This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3094513, IEEE Transactions on Cognitive and Developmental Systems

Discovering Schema-based Action Sequences through Play in Situated Humanoid Robots

Suresh Kumar*, Alexandros Giagkos[†], Patricia Shaw[‡], Raphaël Braud[§], Mark Lee[‡], Qiang Shen[‡]
*Center of Excellence for Robotics, Artificial Intelligence & Blockchain (CRAIB),
Sukkur IBA University, Sukkur, Pakistan. Email: suresh@iba-suk.edu.pk
[†]College of Engineering and Physical Sciences, Aston University, Birmingham, UK. Email: a.giagkos@aston.ac.uk
[‡]Department of Computer Science, Aberystwyth University, UK. Email: {phs|mhl|qqs}@aber.ac.uk
[§]Institut de Recherche Technologique SystemX, France. Email: raphael.braud@gmail.com

***Abstract*—Exercising sensorimotor and cognitive functions allows humans, including infants, to interact with the environment and objects within it. In particular, during everyday activities, infants continuously enrich their repertoire of actions, and by playing, they experimentally plan such actions in sequences to achieve desired goals. The latter, reflected as perceptual target states, are built on previously acquired experiences shaped by infants to predict their actions. Imitating this, in developmental robotics, we seek methods that allow autonomous embodied agents with no prior knowledge to acquire information about the environment. Like infants, robots that actively explore the surroundings and manipulate proximate objects are capable of learning. Their understanding of the environment develops through the discovery of actions and their association with the resulting perceptions in the world. We extend the development of Dev-PSchema, a schema-based, open-ended learning system, and examine the infant-like discovery process of new generalised skills while engaging with objects in free-play using an iCub robot. Our experiments demonstrate the capability of Dev-PSchema to utilise the newly discovered skills to solve user-defined goals beyond its past experiences. The robot can generate and evaluate sequences of interdependent high-level actions to form potential solutions and ultimately solve complex problems towards tool-use.**

***Index Terms*—Developmental robotics, artificial play, schema-based learning, multi-modal action discovery, tool-use, iCub.**

I. Introduction

Starting in early months, play behaviours allow infants to develop their primary learning capabilities [1, 2]. Different skills related to problem-solving are shaped and sharpened, while infants build their understanding and reason about the world. As various situations are faced, skills related to previously solved problems are transferred, exploited and combined, unlocking new behaviours for cases that share similar contextual characteristics [3, 4, 5]. With active and passive interactions, infants exercise their initially premature skills to predict the outcomes of their actions. The manipulation of an object in a particular manner affects the way the object is perceived and allows associations between actions and effects in multi-modal sensorimotor spaces to be discovered. The resulting perceptions of such manipulations are then considered as a target state that the infant can achieve by inducing changes to the initial state through known behaviours and learnt skills [1].

Developmental psychologists examine situations where infants and toddlers are exposed to testing conditions and their essential learning processes [6, 7]. The results of such studies can be used to validate the extent of how humans progress from immature exploratory activities to building hierarchical knowledge and to developing skilful behaviours over their span of life. For example, experiments have shown that 2.5 year old children are capable of learning complex skills while playing with a toy machine controlled by an activator of either a certain colour or shape [6]. That is, children can extend their knowledge to interacting with a novel toy machine through developed generalisations, improving their ability of interacting with the toy machine via training or free play. Similarly, Gweon and Schulz [7] found that young children are not only able to learn complex action sequences from a single demonstration, they are also able to demonstrate or teach the sequence to others. They found that children as young as 4 years old can associate a device that produces light, an activator and a mat to place the activator on to control the light. Following a 1-minute exploratory play after being exposed to such a complex association, resulting from a single guided experience, the toddlers are found to be able to easily reproduce the sequence and demonstrate the skill to an introduced puppet, when asked. Such results reveal that toddlers can develop complex skills in a multiple-step process through active exploration and extract information useful for other learners.

To achieve high-level goal states in the environment, the sequences of actions are normally planned and executed. For instance, grasping an object on the table requires planning of moving the hand to the correct position and closing the fingers. Studies, such as [8], suggest that three to 5 year old infants actively change the hand orientation while grasping an object according to a certain desired target state. An example is performing a task involving placement of a dowel, where efficiency in the task increases with age. Similarly, it has been demonstrated that when presented with a spoon loaded with food 4 to 19 month old infants reach out and grasp the spoon with their preferred hand [9]. However, 9 month old infants may find certain initial spoon orientation challenging to handle, ending up with the handle side of the spoon in

ACKNOWLEDGEMENT: The Aberystwyth University Doctoral Training Programme supports this research, as well as the Faculty Development Program Sukkur IBA University Pakistan and the UK Engineering and Physical Sciences Research Council (EPSRC), grant No. EP/M013510/1.

their mouth instead of the food, indicating ineffective action planning and inability to predict the outcome of their actions. These studies support the idea that action sequences can affect the given state of the environment, leading to a desirable end state or goal state. It is also found that repetitions of action sequences are a natural way to refine the understanding of objects and to increase confidence in performing a sequence of actions as a single smooth compound action [10].

To investigate how a robotic system may be facilitated to perform open-ended learning of new skills and develop its knowledge we extend Dev-PSchema, a schema-based developmental learning mechanism that allows robots to autonomously discover associations between actions and objects through intrinsically motivated active explorative play [11]. Driven by the inner desire to act towards novelty, Dev-PSchema collects sensory observations that reflect changes in the scene before and after the execution of an action. These observations are used to identify when habituation based on repeated stimulation occurs, leading to schema generalisations that can be transferred into novel situations [12]. Thus, generalising schemas allows the robot to acquire knowledge that is both abstract and reusable.

We address the problem of learning via action sequences, and we demonstrate skill development through schema chains in Dev-PSchema. The system allows an embodied agent to acquire new skills by discovering combinations of primitive actions in a developmentally plausible fashion. Employing action of schema chains leads to more productive exploratory behaviours under novel scenarios, where the robot gradually as well as autonomously expands its knowledge and enriches its repertoire of high-level skills. In our previous work, we demonstrated action sequences using concrete schemas only. Generalised schemas were used to extend single actions because they could provide detailed descriptions of concepts related to contingencies, i.e., a generalised grasp schema learnt from different concrete instances would expect the touch sensation when an object is grasped. However, they did not participate in creating chains, thus could not be part of high-level actions that could be re-utilised in novel scenarios.

In this manuscript, we document further developments that allow the agent to create chains using concrete and generalised schemas, rendering contingencies related to particular objects (i.e., concrete schemas) applicable to novel scenes with unfamiliar objects. Therefore, the chaining mechanism has been extended to process generalised schemas, enabling the agent to explore further interactions with the world and the objects within it whilst playing. This new way of creating sequences of actions extends the agents ability to solve complex problems by considering alternative solutions. We describe the learning process and discuss results that show how behaviours emerge, not only while learning but also when the agent utilises its experiences to solve novel problems. An experiment is also reported that examines the potential of such a robot to discover associations between objects, actions and their effect in the world; developing an understanding towards tools-use.

The remainder of this paper is organised as follows. Section II discusses existing studies on action planning and action sequences. Section III describes the low-level mechanism for acquiring perceptions and developing primitive actions, and Section IV presents the implemented Dev-PSchema system with an emphasis on the underlying techniques built for schema chaining and problem-solving mode, enabling the agent to achieve a user-defined target state in a complex environment. The experimental methodology is found in Section V, followed by Section VI where the obtained results are analysed. A conclusion is given in Section VII.

II. Related Studies

Investigating robotic systems capable of learning actions that can be performed in sequences, either by passively observing or actively exploring, has been the subject of various works in the literature. In [13], an extended framework that learns semantic event chains as representations of object manipulations is demonstrated. The framework deals with the analysis of a sequence of changes that are observed interconnected between while an acting agent manipulates them. Aligned with the Piagetian theory [1], changes are semantically compared to already memorised event chains in order to determine whether novel actions or elements of known actions are discovered. Trajectory information encoded with modified Dynamic Movement Primitives is used as one of the event descriptors for comparison. The framework is capable of reasoning about similar events and of classifying cutting, chopping and stirring actions. It abstracts related actions to represent their common characteristics within its memory structure. However, its learning depends on a large number of observations being made, that has a negative effect on its scalability.

Manoury et al. [14] proposed a skill learning algorithm and demonstrated its performance using a simulated mobile robot. The algorithm combines two important aspects of developmental learning, namely a strong association between visual cues and motor capabilities of the embodied agent, and the intrinsic motivation to guide its exploration. Starting with a set of primitive actions (e.g., motor commands sent to the actuators of the robot), the algorithm associates actions with the outcomes of their consciences. Furthermore, it employs a goal babbling action that allows the robot to set a goal to attain. Results from the evaluation of the proposed algorithm demonstrate that the virtual mobile robot can discover non-predefined affordances. Although not evaluated in the real-world and lacking the ability to suggest alternative solutions to a given problem, the algorithm was designed to rely on visual classifiers extracted from sensory information to favour generalisation. The agent running this algorithm has the ability to update existing associations or create new affordances.

Different from infant playing, a goal-discovering robotic architecture for intrinsically-motivated learning is presented in [15]. Goals are formed by capturing the effect of events in the environment as changes to the visual input. At the selection layer, goals are selected to drive the autonomous exploration. Then, the control layer activates an expert, implemented with a neural network trained to utilise the appropriate actuator, resulting in a solution towards the goal. The proposed architecture has been thoroughly evaluated, demonstrating its ability to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3094513, IEEE Transactions on Cognitive and Developmental Systems

learn simple skills such as reaching towards interesting targets, in an open-ended approach. The authors highlight the need for complex skill acquisition and a hierarchy of simple skills that combined can offer high-level actions capable of being applied to novel scenarios.

Employing a goal-babbling approach, Forestier and Oudeyer [16, 17] proposed modular and hierarchical, active curiosity-driven model babbling architectures. Their work was inspired by the intrinsic motivation for driving spontaneous exploration in infant free-play to build hierarchies of representations of a robot's world. The architectures render the associations between multi-dimensional motor and sensory spaces possible, while the robot interacts using motor primitives with the environment, to reach self-generated goals. Although the results from the two curiosity-driven architectures illustrate autonomous acquisition of skills related to objects in the scene, skill generalisation is not addressed, differing from Dev-PSchema, the learning mechanism of this paper.

Learning action sequences is also reported in [18], where a simulated robotic system can recognise and learn sequences of object interactions and their effects from a demonstration. High-level skills are translated into low-level operations that become available to the learning mechanism. Extracted visual frames allow the identification of abstract concepts and their associations within the given environment before and after the manipulation is performed. Whilst the system can learn abstractions of new skills from demonstration similar to being described in natural language, the properties of event and object descriptions have to be set manually.

Techniques for planning sequences of actions that result in solutions to goals described by human instructors are also presented in [19]. The work bridges the interpretation of instructions given in a natural language with the execution of low-level motor commands that deal with object manipulation. The system learns by analysing the verbal input given by the trainer and generates a probabilistic plan of actions. The rules and symbols, being the important ingredients that symbolically describe actions and object affordances, are combined with the system's past experiences and the current state of the world. The system is able to plan actions and to react against the environment by manipulating objects thereby delivering the desired outcome but it requires user-defined task to be specified.

Singh et al. [20] presented an intrinsically motivated reinforced learning mechanism employing a semi-Markov decision process (SMDP). An agent autonomously learns new skills through random interactions with objects in the environment. However, those interactions with objects are prolonged if they produce salient events. The mechanism allows only selected actions related to the current world state, i.e., a kick action can be a candidate only after a ball object is reached. Furthermore, the agent's interest in exploring actions related to particular objects may only decay over time, rendering habituation irreversible. Under these conditions, the exploration for the discovery of novel phenomena is limited, e.g., an agent can never learn that the kick action can be applied to a button when reached by the manipulator. Furthermore, the proposed system does not explicitly address generalisation, implying that every learnt skill is only applicable to particular objects, thus, rendering its application limited in novel situations.

In a very similar approach, Santucci et al. [21] demonstrated the performance of the open-ended Goal-Discovering Robotic Architecture for Intrinsically-Motivated Learning (GRAIL). The proposed system discovers interesting events while it interacts with the environment and sets "goals", which are later used to drive the learning in an intrinsically motivated manner. The architecture builds separate skills to achieve each goal, focusing on achieving the highest overall competence (i.e. reliability) for all skills as fast as possible. However, the architecture resides in the fact that although it can select (and learn) hierarchical tasks, it cannot retain these "chains" after the learning process has completed. As such, they cannot be selected and, in turn, performed as new high-level skills. Moreover, the system lacks generalisation; hence new sets of skills need to be learned for every new object.

Constructing Skill Trees (CST), a learning from demonstration (LfD) algorithm, is presented in [22]. CST uses a demonstrated task to generate a sequence of skill, which later can be reused in other sequences. Each trajectory is divided into segments representing skill abstractions that, using calculated probabilities, can be chained and merged to form novel sequences of actions. Thus, learning in this LfD approach relies on the existence and effectiveness of the demonstrated trajectories. On the contrary, Dev-PSchema is based on gradually scaffolding the learning of novel skills through open-ended exploration (free play). This bottom-up approach starts by discovering the building blocks first, i.e., it moves from concrete, object-specific actions to generalised ones that constitute the action chains.

PSchema, an open-ended learning system that employs a practical implementation of the Drescher's schema mechanism [23] is introduced in Sheldon [24]. Schemas encapsulate information about actions and perceptions, built as the result of an agent's interactions and their effect on the world. A network of world states is constructed from which PSchema identifies schema chains (i.e., schemas that have linked pre-conditions and post-conditions) using an implementation of Dijkstra's algorithm. Paths are weighted, based on their probability of success, and a method to determine the shortest chain of actions is required to achieve a goal. Although chains of actions can occur in PSchema, they are not considered as new actions because they do not result from autonomous, intrinsic exploration. Instead, they are built outside the open-ended learning process as a response to a user-defined goal, rendering them highly dependent on historical data and statistics based on human choices. As such, they are neither kept in the memory nor considered in future exploratory play behaviours by the agent.

The aforementioned works deal with the development of high-level action sequences that allow a robotic agent to solve complex manipulation problems. Here, we describe a novel chaining mechanisms of Dev-PSchema; a open-ended learning tool based on PSchema (which was firstly introduced in [25] and [26]) to extend its problem-solving capability. It is designed to discover a set of possible solutions capable of achieving the desired target state of a given problem through artificial play. That is, a process during which the agent executes previously learnt actions and sequences of actions in an attempt to determine their potential in reaching the target state. Focusing on building sequences, the mechanism of schema chaining is herein discussed as a means of novel action discovery for actions that would not be able to emerge otherwise. Instead of finding and executing action sequences when a human observer provides the target state (as in the case of PSchema), here they are an integral part of the system; subject to an excitation mechanism that intrinsically drives the robot towards the discovery of new ways to interact with in a given environment.

III. Acquiring perceptions and primitive actions

The architecture of the system implemented in this work consists of two interconnected subsystems: a low-level sensorimotor controller and a schema-based decision generator, facilitated by Dev-PSchema. The former is responsible for i) learning multi-modal object perceptions and primitive actions, and ii) their interaction with Dev-PSchema. Primitive actions constitute the repertoire of skills the robot has to interact with the environment, and the object perceptions allow it to learn and recognise familiar objects in the scene. Once learnt, the object perceptions are communicated to Dev-PSchema in order to describe the state of the world before and after one or a sequence of actions is performed. To implement the latter, the decision generator, Dev-PSchema employs the primitive actions offered by the low-level subsystem via sending appropriate requests to fixate, reach or grasp an object, etc. In the rest of this section, the two subsystems are described, starting from the low-level object perception and primitive action mechanisms.

A. Object perceptions

Understanding objects requires a combination of visual and haptic perceptions of them in the scene. In this work, visual representations are referred to as proto-objects, consisting of salient features that share a consistency of motion. According to Casati [27], a proto-object is an operational object that, throughout visual tasks, can be traced while moving in front of a static background.

Extracting and identifying salient features begins in the retina of the robot's camera, with images being processed using the method described in [28]. The extracted features are stored as fields into maps that represent different feature spaces, i.e., colour, brightness, motion and edges [29]. Each feature field in a map has a type and a unique ID, as well as a radius. The latter plays an essential role in triggering the same field when a target area with close related properties is identified. The larger the field radius, the less accurate the system becomes in differentiating between features. Note that the smaller the radius, the less overlap between stimulated areas in a map that represent adjacent yet different features. On the contrary, an increased feature field radius makes the system more tolerant to environmental noise. Feature information stored in feature maps is used to build visual representations, in the form of graphs of paired features that share the characteristic of motion, based on the assumption that features that co-occur on the retina are most likely to be associated with the same animated object [30]. Note that multiple instances of the same feature may appear.

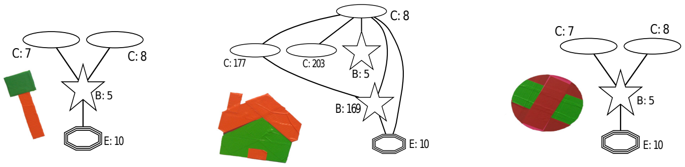


Fig. 1: Proto-object graphs as visual representations of object perceptions. Oval, stars and octagons represent colour, brightness and edges respectively with unique feature IDs also depicted.

To identify salient feature instances on the retina, the system considers the changes to the shortest distance of all identified feature instances between consecutive images. Those features are paired and linked together to form proto-object graphs, as shown in Figure 1. The latter are used to represent learnt objects in memory and to match with what is currently observed. Note that when a proto-object graph in memory best matches an observation, it is selected for recognition. Ultimately, Dev-PSchema receives the following information: i) the unique ID of the proto-object graph, ii) the types of features (i.e., colour, edges, etc.), IDs and values of all features that form the graph, and iii) the average x and y coordinates of them, thus the position of the object in the gaze space. The implementation details of the proto-object recognition are found in the supplementary material.

B. Primitive actions

Several primitive actions are desired to be available for execution, namely, reaching, grasping, releasing and pushing. They are either learnt or designed to mimic infant behaviour.

To develop reaching, the system undergoes a learning process by which progressive associations between the hand position in the robot's gaze space V and its arm's motor space M are drawn. This process is inspired by the behaviour of hand regard, which is met in infants roughly between the second and the 5^{th} month [31]. The behaviour is observed as the hand attracts much attention and monopolises visual exploratory efforts, while it manoeuvres within the infant's narrow field of view.

Learning associations between V and M, is achieved by the robot sending motor babbling commands to the arm and observing the visual changes on the retina. Note that while the hand moves in V, it is visually located by a coloured marker that allows the robot to fixate its camera on it, determining its gaze space position. The triggered gaze field is linked with the associated motor field in M that represents all motor values of the hand movement. The resulting map associations are bidirectional; motor commands can derive from reaching the desired position in V. The more populated the two maps are, the more information is available for the reaching system to calculate motor commands. Being developmentally plausible, if V and M are not well populated (as in the early stages of the infant), remote motor fields will be considered. Subsequently, this leads to non-refined hand trajectories while reaching. In-depth discussion about map calculations in the context of reaching can be found in [32, 33].

To grasp, the system employs the palmar grasp reflex mechanism as evaluated in [34]. Tactile and proprioceptive sensory information is combined to perform a power grip on an object. During grasping all fingers close reflexively until no further movement is possible. The final hand posture reflected by the encoder values is measured and stored in the hand motor space. Note that if a grasped object has previously been recognised as a proto-object, the association between the proto-object and the grasping pattern is possible. In this work, after the robot grasping an object, the following information becomes available to Dev-PSchema: i) the hand ID (to distinguish between left and right), ii) the proprioceptive grip information normalised between 0-1 (0 being fully open, 1 being fully closed) and iii), the x and y coordinates of the hand in the gaze space.

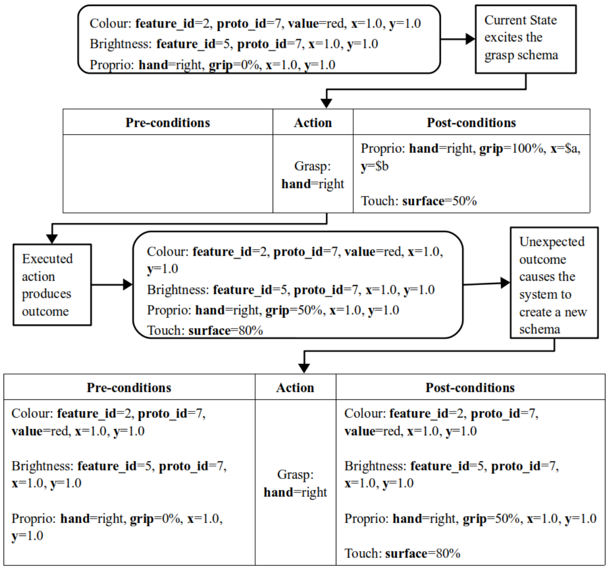
The system supports a release action as a reverse of the action grasping described above. During a release, all digits gradually open till they reach a configuration of a fully opened hand. Lastly, a primitive action of push is available, such that when requested, the agent rotates the wrist joint for its pushing hand for the palm to reach a vertical position with the torso towards the opposite direction of the pushing hand. For example, if the system requires to push using the left robotic hand, the torso will rotate to the right allowing the left hand to push. The torso returns to its initial position, followed by the hand's wrist joint.

IV. Dev-PSchema & schema chains

Dev-PSchema is a play generator that, when interfaced with an embodied agent, drives the agent's exploration while facilitating egocentric learning through sensorimotor experiences. It is designed following the sensorimotor stage in Piaget's cognitive theory [1]. Whilst the agent interacts with the environment, blocks of knowledge or schemas that connect actions and sensory information are recorded. Schemas consist of pre-conditions, i.e., sensory information before an action, an action and the post-conditions, i.e., sensory information after an action. The sensory information in pre-conditions and post-conditions is described as a high-level perceptual representation of the environment, provided by the low-level mechanism. As such, the schemas can be exploited to synthesise solutions to complex manipulation problems. The sensory information used as the pre-conditions or post-conditions in a schema is recorded in discrete time-steps before and after an action respectively. The time-step in the system is updated after an action execution, helping to calculate statistics for schemas and schema chains. Statistical properties are used in the mathematical model of the excitation mechanism, as to discussed later in Section IV-B.

A. Bootstrap, concrete and generalised schema generation

Learning is facilitated by initially using basic primitive actions provided in the shape of bootstrap schemas. They are analogous to reflex behaviours an infant can perform without considering any aspect of the environment (i.e., a reaction to stimuli) [1]. Bootstrap schemas only contain actions and the post-condition that consists of proprioceptive information. As such, bootstrap schemas represent basic motor commands that the agent can perform. Using these schemas, the agent interacts with its environment and builds higher-level schemas, containing pre-conditions, action and post-conditions.



Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=1.0, **y**=1.0
Brightness: **feature_id**=5, **proto_id**=7, **x**=1.0, **y**=1.0
Proprio: **hand**=right, **grip**=0%, **x**=1.0, **y**=1.0

Current State excites the grasp schema

Pre-conditions	Action	Post-conditions
	Grasp: **hand**=right	Proprio: **hand**=right, **grip**=100%, **x**=$a, **y**=$b Touch: **surface**=50%

Executed action produces outcome

Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=1.0, **y**=1.0
Brightness: **feature_id**=5, **proto_id**=7, **x**=1.0, **y**=1.0
Proprio: **hand**=right, **grip**=50%, **x**=1.0, **y**=1.0
Touch: **surface**=80%

Pre-conditions	Action	Post-conditions
Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=1.0, **y**=1.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=1.0, **y**=1.0 Proprio: **hand**=right, **grip**=0%, **x**=1.0, **y**=1.0	Grasp: **hand**=right	Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=1.0, **y**=1.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=1.0, **y**=1.0 Proprio: **hand**=right, **grip**=50%, **x**=1.0, **y**=1.0 Touch: **surface**=80%

Fig. 2: Example of building a concrete schema from a bootstrap one.

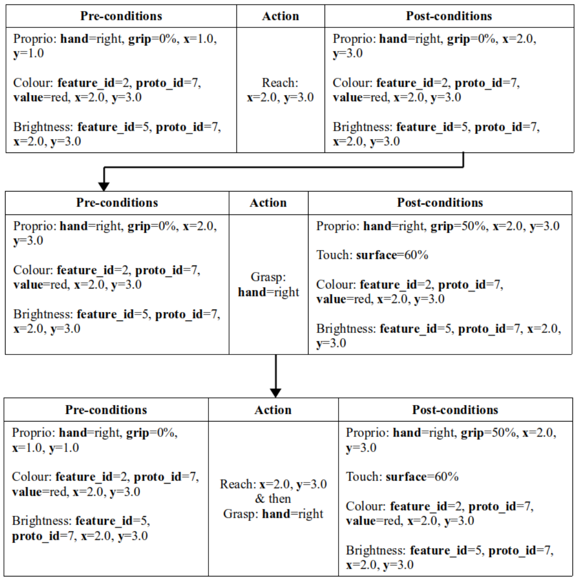
Figure 2 demonstrates how the perceived state of an environment (referred to as a world state) triggers the grasp bootstrap schema for execution. The result is the creation of a new concrete schema, containing all the differences between the actual world state and the expected outcome found in the bootstrap schema. In a concrete schema, all sensory perception properties have fixed values and, unlike a bootstrap schema, it contains pre-conditions and post-conditions. Concrete schemas are therefore used to associate specific actions with specific objects, under specific circumstances. Aligned with Piaget, concrete schemas reflect the ability of infants to represent the world around them in detail, an important stage to reach before turning in-depth into generalised knowledge.

In a world of both static and dynamic objects, there exists evidence which indicates that children are capable of predicting changes in the environment using previously learnt experiences [35, 36, 37]. The anticipation of dynamic, symmetric and asymmetric visual events, in turn, leads to the development of generalisations for objects, events and situations that infants experience while playing. Following these concepts, Dev-PSchema models generalisation of schemas through inference after performing schemas of similar actions, and by monitoring changes to associated the pre- and post-conditions. Any property, present in the pre-condition and post-conditions, is generalised if two concrete examples of the property are given in two schemas of the same action type. A schema is labelled as a generalised schema if at least one of the properties present in the pre-condition or post-conditions, is generalised.

Generalised schemas are used to anticipate action outcomes within similar environments; that is, situations where similar objects are present. Generalised schemas can form part of (schema) chains, leading the agent to achieve complex goal states in novel situations. This mechanism is also modelled on infants' behaviour of generalising their actions [4, 38, 39].

The low-level mechanisms of generalisation were initially introduced in [11, 12], enabling Dev-PSchema to perform manipulative actions to novel objects which share similarities with the objects that the agent experienced previously. Note that generalisation satisfies two requirements; i) it renders existing knowledge applicable to both novel and similar scenes via generalised schemas, and ii) it limits the number of concrete schemas in the system [24]. For a generalised schema to be used, the system first needs to instantiate it, that is, to replace its generalised properties with actual (concrete) values taken from the currently observed environment.

Developmental psychology provides extensive evidence for sequences of actions to be planned in order to execute a single high-level action [8, 9, 40, 41, 42]. Dev-PSchema models the development of high-level actions in the form of schema chains from primitive actions, by finding links between post-conditions and pre-conditions of different schemas. High-level actions are used to achieve a distant sensory state which is, otherwise, not achievable by a single action. For example, holding an object from an initial state where the hand of the robot and the object are at different positions, may result from an action chain of reach and grasp. Figure 3 demonstrates how two different schemas, i.e., reach and grasp, are used to develop a chain by considering the post-conditions of the former and the pre-conditions of the latter.



Pre-conditions	Action	Post-conditions
Proprio: **hand**=right, **grip**=0%, **x**=1.0, **y**=1.0 Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=2.0, **y**=3.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=2.0, **y**=3.0	Reach: **x**=2.0, **y**=3.0	Proprio: **hand**=right, **grip**=0%, **x**=2.0, **y**=3.0 Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=2.0, **y**=3.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=2.0, **y**=3.0

Pre-conditions	Action	Post-conditions
Proprio: **hand**=right, **grip**=0%, **x**=2.0, **y**=3.0 Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=2.0, **y**=3.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=2.0, **y**=3.0	Grasp: **hand**=right	Proprio: **hand**=right, **grip**=50%, **x**=2.0, **y**=3.0 Touch: **surface**=60% Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=2.0, **y**=3.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=2.0, **y**=3.0

Pre-conditions	Action	Post-conditions
Proprio: **hand**=right, **grip**=0%, **x**=1.0, **y**=1.0 Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=2.0, **y**=3.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=2.0, **y**=3.0	Reach: **x**=2.0, **y**=3.0 & then Grasp: **hand**=right	Proprio: **hand**=right, **grip**=50%, **x**=2.0, **y**=3.0 Touch: **surface**=60% Colour: **feature_id**=2, **proto_id**=7, **value**=red, **x**=2.0, **y**=3.0 Brightness: **feature_id**=5, **proto_id**=7, **x**=2.0, **y**=3.0

Fig. 3: Example of creating a schema chain.

B. Schema excitation mechanism

Babies spend most of their awake time interacting with the environment by manipulating surrounding objects. Their actions are not constrained by predefined rules, other than those that relate to physical capabilities. To guide such an active exploration, Dev-PSchema employs an excitation mechanism where agents explore their environment through intrinsically motivated actions. The action selection depends on measurements of both object-related and schema-related data found in memory. It includes the number of times an object has appeared in the environment (similarity S), the number of times it has been used in schemas (novelty N), and the success rate and frequency of use of those schemas in memory (habituation H). These concepts are fully described in [25] and modelled as shown in the supplementary material.

Besides, for any schema whose action does not require coordinates to be performed (i.e., a grasp, release or push) a penalty is applied, when the hand of the agent is not located at the object's location. A penalty of 50% is used in the present implementation. This mechanism ensures that actions unrelated to the current circumstance have far fewer chances to be selected, but still constitute options that encourage any further action exploitation.

In summary, the excitation system described in this work enables the agent to explore the environment and to develop its own learning. While in the problem-solving mode, the robot is provided with a goal state, given by a human observer, that needs to be achieved through a combination of actions. Utilising any previously learnt knowledge, the robot may suggest several solutions with different probabilities of success in leading it to the desired goal state.

C. Problem-solving using schema chains

Learning is an ongoing, continuous process, by which the agent is accumulating experiences and scaffolding knowledge. Having the excitation mechanism in its core, the intrinsic exploratory behaviour modelled above allows the robot to create, combine and evaluate schema-based sequences. Expanding the previous system [25], we designed a problem-solving mechanism capable of producing multiple alternatives, while taking advantage of the extended chaining possibilities and thus the agents learning capacity. Figure 4 depicts the interactions between the user and the autonomous agent. The former provides a target by describing the desired world state, i.e., a descriptive perceptual goal state for the agent to achieve. Such descriptions contain the high-level perceptual representations similar to the pre-conditions and post-conditions of in schemas. In turn, the agent utilises the schemas in memory and generates schema chains that can lead to the state's goal. The algorithm and implementation details related to the chaining mechanism are given in the supplementary material. Note that the mechanism is extended to utilise generalised schemas in problem-solving within novel scenarios.

Multiple solutions may exist, therefore a chain excitation calculation mechanism is necessary, as shown in Algorithm 1. This mechanism calculates the excitation level of a given chain C and thus, the suitability of each chain in achieving a target state. Two factors are considered: i) the similarity between the pre-conditions of the chain's first schema and the current world state, and ii) the average success rate of all schemas in the chain. The first factor examines the degree

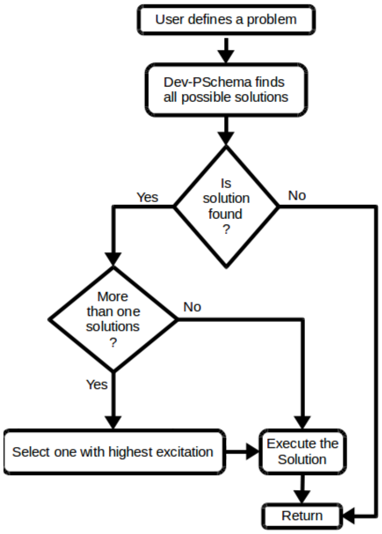

User defines a problem
Dev-PSchema finds all possible solutions
Is solution found ?
Yes
No
More than one solutions ?
No
Yes
Select one with highest excitation
Execute the Solution
Return

Fig. 4: Flow chart of problem-solving phase.

to which a chain is suitable for a currently perceived scene. In contrast, the second factor quantifies the robot's ability to execute the chain as a single high-level action successfully, through considering the success rate of each schema in the chain. Two constants, K_{sim} and K_{rate} are used to produce a weighted sum of the two factors. To favour those chains that are more relevant to the current circumstances, K_{sim} and K_{rate} are considered 0.7 and 0.3 respectively. Ultimately, the overall calculation is multiplied by C_{rate}, the ratio between the number of successful executions over the total times the chain has been executed. A newly created chain is guaranteed to have $C_{rate} = 1$ to encourage further exploration. As chains are executed, their statistics change, rendering the long-term memory able to be continuously developed.

Algorithm 1 Chain excitation calculation for problem-solving

```
Require:
    Current world state $WS_c$ and a chain $C$
    List $mem$ of all schemas in memory
Ensure: Excitation value for chain $C$
function CALCEXCITATION($WS_c$, $C$, $mem$)
    $C_{sim} \leftarrow GetSimilarity(C.first.pre, WS_c)$
    $s_{rate} \leftarrow 0$
    for each $s \in C$ do
        $s_{rate} \leftarrow s_{rate} + GetSuccessRate(s)$
    end for
    $s_{rate} \leftarrow s_{rate}$ / $|C|$
    $C_{rate} \leftarrow GetSuccessRate(C)$
    return $\Big((K_{sim} \times C_{sim}) + (K_{rate} \times s_{rate})\Big) \times C_{rate}$
end function
```

V. Experimental methodology

The experiments we present in this work are designed to investigate the ability of Dev-PSchema to discover new actions and thus to acquire new object manipulation skills. The iCub humanoid robot [43], equipped with Dev-PSchema and a low-level sensorimotor control system as described in Section III, is employed as the learning agent, placed in front of a table. The low-level mechanism is responsible for the learning and recognition of objects as the robot interacts with them, and for the feeding of such information to Dev-PSchema. It also provides several primitive actions that can be used to perform object manipulation, ranging from reaching and grasping to releasing and pushing objects. Note that apart from the push action, which consists of a fixed rotation to the vertical axis of torso allowing the extended arm to push objects, all other primitive actions are learnt as discussed in [33].

The system favours the discovery of new actions and their associations with objects, rather than the discovery of new objects in the scene. This is encouraged by excitation parameters that increase the system's excitation related to actions' effects rather than object perceptions ($\omega_1 = 0.5$, $\omega_2 = 0.5$, $\omega_3 = 0.3$ and $\omega_4 = 0.7$). By changing these parameters, the agent's behaviour can lean towards either the discovery of new objects or new skills. Thus, changing the values to favour new objects will delay the discovery of new actions. Nevertheless, in a complex environment, full of novel objects, the agent is expected to spend more time trying out existing actions on the same objects.

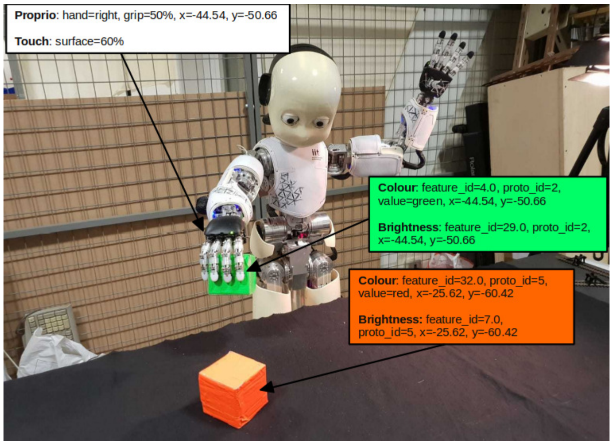

Proprio: hand=right, grip=50%, x=-44.54, y=-50.66
Touch: surface=60%
Colour: feature_id=4.0, proto_id=2, value=green, x=-44.54, y=-50.66
Brightness: feature_id=29.0, proto_id=2, x=-44.54, y=-50.66
Colour: feature_id=32.0, proto_id=5, value=red, x=-25.62, y=-60.42
Brightness: feature_id=7.0, proto_id=5, x=-25.62, y=-60.42

Fig. 5: iCub performing a reach action with associated visual sensory information sent from the low-level mechanism to Dev-PSchema.

The experiments carried out consist of a learning and a problem-solving phase. During learning, the robot is allowed to freely interact with objects by initially selecting an action from the set of primitives. Initially the Dev-PSchema's memory consists of bootstrap schemas that enable the system to perform the available primitive actions, containing only proprioceptive post-conditions. Thus these schemas can be selected under any pre-condition. Any observable changes in the environment are monitored and communicated to Dev-PSchema, where the chaining mechanism gradually facilitates the discovery of interesting action patterns and combinations in order to manipulate objects. A human observer is present but does not interfere with the learning process. Instead, they introduce or remove objects from the scene to enrich visual stimuli and to trigger the agent's excitation mechanism. Throughout the play, i.e., freely interacting with objects in the environment, new chains are generated by combining primitive and/or previously learnt chains in the system. In this case, a new skill is deemed to have been successfully acquired, and the human observer resets the scene to speed up the learning process.

The problem-solving phase commences when learning has progressed, and chains of actions have been generated to represent the emergence of new high-level skills in the robot's repertoire. A human observer defines a target state which the robot must achieve by utilising what it has previously learnt, that is, synthesising primitive and high-level skills to meet a user-defined complex goal. Figure 5 depicts an example of a state of the environment provided by the low-level mechanism to Dev-PSchema.

We also investigate the extent to which Dev-PSchema is capable of discovering object affordances through play. An experiment is therefore designed to examine how associations between actions and objects in the environment are learnt and, consequently, are used as solutions to the novel, complex problems. The iCub starts with primitive and high-level actions that it has learnt from the first experiment, while the presence of the human observer is active, causing changes to the scene that scaffold the learning of the robot. Following a simple approach for developing an understating towards tool-use, the human observer introduces three objects: a red cube, henceforth referred to as the trigger object; a green cube, referred to as the non-trigger object; and a bi-colour cube, referred to as the toy object that is introduced after moving the trigger object towards a particular position. The experiment situation can be considered analogous to a switch and light scene, where an action on the switch makes the light turn on.

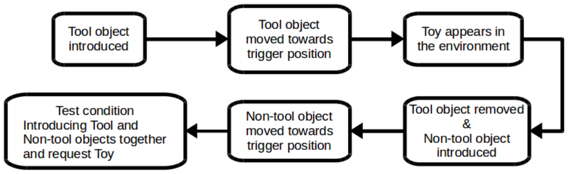

Tool object introduced
Tool object moved towards trigger position
Toy appears in the environment
Tool object removed & Non-tool object introduced
Non-tool object moved towards trigger position
Test condition Introducing Tool and Non-tool objects together and request Toy

Fig. 6: Information flow in learning "object-object-action" associations.

The scenario of the this experiment is as follows. The robot familiarises with both trigger and non-trigger objects equally and is allowed to try any form of interaction that is found to be most exciting at every interaction step. As a special event, the human observer introduces the toy object in the scene, if and only if the trigger object is moved to a specific location in the gaze space of the robot (henceforth referred to as trigger position). Figure 6 illustrates the information flow within the scenario. The other experiment is for the robot to discover a typical light switch property, e.g., a bulb emits light when the switch is on. As such, a user-defined target state, i.e., finding a method to make the toy appear anywhere in the scene, is given to the robot in an effort to evaluate the usability of its learning experiences and their adaptability to dynamic novel situations.

VI. Experimental Results

A. Emergence of high-level skills

This section presents and discusses experimental results on both the process of autonomous high-level skill learning by play and the emergent behaviours of the system suggesting alternative ways to solve manipulation problems. We also report the process of learning associations with substantial potential for the discovery of object affordances.

The iCub is placed in front of a table, and its memory contains only primitive schemas. The bootstrap schemas are neither generalised and hence can be used to objects that partially share characteristics) nor contain any pre-conditions. Allowed to freely interact with objects on the table according to the excitation levels calculated by Dev-PSchema, the robot tries possible actions available and discovers new actions, which are represented as generalised combinations of schemas already known and which are then added to the memory. Note that new action schemas are repeatable and applicable to objects that share known characteristics and can, therefore, be employed to solve user-defined problems.

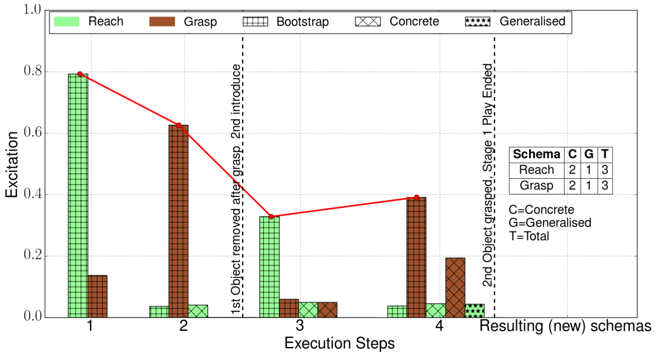


Fig. 7: Play behaviour to develop generalised "reach $" and "grasp $" schemas, where concrete reach and grasp and generalised reach and grasp schemas are filled with similar patterns.

1) Discovering a seize action: Initially presented with only a red cube, the agent selects a bootstrap reach action, placing a hand to the object's position. The corresponding visual and proprioceptive stimuli of the object and the hand are subsequently stored to a concrete reach schema for the red cube. The bootstrap grasp action is then selected after the reach action. Its success adds tactile sensation to the object perception resulting in the generation of a concrete grasp schema that reflects the experience. The human observer then removes the red cube from the scene and introduces a green cube. The new novel object triggers the robot's excitation mechanism to execute a bootstrap reach action towards it, followed by a bootstrap grasp. Like before, the success of these actions results in the generation of concrete schemas for the green object. The generalisation mechanism is then activated, and a generalised schema for each action is generated and stored in memory. Knowing how to reach and grasp cube objects of any colour but similar edge and brightness information, the chaining mechanism is triggered for the green cube. In particular, the post-condition of a generalised reach action towards an object, including the agent's hand position (proprioceptive perception of the hand), becomes the pre-condition to generalised grasp of the object. These form a generalised chain schema that, if performed, it will phenotypically reflect a seize action. Figure

7 depicts the sequence of actions along with their excitation levels that the agent performed to learn the generalised reach and generalised grasp schemas that constitute a high-level seize action.

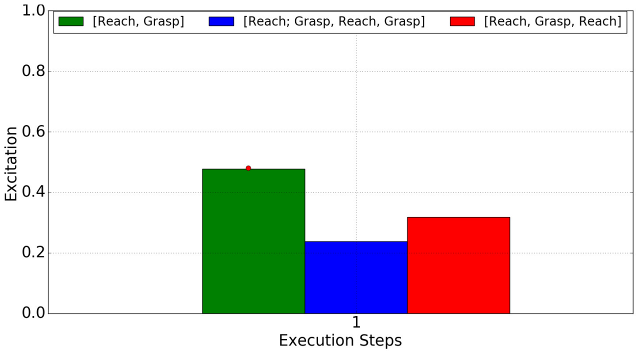


Fig. 8: Suggested chains and their excitation levels to grasp an object.

After resetting the scene, the human observer presents a novel cube in front of the robot, the user-defined problem becomes to seize the unknown object on the table. In the system's context, that is given in the form of the desired post-condition: "hand at novel cube's position, with closed grip and touch sensation, received". Figure 8 depicts the result of searching for alternative solutions to meet such requirements. From the current repertoire of the robot's skills, the emergent seize action is given the highest excitation level. Note that the reach actions after a grasp in the second and third suggested chains refer to a reach action to the same position, as the robot has never experienced reaching to another position while grasping an object yet. This phenomenon is due to over-generalisation in reach and grasp schemas, since a generalised property is easily linked (after instantiating with the same value) between pre-conditions and post-conditions. That is, over-generalisation may occur as a result of excessively abstracting schema properties based on fewer past experiences. Due to their level of granularity, the system leniently deems those schemas suitable matches to the world states. This is aligned to the findings of developmental psychology in that infants and young children are found to over-generalise their understanding of actions and the associated outcomes in [38, 44].

2) Discovering a hold action: After the previous high-level action discovery, the agent is presented with the red cube. Figure 9 depicts the excitation level of each action for three time-steps, during which the agent discovers the hold action. As shown in this figure, a generalised reach schema is selected at first because it offers the highest level of excitement, followed by a generalised grasp. Although a seize action was previously learnt, the agent does not choose it due to an insufficient number of confirmations. At this point, the agent does not possess any schema to reach a different position in the space. To do that, the agent's focus needs to change a certain new source of stimuli in the space. When a green cube is placed on the table by the human observer, the necessary change of focus occurs, and thus, the agent is found to select a generalised reach schema towards the green cube's position at the third step. The fact that the agent is holding the red cube, causing the grasp-related perception miss-match to the post-condition, is reflected by the reduction in excitation noticed of the generalised reach schema, as compared to the similar selection during the first execution step.

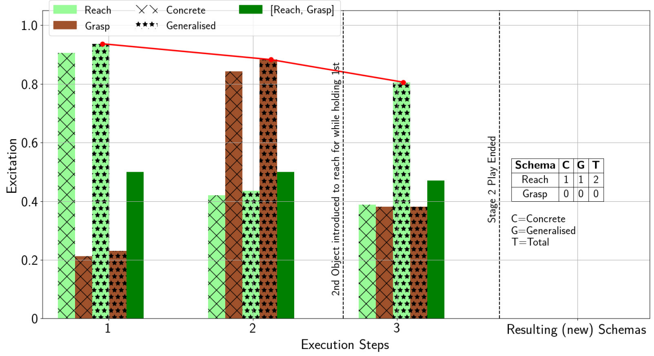


Fig. 9: Play behaviour to develop a generalised "reach $", while holding an object, with excitation of all schemas and chains shown.

Subsequently, the post-condition for the generalised reach action includes information of the holding hand and the two cube objects in the same location. As a result, a new concrete schema that reflects the experience of reaching while holding an object is added into the memory, along with a newly generalised reach schema capable of reaching using any cube-like object towards any other position.

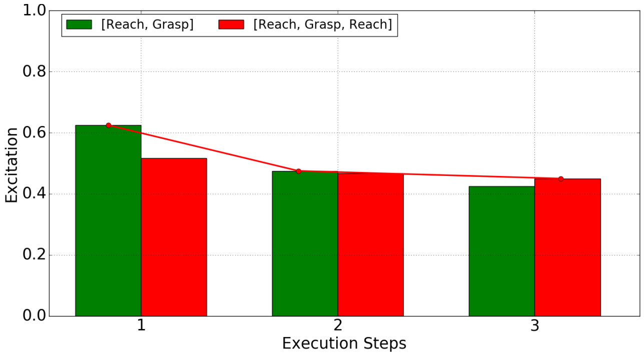


Fig. 10: Suggested chains and their excitation levels to transport novel object to a different position.

To test the new schemas a novel user-defined problem is set. The desired post-condition after presenting the agent with only the green cube becomes "both the hand and the cube at a novel position, with closed grip and touch sensation received". The resulting chains and the sequence of actions selected by the agent to solve the problem are shown in Figure 10. It is observed that it takes the agent two failed attempts to recognise and utilise the new hold action chain. As the positions of both hand and object are generalised, the previously learnt seize chain and hold action chain have the potential to achieve their corresponding desired post-condition. After two failed attempts, the agent reshapes its learning and ultimately selects hold chain over the seize to hold the object at the desired position. The sequence of failed attempts is a demonstration of the refinements within the schema memory that enable the agent to reshape its learning.

3) Discovering move-by-hold action: Having learnt how to hold an object and move it in space, i.e., the object is grasped and follows the position of the hand, and the agent does not release the grasped object. This novel high-level action is referred to as an object relocation action. It emerges as a result of merging the previously learnt generalised reach and grasp schemas, with a generalised release schema (which reflects the release action at any position and any object). The playing that led to the generation of the generalised release schema is shown in Figure 11.

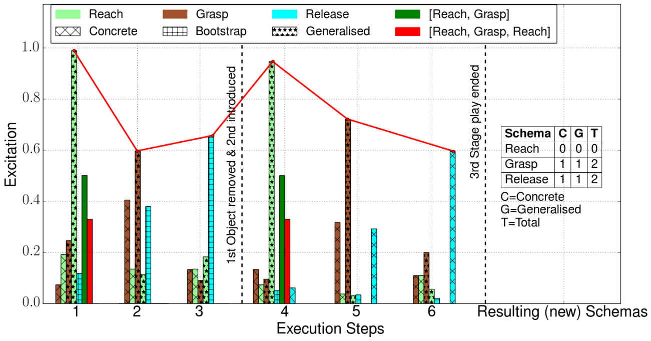


Fig. 11: Play behaviour to develop a generalised "release $" action.

After reach and grasp are performed on an object, i.e., a red cube depicted in the figure as the outcome of the second execution step, the following excited schema to perform in the third step is the bootstrap release, causing the agent to drop the object. At this point, a concrete release schema reflecting the phenomenon of dropping the red cube is created. Subsequently, a human observer removes the red cube and introduces a green cube on the table to trigger the agent's attention mechanism towards a new position. Again, the agent utilises the generalised reach and grasp schemas to obtain the second object into the hand. Thanks to the partial matching mechanism explained previously in Section IV, the most excited schema to perform now is the concrete release that was generated before (at the end of execution step three). Although a cube of a different colour, the execution of the concrete release schema allows the green object to drop and enables the creation of a generalised release schema, which is added to the system's memory after execution step 6.

At the end of the sixth execution step, the agent has generalised reach, grasp and release schemas. The human observer introduces a novel object at a random position on the table. The user-defined problem now becomes "object at a new position, with open grip and no-touch sensation received", implying that the object is to be relocated from its initial position to the new user-defined one. Figure 12 depicts the learning attempts of the system to provide a solution to the problem.

The first observation is that given all knowledge stored in the schemas, the agent attempts to use three alternative chains to achieve the resulting post-condition. The highest excitation

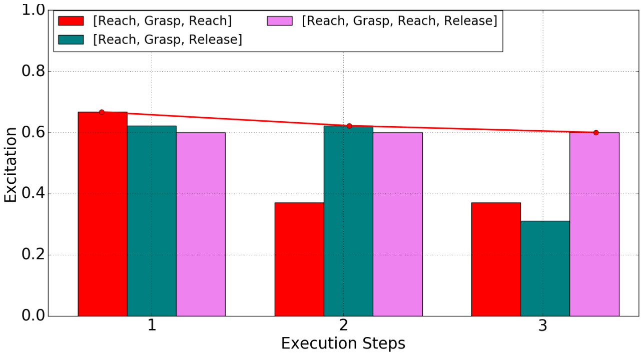
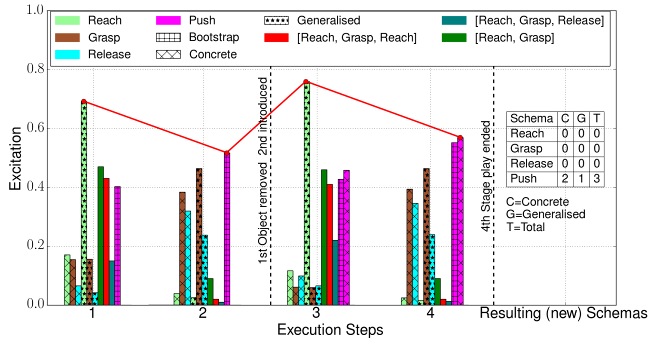


Fig. 12: Suggested chains and excitation levels to transport the novel object to a different position.

is given by the hold "reach-grasp-reach" chain, discovered previously, given the highest number of matched properties in terms of both object characteristics and hand configurations. Aligned with the idea that playing is a mechanism to refine learning, after applying the hold chain, the system discovers that although the position of the object has changed, the hand's grip is not appropriate. Therefore, the hold chain is penalised making the chain "reach-grasp-release" the most excited one. Like the hold chain, this chain also matches the post-condition, i.e., the object is moved to another generalised position and is released from the hand. These two characteristics make the chain suitable. Note that the agent utilises all past experiences to generate appropriate solutions for user-defined problems. During the learning stage, the agent experimented with employing a release action, the results of which have led to the object being dropped from the robotic hand. Unlike in simulations, dropping an object in the real-world may cause the object to roll on the table slightly and thus have a somewhat different position from the hand. This is why a "reach-grasp-release" chain is suggested as the next most exciting chain to perform. As expected, the object's position does not match the desired rendering a third attempt necessary. The third execution step leads the "reach-grasp-reach-release" chain being suggested, with an excitation level over 0.6 where others are penalised. This attempt, representing the agent's high-level action to relocate objects by holding them, allows the robot to achieve the user-defined post-conditions.

4) Discovering a move-by-push action: A sequence of reach and push actions is hereafter referred to as *move-by-push* to differentiate from a move-by-hold where the robot is relocating an object by holding it in its hand.

To learn how to move objects by pushing them, the iCub is provided with a bootstrap push schema and is presented with a red cube on the table. Like before, the agent is given time to explore actions through free playing. Similarly to grasp and release actions, any push schema (i.e., bootstrap, concrete or generalised) does not have any coordinates in its action. As explained in Section IV, the excitation of such schemas is penalised when the hand does not share the position with an object, as a push action would not create any change in the environment without any object at the hand position. Figure 13 summarises the execution steps while playing freely towards



Schema	C	G	T
Reach	0	0	0
Grasp	0	0	0
Release	0	0	0
Push	2	1	3

Fig. 13: Play behaviour to develop a generalised "push $" action.

the discovery of a high-level move-by-push action. colorredAt first a generalised reach is found to be the most excited schema to be selected. As excitation parameters, discussed in Section V, are tuned to favour actions over objects, the bootstrap push action is selected. As a result, the iCub discovers the relocation of the object in the scene using a push action. This new knowledge is stored in the memory, and the human observer replaces the red cube with a green one. The agent interacts with the new object following a similar sequence of actions, reach action followed by a push, and generates a concrete push schema for the green cube and a generalised push schema.

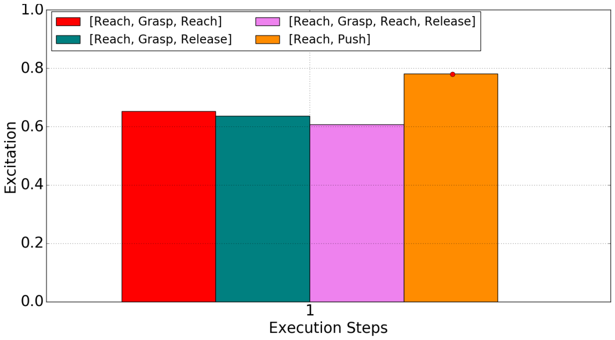


Fig. 14: Suggested chains and their excitation levels to displace novel object to a different position.

After having learnt a push action, the user-defined problem is set to be "object at a different position" with the iCub placed in front of a novel object. This problem's post-condition does not include any information about the robotic hand. The suggested solutions for fulfilling the problem requirements are depicted in Figure 14. It is observed that four chains are recommended, with "reach-push" having the highest excitation level due to its shorter length and novelty. More interestingly, it is found that the agent suggested hold ("reach-grasp-reach") and move-by-hold ("reach-grasp-reach-release"), the two previously learnt high-level actions. As there is no description of the hand configuration in the problem's post-condition, they are both valid and can offer the desired outcome.

The results show that the agent, provided with the opportunities, is able to discover higher level skills through free-play, increasing in complexity at each step of the experiment. The agent is also able to utilise its developed skills to achieve a goal state, instructed by an user, through utilising the chaining mechanism in the system.

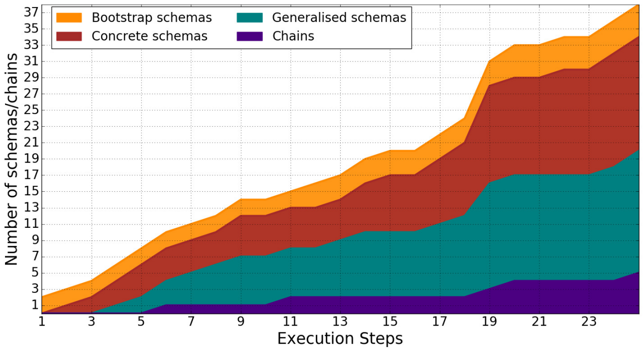


Fig. 15: Number of schemas/chains in memory.

During free play where the agent was presented with objects, 29 new schemas, including 14 concrete and 15 generalised, were created. In addition, five new chains were added in memory, including those representing the high-level actions described above, consisting of generalised schemas that render them suitable solutions to manipulate different objects. As a summary Figure 15 shows the total number of bootstrap, concrete and generalised schemas, and chains in the memory throughout this first experiment.

B. Understanding towards tool-use and discovery of object affordance

In the previous set of experimental results, learning has led to the discovery of several high-level actions, manifested as chains in the Dev-PSchema memory. Such chains consist of learnt schemas that provide solutions to either a problem with particular characteristics, i.e., concrete schemas, or to classes of problems, i.e., generalised schemas.

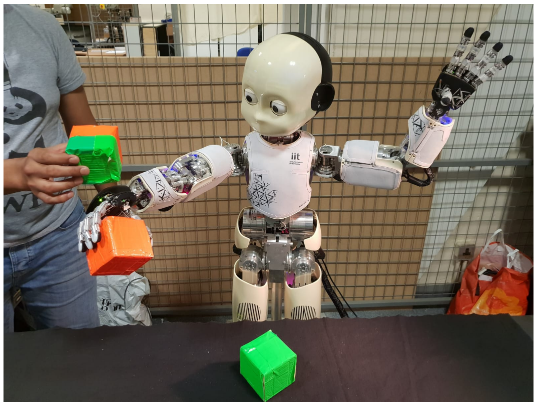


Fig. 16: Final step of executed chain is to lift the trigger object (red cube) to the trigger position in space for the observer to bring the toy object in the scene.

With a human observer setting a user-defined problem by using high-level perceptual representations as discussed in Section IV, the second experiment demonstrates the ability of the agent to apply those high-level actions to a novel problem. We examine the ability of the agent to associate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3094513, IEEE Transactions on Cognitive and Developmental Systems

previously learnt high-level actions with certain phenomena that require the involvement of familiar objects to occur. Through playing, the discovery of the associations of objects to achieving the desired solution for a given problem is an essential step towards the understanding of their affordances. In turn, it is directly related to tool-use, where the agent develops an understanding about the effect on a object caused by a direct interaction with another object. To help the robot draw associations, the human observer's interference is used. Two objects, including a trigger object, are placed on the table one by one, for the robot to manipulate. It is only when the robot moves the trigger object to a specific point in space, i.e., lifting the object as seen in Figure 16, that the human observer reveals a toy object (a bicoloured cube), offering extra visual stimuli to the robot. Revealing the toy object is only repeated when the trigger object reaches the same position, referred to as the trigger position. Upon the completion of the learning phase, the robot is presented with the trigger and non-trigger objects of the experiment and is asked to make the toy object appear.

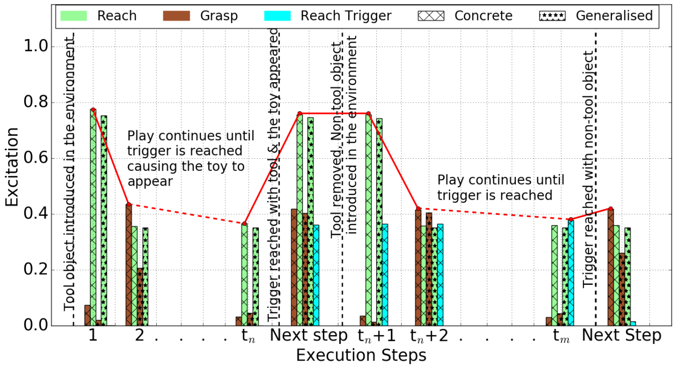


Fig. 17: Play behaviour to learn association between trigger and trigger position.

1) Learning process: To demonstrate this process, the agent is set to start by knowing how to seize, hold and transport objects within its proximity. With a red cube (hereinafter referred to as the trigger object) being the only object placed on the table and the robot being driven by its play algorithm, reach and grasp actions are performed repeatedly. The sequence of actions taken by the robot during play is depicted in Figure 17. After observing certain random hold actions, the human observer decides to consider the trigger position from one of the previous reach positions and reveals the toy object once the position is reached at execution step t_n. At this point, the schema generation mechanism produces a new schema that links concrete information about the physical appearance of the toy and the trigger objects, along with their positions in the environment.

At this milestone step, the scene is reset, and a non-trigger object is placed on the table. At the next execution steps, t_n+1 and t_n+2 respectively, the robot manipulates the new object by reaching and grasping. After several execution steps, a hold action brings the non-trigger object to the trigger position. Unlike before, the toy object does not appear, preventing the association between the toy object and a generalised trigger-object along with the sequence of actions to be made.

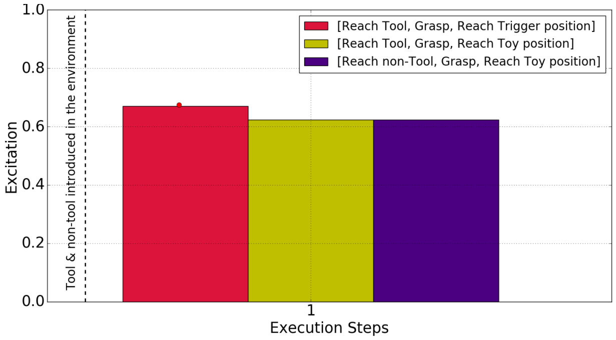


Fig. 18: Suggested chains with user defined goal to bring the toy object in the scene.

2) Problem solving: In this phase, the robot is placed in front of the table and is presented with both the trigger and non-trigger objects. The user-defined problem is now set to "toy object at trigger position". Importantly, the desired post-condition does not contain descriptions of any of the objects, nor the positions that need to be considered to reveal the toy object. The robot is expected to find previous experiences and to consider them in order to suggest sequences of actions that may maximise the potential of producing the target world state.

Figure 18 shows the resulting schema chains and their excitation levels. The agent suggests three alternatives, with the one that reaches and grasps the trigger object and moving it to the trigger position being the one with the highest excitation. This solution is the only true positive amongst the suggested other two only offer a partial match to the desired outcome. This expected behaviour results from the over-generalisation that takes place while the agent successfully creates several concrete and generalised schemas to learn the association between the trigger and the toy objects.

VII. Conclusions

In this paper, we have presented a novel approach for the generation and use of schema chains for an agent to achieve goals that require complex object manipulation actions. We take a holistic approach where, unlike other related studies such as that of Manoury et al. [14], the learning algorithm receives input from a low-level subsystem capable of extracting visual and tactile information to build object representations. Such representations are then utilised to autonomously develop high-level actions through play. These actions result from performing combinations of previously learnt primitives such as reach and grasp.

Our experimental methodology is designed to exhibit the autonomous learning of a humanoid robot in two stages, where a human observer (the user) systematically gives visual stimulation to encourage the robot's unhindered exploratory behaviour. In the first stage, the play generator employed drives the robot's attention and decision making towards the discovery of generalised schemas that make the manipulation of novel objects possible. Through playing, those schemas are combined by the schema chain mechanism in attempts to associate objects with sequences of actions. In the second stage, newly acquired knowledge is put to the test by a human observer. With the latter now setting the desired goal, we demonstrate that the agent generates solutions beyond what was previously experienced in novel contexts, with a target perceptual state being described as a post-condition for the agent to produce. Being an open-ended process, the learning continues to occur while the robot suggests, tries and ultimately evaluates the outcome of alternative solutions. Given sufficient time to interact with familiar and novel objects and to learn by experience, the robot gradually develops its understanding of the scene.

Through a representative experimental run, we document the process of learning high-level actions by the system implemented with an iCub. We investigate the potential of this system to discover affordances and utilise objects to achieve target goals. Moreover, we demonstrate the new architectures potential to learn and combine sequences of interdependent skills to generate solutions for a user-defined, desired state without describing any intermediate steps explicitly or giving detailed information about how to achieve it. The description of the latter is decoupled from the key elements that may contribute to the solution. In experiment II, for instance, the target state is simply to make the toy object appear; no information about the hand and tool object, actions, nor positions are necessary. In particular, we presented a step-by-step schema chain generation, whereby the iCub autonomously learns the effect of actions to specific objects (concrete schemas) and objects with similar characteristics (generalised schemas). Developmental psychology studies document basic problem-solving capability in early infancy. McCarty et al. [9] introduced different spoon orientations that led to infants of varying age making mistakes. These included the recruitment of previously successful actions, and the application of corrections, i.e., planning sequences to achieve the desired target states. Equipped with Dev-Pschema, the iCub is able to evolve and develop via a process of making mistakes and using the outcome of corrected actions to shape its future decisions.

We have also highlighted the potential of the implemented system to discover object affordance through playing, showing the generation of sequences to produce an effect on an object (i.e., making the toy object appear) through direct manipulation of another (i.e., moving an object towards a specific position). This demonstrates an ability of the system to develop the understanding towards tool-use related problems. The iCub learns how to relate a trigger object with the appearance of a toy object in space, treating the learning of such associations as an ongoing process, during which the robot refines its schema memory by considering the effect of both successful and unsuccessful actions to the scene. Similar behaviour has been observed in young children as reported in developmental psychology, where the children were able to develop an association of an action on one object causing a particular effect on another object [7].

In solving a novel tool-use related problem, the iCub is able to suggest several alternatives through which the robot can manipulate the environment towards the desired result. Note that making mistakes is an integral part of human learning process; learning from incorrect use of non-tool objects as tools has been reported in numerous studies of infants in developmental psychology Goubet et al. [45]. This developmental learning behaviour has been reflected by the iCub implementing the present approach.

The presented architecture addresses the following developmental learning aspects desired for any psychologically plausible system [46]. It incorporates sensorimotor contingencies' autonomous discovery and maintains a memory that allows their reusability. The success rate of unsuccessful and non-applicable sensorimotor associations is reduced over time, making them less likely to be selected. The architecture generalises sensory-motor contingencies to utilise them in novel situations and, combined with the chaining mechanism, it can achieve goal-directedness.

Designing systems that allow an agent to learn objects and discover their use, i.e., the association of objects, actions and effects, is an essential concept in autonomous robotics. However, Dev-PSchema may develop over-generalised schemas. In our future work, we aim to address over-generalisation by amending the generalised schemas in memory using deductive reasoning; a method that helps to change the level of granularity from very abstract to specific based on captured experiences [38]. The extended generalisation algorithm would improve the generalised schemas by de-generalising those with poor performance, e.g., giving concrete values to their problematic properties to potentially achieve accurate matches in future. The generalisation mechanism can be further extended to learn ranges of values and limitations of specific actions. A generalised reach schema may contain generalised coordinates, which theoretically allows the agent to reach anywhere within the visual space. Being a stationary agent, that is currently not possible. Thus, extending Dev-PSchema such that it can identify possible extreme values for each generalised property through learning, forms another piece of interesting future work. Furthermore, the work can be improved by introducing a house-keeping mechanism where generalised schemas, which are less likely to produce successful results, can be removed from the memory.

As mentioned in Section V, tuning the weights (i.e., ω_1 to ω_4) allows us to simulate different infant behaviours, leaning towards learning either novel objects or actions. In the future, we want to investigate potential strategies to tune these parameters online. We anticipate that by monitoring the learning rate (e.g., the number of significantly novel generated schemas over the number of successfully performed actions), we can potentially regulate the weights dynamically and, as such, affect the learning bias accordingly.

References

[1] J. Piaget, M. Cook, and W. Norton, *The origins of intelligence in children*. Intl. Universities Press New York, 1952, vol. 8, no. 5.

[2] E. J. Gibson, "Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge," *Annual review of psychology*, vol. 39, no. 1, pp. 1–42, 1988.

[3] D. A. Baldwin, "Infants' ability to consult the speaker for clues to word reference," *Journal of child language*, vol. 20, no. 2, pp. 395–418, 1993.

[4] S. A. Graham and D. Poulin-Dubois, "Infants' reliance on shape to generalize novel labels to animate and inanimate objects," *Journal of Child Language*, vol. 26, no. 2, pp. 295–320, 1999.

[5] E. A. Zack, "Infant transfer of learning across two-dimensional/three-dimensional dimensions: A touch screen paradigm," Ph.D. dissertation, Georgetown University, 2010.

[6] Z. L. Sim and F. Xu, "Learning higher-order generalizations through free play: Evidence from 2-and 3-year-old children." *Developmental psychology*, vol. 53, no. 4, p. 642, 2017.

[7] H. Gweon and L. Schulz, "From exploration to instruction: Children learn from exploration and tailor their demonstrations to observers goals and competence," *Child development*, vol. 90, no. 1, pp. e148–e164, 2019.

[8] M. Weigelt and T. Schack, "The development of end-state comfort planning in preschool children," *Experimental Psychology*, pp. 476–482, 2010.

[9] M. E. McCarty, R. K. Clifton, and R. R. Collard, "Problem solving in infancy: the emergence of an action plan." *Developmental psychology*, vol. 35, no. 4, p. 1091, 1999.

[10] B. Flunkert, "The role of the contingent negative variation in chunking. evidence from a go/nogo discrete sequence production task." 2009, B.S. thesis, University of Twente.

[11] S. Kumar, P. Shaw, D. Lewkowicz, A. Giagkos, Q. Shen, and M. Lee, "Developing object understanding through schema generalisation," in *Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2016 Joint IEEE International Conference on*. IEEE, 2016, pp. 33–38.

[12] S. Kumar, P. Shaw, D. Lewkowicz, A. Giagkos, M. Lee, and Q. Shen, "Generalising predictable object movements through experience using schemas," in *International Conference on Simulation of Adaptive Behavior*. Springer, 2016, pp. 329–339.

[13] E. Aksoy, M. Tamosiunaite, R. Vuga, A. Ude, C. Geib, M. Steedman, and F. Worgotter, "Structural bootstrapping at the sensorimotor level for the fast acquisition of action knowledge for cognitive robots," in *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*. IEEE, 2013, pp. 1–8.

[14] A. Manoury, S. M. Nguyen, and C. Buche, "Hierarchical affordance discovery using intrinsic motivation," in *Proceedings of the 7th International Conference on Human-Agent Interaction*, 2019, pp. 186–193.

[15] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Grail: a goal-discovering robotic architecture for intrinsically-motivated learning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 3, pp. 214–231, 2016.

[16] S. Forestier and P.-Y. Oudeyer, "Curiosity-driven development of tool use precursors: a computational model," in *8th Annual Conference of the Cognitive Science Society (CogSci 2016)), Aug2016, Philadelphie, PA, United States*, 2016, pp. 1859–1864.

[17] S. Forestier and P. Y. Oudeyer, "Modular active curiosity-driven discovery of tool use," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3965–3972.

[18] R. Cubek and W. Ertel, "Learning and application of high-level concepts with conceptual spaces and pddl," in *PAL 2011 3rd Workshop on Planning and Learning*, 2011, pp. 76–83.

[19] A. Antunes, L. Jamone, G. Saponaro, A. Bernardino, and R. Ventura, "From human instructions to robot actions: Formulation of goals, affordances and probabilistic planning," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5449–5454.

[20] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically Motivated Reinforcement Learning," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2004, p. 12811288.

[21] V. G. Santucci, G. Baldassarre, and E. Cartoni, "Autonomous reinforcement learning of multiple interrelated tasks," in *2019 Joint IEEE 9th international conference on development and learning and epigenetic robotics (ICDL-EpiRob)*. IEEE, 2019, pp. 221–227.

[22] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[23] G. L. Drescher, *Made-up minds: a constructivist approach to artificial intelligence*. MIT press, 1991.

[24] M. Sheldon, "Intrinsically motivated developmental learning of communication in robotic agents," Ph.D. dissertation, Aberystwyth University, 2013.

[25] S. Kumar, P. Shaw, A. Giagkos, R. Braud, M. H. Lee, and Q. Shen, "Developing hierarchical schemas and building schema chains through practice play behaviour," *Frontiers in neurorobotics*, vol. 12, p. 33, 2018.

[26] S. Kumar, "Learning with play behaviour in artificial agents," Ph.D. dissertation, Aberystwyth University, 2019.

[27] R. Casati, "Object perception," *Oxford handbook of philosophy of perception*, pp. 393–404, 2015.

[28] A. Giagkos, D. Lewkowicz, P. Shaw, S. Kumar, M. Lee, and Q. Shen, "Perception of localized features during robotic sensorimotor development," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 2, pp. 127–140, 2017.

[29] D. Lewkowicz, A. Giagkos, P. Shaw, S. Kumar, M. Lee, and Q. Shen, "Towards learning strategies and exploration patterns for feature perception," in *Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2016 Joint IEEE International Conference on*. IEEE, 2016, pp. 278–283.

[30] J. Fiser and R. N. Aslin, "Statistical learning of new visual feature combinations by infants," *Proceedings of the National Academy of Sciences of the United States*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3094513, IEEE Transactions on Cognitive and Developmental Systems

of America, vol. 99, no. 24, pp. 15 822–15 826, 2002.

[31] B. L. White, P. Castle, and R. Held, "Observations on the development of visually-directed reaching," *Child development*, pp. 349–364, 1964.

[32] K. Earland, M. Lee, P. Shaw, and J. Law, "Overlapping structures in sensory-motor mappings," *PloS one*, vol. 9, no. 1, p. e84240, 2014.

[33] R. Braud, A. Giagkos, P. Shaw, M. Lee, and Q. Shen, "Robot multi-modal object perception and recognition: Synthetic maturation of sensorimotor learning in embodied systems," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[34] A. Giagkos, Raphaël, P. Shaw, M. Lee, and Q. Shen, "Assessing Humanoid Multimodal Grasping Towards Object Recognition," in *2nd Robot Manipulation Workshop*, Imperial University, London, July 2017.

[35] C. von Hofsten, "Predictive reaching for moving objects by human infants," *Journal of experimental child psychology*, vol. 30, no. 3, pp. 369–382, 1980.

[36] R. L. Canfield and M. M. Haith, "Young infants' visual expectations for symmetric and asymmetric stimulus sequences." *Developmental Psychology*, vol. 27, no. 2, p. 198, 1991.

[37] M. Paulus, S. Hunnius, C. van Wijngaarden, S. Vrins, I. van Rooij, and H. Bekkering, "The role of frequency information and teleological reasoning in infants' and adults' action prediction." *Developmental psychology*, vol. 47, no. 4, p. 976, 2011.

[38] A. N. Welder and S. A. Graham, "The influence of shape similarity and shared labels on infants inductive inferences about nonobvious object properties," *Child development*, vol. 72, no. 6, pp. 1653–1673, 2001.

[39] T. Wilcox, "Object individuation: Infants use of shape, size, pattern, and color," *Cognition*, vol. 72, no. 2, pp. 125–166, 1999.

[40] S. A. Jax and D. A. Rosenbaum, "Hand path priming in manual obstacle avoidance: evidence that the dorsal stream does not only control visually guided actions in real time." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 33, no. 2, pp. 425–441, 2007.

[41] S. W. Kent, A. D. Wilson, M. S. Plumb, J. H. Williams, and M. Mon-Williams, "Immediate movement history influences reach-to-grasp action selection in children and adults," *Journal of motor behavior*, vol. 41, no. 1, pp. 10–15, 2009.

[42] P. Dixon, S. McAnsh, and L. Read, "Repetition effects in grasping." *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 66, no. 1, pp. 1–17, 2012.

[43] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 50–56.

[44] J. M. Mandler and L. McDonough, "Studies in inductive inference in infancy," *Cognitive Psychology*, vol. 37, no. 1, pp. 60–96, 1998.

[45] N. Goubet, P. Rochat, C. Maire-Leblond, and S. Poss, "Learning from others in 9-18-month-old infants," *Infant and Child Development: An International Journal of Research and Practice*, vol. 15, no. 2, pp. 161–177, 2006.

[46] L. Jacquey, G. Baldassarre, V. G. Santucci, and J. K. ORegan, "Sensorimotor contingencies as a key drive of development: from babies to robots," *Frontiers in neurorobotics*, vol. 13, p. 98, 2019.

Suresh Kumar received the degrees of B.E. in Electronics Engineering from Mehran UET Pakistan, M.S. in Control Engineering from GCU Lahore Pakistan and the Ph.D. degree in Intelligent Robotics from Aberystwyth University, UK. He is an Assistant Professor and Head of Robotics Cluster in CRAIB at Sukkur IBA University, Pakistan. His regions of research interest are sensorimotor learning, modelling play behaviour, developmental and cognitive robotics.

Alexandros Giagkos received the B.Sc. degree in computer science, the M.Sc. degree in Internet and distributed systems, and the Ph.D. degree in computer science from Aberystwyth University, Aberystwyth, U.K. He is a Lecturer at Aston University, Birmingham, U.K. His research interests include developmental, evolutionary, and swarm robotics.

Patricia Shaw received her B.Sc. in Artificial Intelligence (2005) and Ph.D. in Computer Science (2010) from the University of Durham. She is a Lecturer in the Intelligent Robotics Group at Aberystwyth University. Her research interests include biologically and psychologically inspired architectures for developmental learning in robotic systems.

Raphaël Braud received the M.Sc. degree in intelligent systems and robotics and the Ph.D. degree in developmental robotics on the modeling of cognitive mechanisms for sensorimotor control and tooluse from the University of Cergy-Pontoise, Cergy, France, in 2012 and 2017, respectively. He is a Research Engineer in machine learning with the French Institute of Technologie SystemX, Palaiseau, France.

Mark Lee received the degrees of B.Sc. (1967) and M.Sc. (1969) in Electrical Engineering from the University of Wales, Swansea, and the Ph.D. (1980) in Psychology from Nottingham University. He is an Emeritus Professor of Intelligent Systems in the Department of Computer Science at Aberystwyth University, and a Fellow of the Learned Society of Wales (the national academy of Wales). Prof. Lee was PI on four recent EPSRC and EC funded research projects on robotic sensory-motor learning, adaption and development.

Qiang Shen received the Ph.D. degree from Heriot-Watt University, Edinburgh, U.K. in 1990, and the D.Sc. degree from Aberystwyth University, Aberystwyth, U.K. in 2013. He holds the Established Chair in Computer Science and is Pro Vice-Chancellor for Business and Physical Sciences, Aberystwyth University. He has authored two research monographs and more than 400 peer-reviewed papers, including one receiving an Outstanding Transactions Paper Award from IEEE.