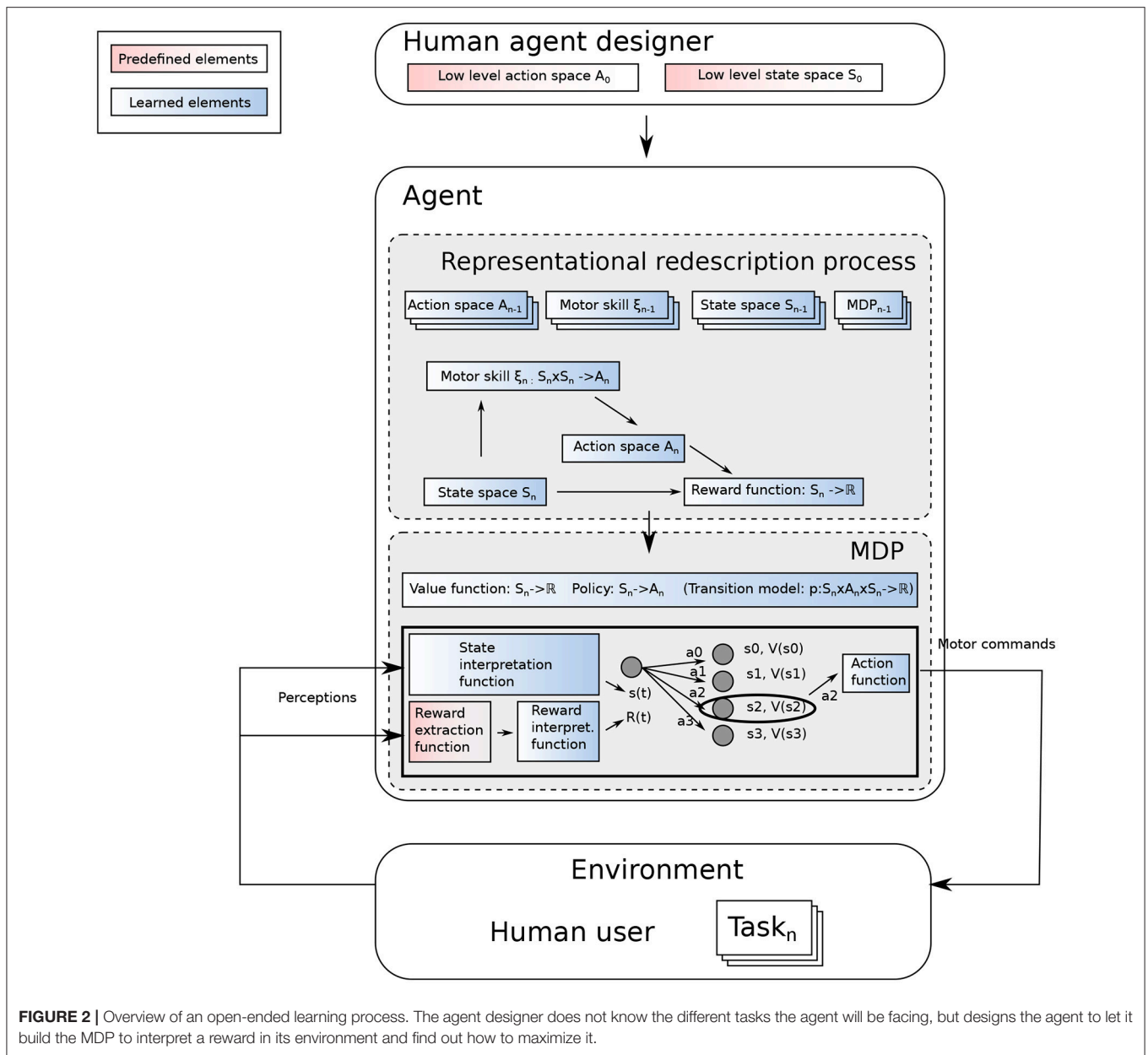
FIGURE 2 | Overview of an open-ended learning process. The agent designer does not know the different tasks the agent will be facing, but designs the agent to let it build the MDP to interpret a reward in its environment and find out how to maximize it.

3.5. Representational Redescription

In the proposed framework, open-ended learning needs to build MDPs on-the-fly, including the state and action spaces. Considering that the process starts from initial state and action spaces (S_0, A_0), this particular feature is captured by the concept of *representational redescription*.

DEFINITION 5. *A representational redescription process is a process that builds the state and action spaces enabling the definition of an MDP able to (1) solve a given task defined by observed reward values (2) in a particular domain and (3) with a particular decision or learning process. To this end, it relies on the representations of states and actions that have been previously acquired and can thus be described as a process transforming*

existing representations into new ones that are more fitted to the context.

3.6. Motor Skills: Controlling Transitions Between States

In an MDP, the set of provided actions is built to allow the robot to move in the state space. If a state space is built on-the-fly, the agent should be able to control it and move from one state to another. With the proposed definitions, the open-ended learning process needs to build actions to reach each part of the state space. The notion of motor skill is defined to capture this process.

DEFINITION 6. *A motor skill is an action generator:* $\xi_k : S^{(i)} \times S^{(g)} \to A_k$, *where* $S^{(i)}, S^{(g)} \subset S_k^2$. *It is an inverse model defined in*

a particular action space A_k to reach a target state from an initial state.

$\xi(s_i, s_g)$ is an action that, starting from $s_i \in S^{(i)}$, reaches $s_g \in S^{(g)}$ with the highest possible probability. The state s_g is the goal state of the corresponding action, and the corresponding reward function is intrinsic (see section 3.8).

3.7. Open-Ended Learning
On the basis of the proposed definitions, we can define an open-ended learning process as follows:

DEFINITION 7. *An* open-ended learning process *builds the MDPs required to solve the tasks that the agent faces. Task k is defined through an observed reward value $r_k(t)$. Starting from an initial state space S_0, an initial action space A_0 and a decision or learning process $\mathcal{P}$, the open-ended learning process aims at building (1) state spaces, (2) action spaces and (3) motor skills to control the state with appropriate actions. State and action spaces need to fulfil the following features:*

1. *The state space should help interpret the reward occurences, i.e., learn R_k to model the observed r_k;*
2. *The action space should allow control of the state space through dedicated motor skills;*
3. *The state and action spaces should make the agent's state trajectory as predictable as possible;*
4. *From the state and action spaces, $\mathcal{P}$ should be able to converge to a policy maximizing r_k.*

3.8. Intrinsic Motivations
Task-based rewards are not enough to drive a representational redescription process. There is a need for other drives that push the agent to explore and create new knowledge. This is the role of intrinsic motivations (Oudeyer and Kaplan, 2009; Baldassarre and Mirolli, 2013). In the context of open-ended learning through representational redescription, we propose the following definition:

DEFINITION 8. *An* intrinsic motivation *is a drive that complements drives associated with external task-based rewards to build appropriate state and actions spaces as well as motor skills.*

Intrinsic motivations play a critical role at different stages of the proposed representational redescription process, for instance:

- To organize learning and select in what order to learn skills and build state spaces;
- To acquire relevant data for state representation learning (before building an appropriate MDP);
- To build the skills required to control the state space (focusing learning on areas that are within reach and ignoring the rest).

4. CHALLENGES

This section recasts the challenges of open-ended learning with the proposed conceptual framework.

CHALLENGE 1. *Interpreting observed reward: Building an appropriate state space to interpret an externally provided reward, i.e., build a state space S_k that allows easy modeling of the observed reward value r_k.*

CHALLENGE 2. *Skill acquisition: Controlling the displacements in an acquired state space S_k by building the appropriate action space A_k and skill $\xi_k : S^{(i)} \times S^{(g)} \to A_k$, where $S^{(i)}, S^{(g)} \subset S_k^2$, to give the agent the ability to move from one state to another as accurately as possible.*

To address Challenge 1, state representations can be learned from known actions (Jonschkowski and Brock, 2015) and, likewise, to address Challenge 2, actions can be learned when the state space is known (Rolf et al., 2010; Forestier et al., 2017).

CHALLENGE 3. *Simultaneously learning state space, action space, and policy: The state and action spaces are interdependent with each other and with the policy. For open-ended learning, all need to be learned jointly to solve a task, and doing so tractably is a challenge.*

CHALLENGE 4. *Dealing with sparse reward: available state and action spaces may not allow to easily obtain reward $r_k(t)$ associated to Task k. This is particularly true at the beginning of the process, when starting from (S_0, A_0): this is the bootstrap problem. The challenge is to design an exploration process that converges toward reward observations in a limited time.*

A possibility to address the bootstrap problem is to rely on a motor babbling approach (Baranes and Oudeyer, 2010; Rolf et al., 2010). Another possibility would be to rely on a direct policy search including an intrinsic motivation toward behavior diversity and followed by a process to extract adapted representations from it (Zimmer and Doncieux, 2018).

The next challenges are related to the unsupervised acquisition of a hierarchy of adapted representations.

CHALLENGE 5. *Detecting task change: In the case where tasks are not explicitly indicated to the robot, detecting task change from k to $k+1$ on the basis of observed rewards r.*

The efficiency of a learning system is influenced by the order of the tasks it is facing (Bengio et al., 2009).

CHALLENGE 6. *Ordering knowledge acquisition and task resolution: An open-ended learning system needs to be able to select what to focus on and when. Does it keep learning representations for task k (even if r_k has momentarily disappeared), or does it focus on a new task $k+1$?*

CHALLENGE 7. *Identifying the available knowledge relevant to build the new representations MDP_k: as the set of available MDPs grows, it becomes a challenge to figure out what knowledge can help to build a new and adapted representation, i.e., $\{MDP_l, \pi_l\}_{l \le k} = \{< S_l, A_l, p_l, R_l >, \pi_l\}_{l \le k}$.*

CHALLENGE 8. *Using transfer learning for speeding up state and action spaces learning along time: as the number of tasks and domains the agent can deal with grows, it becomes interesting when facing a task $k+1$ to reuse the knowledge available to avoid learning MDP_{k+1} and π_{k+1} from scratch.*

5. DISCUSSION

In contrast to many works in multitask learning (Zhao et al., 2017; Riedmiller et al., 2018), we assume here that each task should be solved with its own state and action representation, and learning these representations is a central challenge. We adopt a hierarchical perspective based on representational redescription which differs from the hierarchical RL perspective (Barto and Mahadevan, 2003) from the fact that we do not assume that the lowest level is necessarily described as an MDP and we assume that each intermediate level may come with its own representation.

The proposed framework is related to end-to-end approaches to reinforcement learning (Lillicrap et al., 2015; Levine et al., 2016), but instead of black box approaches, it emphasizes knowledge reuse through the explicit extraction of relevant representations.

Open-ended learning is expected to occur in a lifelong learning scenario in which the agent will be confronted with multiple challenges to build the knowledge required to solve the tasks it will face. It will not be systematically engaged in a task resolution problem and will thus have to perform choices that cannot be guided by a reward. Intrinsic motivations are thus a critical component of the proposed open-ended learning system. They will fill in multiple needs of such a system: (1) a drive for action and state space acquisition (Péré et al., 2018), (2) a selection of what to focus on (Oudeyer et al., 2007) and (3) a bootstrap of the process in the case of sparse reward (Mouret and Doncieux, 2012).

AUTHOR CONTRIBUTIONS

This article is the result of a joint work within the DREAM project. Each author has participated to the discussions that have lead to the proposed formalism. SD has coordinated the discussions and the writing.

ACKNOWLEDGMENTS

[2]http://www.robotsthatdream.eu/

REFERENCES

Baldassarre, G., and Mirolli, M. (eds.). (2013). *Intrinsically Motivated Learning in Natural and Artificial Systems*. Berlin; Heidelberg: Springer-Verlag.

Baranes, A., and Oudeyer, P.-Y. (2010). "Intrinsically motivated goal exploration for active motor learning in robots: a case study," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (*IROS 2010*) (Taipei).

Barto, A., and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Syst.* 13, 41–77. doi: 10.1023/A:1022140919877

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, QC: ACM), 41–48.

Böhmer, W., Springenberg, J. T., Boedecker, J., Riedmiller, M., and Obermayer, K. (2015). Autonomous learning of state representations for control: an emerging field aims to autonomously learn state representations for reinforcement learning agents from their real-world sensor observations. *KI-Künstliche Intell.* 29, 353–362. doi: 10.1007/s13218-015-0356-1

Forestier, S., Mollard, Y., and Oudeyer, P.-Y. (2017). Intrinsically motivated goal exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190*.

Harutyunyan, A. (2018). *Beyond Single-Step Temporal Difference Learning*. Ph.D. thesis, Vrije Universiteit Brussel (VUB).

Jonschkowski, R., and Brock, O. (2015). Learning state representations with robotic priors. *Auton. Robots* 39, 407–428. doi: 10.1007/s10514-015-9459-7

Karmiloff-Smith, A. (1995). *Beyond Modularity: A Developmental Perspective on Cognitive Science*. Cambridge, MA: MIT Press.

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: a survey. *Int. J. Robot. Res.* 32, 1238–1274. doi: 10.1177/0278364913495721

Lesort, T., Díaz-Rodríguez, N., Goudou, J.-F., and Filliat, D. (2018). State representation learning for control: an overview. *arXiv preprint arXiv:1802.04181*.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* 17, 1334–1373. Available online at: http://www.jmlr.org/papers/v17/15-522.html

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Mouret, J.-B., and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evol. Comput.* 20, 91–133. doi: 10.1162/EVCO_a_00048

Oudeyer, P.-Y. (ed.). (2015). *IEEE CIS Newsletter on Cognitive and Developmental Systems. Representational Redescription: The Next Challenge?* Vol. 12)(IEEE). Available online at: https://openlab-flowers.inria.fr/t/ieee-cis-newsletter-on-cognitive-and-developmental-systems/129

Oudeyer, P.-Y., and Kaplan, F. (2009). What is intrinsic motivation? A typology of computational approaches. *Front. Neurorobot.* 1:6. doi: 10.3389/neuro.12.006.2007

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Trans. Evol. Comput.* 11, 265–286. doi: 10.1109/TEVC.2006.890271

Péré, A., Forestier, S., Sigaud, O., and Oudeyer, P.-Y. (2018). Unsupervised learning of goal spaces for intrinsically motivated goal exploration. *arXiv preprint arXiv:1803.00781*.

Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., Van de Wiele, T., et al. (2018). Learning by playing-solving sparse reward tasks from scratch. *arXiv preprint arXiv:1802.10567*.

Rolf, M., Steil, J. J., and Gienger, M. (2010). Goal babbling permits direct learning of inverse kinematics. *IEEE Trans. Auton. Mental Dev.* 2, 216–229. doi: 10.1109/TAMD.2010.2062511

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.* 112, 181–211. doi: 10.1016/S0004-3702(99)00052-1

Zhao, C., Hospedales, T. M., Stulp, F., and Sigaud, O. (2017). "Tensor based knowledge transfer across skill categories for robot control," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (AAAI Press), 3462–3468.

Zimmer, M., and Doncieux, S. (2018). Bootstrapping q-learning for robotics from neuro-evolution results. *IEEE Trans. Cogn. Dev. Syst.* 10, 102–119. doi: 10.1109/TCDS.2016.2628817