Formalizing Problem Solving in Computational Thinking : an Ontology approach

Chloé Mercier, Lisa Roux, Margarida Romero, Frédéric Alexandre, Thierry Viéville

Formalizing Problem Solving in Computational Thinking : an Ontology approach

Chloé Mercier
Mnemosyne
Inria
Bordeaux, France
chloe.mercier@inria.fr

Lisa Roux
Laboratoire d'Innovation
et Numérique pour l'Education
Université Côte d'Azur
Nice, France
lisaroux.87@gmail.com

Margarida Romero
Laboratoire d'Innovation
et Numérique pour l'Education
Université Côte d'Azur
Nice, France
margarida.romero@unice.fr

Frédéric Alexandre
Mnemosyne
Inria
Bordeaux, France
frederic.alexandre@inria.fr

Thierry Viéville
Mnemosyne
Inria
Bordeaux, France
thierry.vieville@inria.fr

***Abstract*—We introduce the idea of a symbolic description of a complex human learning task, in order to contribute to better understand how we learn. The learner is modeled on the basis of knowledge from learning sciences with the contribution of cognitive neurosciences, including machine learning formalism, in the very precise framework of a task, named #CreaCube reviewed here, related to initiation to computational thinking presented as an open-ended problem, which involves solving a problem and appealing to creativity.**

We target problem-solving tasks using tangible interfaces for computational thinking initiation, and describe in details how we model the task and the learner behavior in this task, including goal-driven versus stimulus-driven behavior and the learner knowledge construction. We show how formalizing these elements using an ontology offers a well-defined computational model and the possibility of inferences about model elements, analyzing and predicting the learner behavior.

This operationalization of a creative problem-solving task is still at a preliminary stage, but an effective proof of concept is described in this study.

***Index Terms*—Cognitive Neuroscience, Ontology, Problem-Solving, Learning Sciences, Computational Thinking**

I. Introduction

Understanding how we learn is a key issue for improving education worldwide. This is especially true for transversal competencies sometimes referred to as "21st-century skills", such as computational thinking ([25], see also [15] for a recent review). Other transversal skills such as creativity, problem solving and critical thinking are especially important in the present period for the development of citizenship [22]. Such competencies are linked and must be considered in interdependence. For instance, computational thinking is intrinsically related to problem solving (i.e., requires to solve ill-defined tasks, not only solving well-defined problems using a step-by-step approach) and involves creativity in practice (e.g., creative programming), beyond specific know-what (knowledge) and know-how skills. Computational thinking requires techno-creative activities to be developed and must integrate critical thinking development, especially when acculturating citizens of all ages to Artificial Intelligence [4]. How to better understand such human learning ? Our contribution is a modeling proposal of such a problem-solving learning activity as a step towards better understanding these transversal competencies, thus tackling a key challenge in learning science (also referred to as educational science, from the French terminology).

Problem solving: Problem solving in everyday life takes different forms depending on the task and context. Despite the pioneering work of [19] aiming to develop a general problem-solving framework, human problem solving is dependent on the task and context, requiring to consider prior knowledge but also the knowledge construction that is developed through the task process based on the different operations the subject performs on the task. Research on problem solving has focused on a diversity of tasks, most of all on well-defined problems such as the chess game or the Tower of Hanoi, allowing to define task models and problem-solving methods. In short, the idea is to represent the problem space, including the positions of the initial state and of the desired goal (possibly a region of the space if the goal is not defined as a unique position but specified through instructions) and to define a path between the positions. This can be done in both forward and backward directions, where knowledge can provide rules for moving in the space or constraints to obey.

Using tangible artifacts: Taking into account the need to consider the set of specific knowledge required to solve a task, we focus our study on a specific problem-solving task with tangible material. Using tangible artifacts allows not only to infer cognitive process but also to observe behavioral patterns [21]. At a more concrete level, we aim to study creative problem solving through screen-less activities, also known as unplugged activities [12], for instance, using pedagogical

robots or connected objects [21]. These activities require a non-trivial setup in order to track the learning process through the activity. They aim to consider the problem-solving process instead of considering only performance scores of the task. Methodological challenges in process-based data collection also take into account the ethical issues of collecting human data in a way that ensures data privacy. One challenge is the use of efficient measurement devices of reasonable cost and it appears that a precise model of the task allows to define specific observables that are thus more robust to estimate [21]. Another key point is the bias induced by the fact that the learner is observed and monitored, and one solution is to take this as a chance to involve the learner in their own learning process, but this is to be properly studied, see [6] for a discussion on these points. In both cases, the construction of a precise model of the task and of the learner involved in this task is a lever to take up this challenge.

The data challenge: A step further, the challenge here is to work from a relatively small batch of data (a few dozen to compare with the thousands of data used with classic statistical methods). In order for the results to be meaningful, we need to introduce a maximum of prior information upstream to the analysis of the observables, resulting in highly structured data. An ontology is an appropriate tool when it comes to describing and structuring data. We are thus making use of ontology modeling to model the learner and the task, including the observables collected during such a learning activity. This model will then be applicable to the observed learning analytics and allow to interpret them. Phenomenological models, indeed, already exist: the ICAP framework [7] proposes to categorize cognitive engagement behaviors into four modes (Interactive, Constructive, Active,and Passive) and links these behaviors to learning outcomes, while the activity theory [9] is an appropriate framework providing the theoretical context of this study. Based on these epistemological frameworks, our present study is positioned at an operational level.

II. Material and method

A. The #CreaCube task

In the #CreaCube study[1], we aim to analyze problem-solving strategies using modular robotic cubelets, targeting children between 8 and 12 years old. The task engages the participants into building an autonomous vehicle, composed of 4 items, able to move from a point A to another point B. In this context, the participant is given a set of 4 modular robotic cubelets which differ in their appearances (e.g. different colors) and affordances (e.g. wheels, switch...). The learner is expected to understand that the 4 "items" mentioned by the guideline are the 4 cubelets on the table, and that they should assemble those cubelets into a certain configuration which will be able to move autonomously. It is worth noting that this problem does not have a unique solution as several configurations may satisfy the goal. The task is relatively easy in the sense that most participants come up with a solution in less than 15 minutes, but complex to model because of the large problem space and the lack of specification (e.g. the possible actions are not clearly stated).

The #CreaCube task engages the participant in an ill-defined problem, for which the participant has no prior experience of the modular robotic cubelets. Considering this lack of knowledge, there is a need in exploratory orientation, which is defined from a learning science perspective, as a way to gather information from the environment [13] in order to complete the task. Throughout the task, the participant moves towards the goal by mobilizing prior knowledge which appears to make sense in relation to their experience with the material being manipulated and the requirements defined by the guideline. Specifically, in this particular task, exploration aims to understand the artifacts, their characteristics and technological affordances, allowing to formulate hypotheses on the material and define sub-goals to solve the problem. The exploration of the means given to solve the task should facilitate the generation of new stimuli which can then help to make progress towards the goal, by fulfilling sub-goals and validating or invalidating hypotheses.

The task is modeled based on the knowledge needed to solve it but also on the initial states of the hardware and the final state for its success, as detailed in Fig. 1. We can see the different observables taken into account, i.e. the possible configurations of the cubelets (disassembled or assembled into a certain shape), the discovery of affordances, i.e. the practical possibilities offered by an element (e.g. there is a switch, so it can be flipped), the different results obtained (e.g. at the motion level), but also elements linked to the subject, such as their emotions or their attitude (perseverance, abandonment) regarding the task. All of these observables will serve as a basis for modeling the task in the form of structured knowledge.

The file model generated from this interface has been developed in a typed hierarchical format (JSON syntax, with both the raw and computed data as well as the description of each type of information and its relationships to other types) to allow its formal manipulation and to best represent the structure of the collected information, in order to make the link with knowledge representations related to the computational notion of ontology (as used in the Semantic Web) and integrated as an interactive platform to facilitate the ergonomics of manual video analysis [23]. Each video for the #CreaCube task is encoded as a temporal sequence of states corresponding either to a configuration of the task or to a state of the learner in the task as explained above (Fig. 1). Here is an example of such a JSON file generated from the interface, describing for a given participant at which moment he reached which observable:


```
{"clicks": [
    {"time": "46","click": "AS01"},
    {"time": "50","click": "AS02"},
    {"time": "58","click": "AS01"},
    {"time": "67","click": "AS00"},
    {"time": "82","click": "AS01"},
    {"time": "93","click": "AS00"},
```


[1]See https://creamaker.wordpress.com/2019/02/06/publications-within-the-creamaker-project for full details

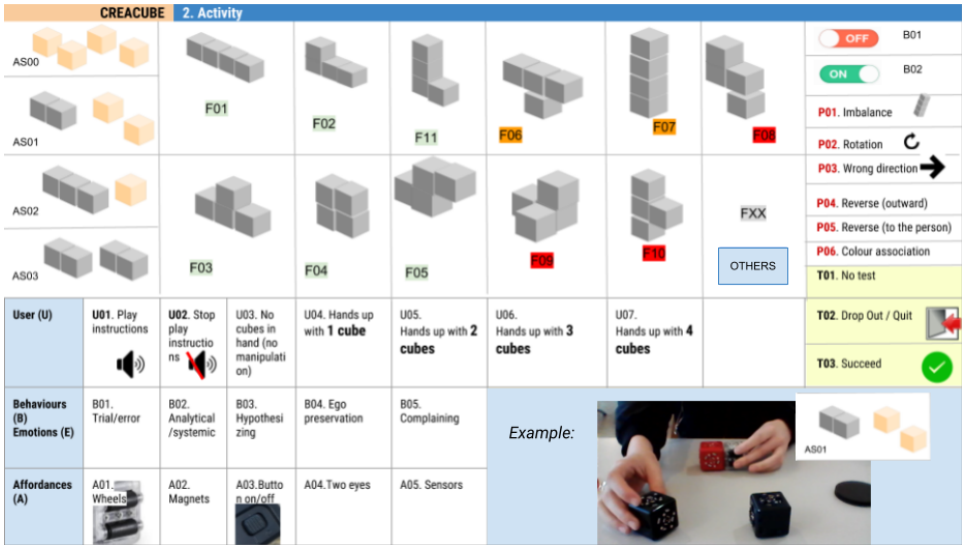

Fig. 1: Interface for the identification of observables, this figure describes a part of the system states to which is added, for example, the identification of each cubelet (recognizable by its color: navy blue battery, black sensor, white motor, red inverter) and the states of the cubelets (e.g. "connected/disconnected" or "on/not on wheels"). This sub-ensemble of the possible states corresponds to the observables that have been chosen to analyse the task (see http://aide-line.inria.fr/public/doc/vid.mp4 for a video of the experiment.)


```
    {"time": "101","click": "AS01"},
    {"time": "109","click": "AS02"}
],"idParticipant": "p362"}
```


The methodology for generating an organized set of learning analytic log from video analysis has been developed from a learning sciences perspective through the ANR CreaMaker project to study problem-solving activities. These videos serve a double purpose: (i) analysis of the observables (ii) and training of the teachers [1].

In order to allow for the analysis of the learner's activity and the context of the learning activity, the description of the learning activity is developed in the form of ontology in the computational sense that we will now explicit.

B. The ontology approach

Terminology: Ontologies, as used in the Semantic Web, allow to represent structured knowledge by defining concepts as well as relationships and hierarchies between these concepts. In the following, we use the Web Ontology Language (OWL) terminology:

(i) **Individuals** represent atomic, real-world objects
(ii) **Classes** represent concepts; a class may therefore be a collection of individuals (which are called instances of the class)
(iii) Individuals may be linked by relationships, which are labeled by **properties**. Properties may have a domain, i.e. the class that they can be applied to, and a range, i.e. the class that they can take values from.

OWL also defines both class hierarchy and property hierarchy.

This formalism allows to perform logical reasoning in order to validate the model, as well as logical inference to find all the assertions that can be deduced from the user-defined statements. Indeed, under certain constraints[2], OWL specifications correspond to description logics, providing computational completeness and decidability (a didactic introduction can be found in the guide [5]).

Ontologies in education: Here we consider the ontology approach to model a very specific problem-solving task, as it has been done with well-defined tasks such as the Tower of Hanoi [10]. A rather large literature is already available regarding the use of ontologies in e-learning systems, as reviewed in [11]. These approaches attempt to model the learning contents, or the learner (rarely both), in order to perform semantic inference on the observables (e.g [14]). They generally aim to propose adaptive learning resources in order to improve the learner's performances, while our main purpose is to better *understand* the learner behavior. Other related work includes the ontology specification of serious games [24] [3] which is inspiring although it does not apply to the description of unplugged activities. In our case, we want to use the OWL vocabulary to specify the #CreaCube task on two levels: (i) the material environment (cubelets, points...) and its affordances, and (ii) the learner in the context of this task, their behaviour and their cognition. A step further, we wish to query our model against the JSON data, using logical reasoning,in order to validate its relevance regarding the observable analysis. We also introduce here computational and cognitive neuroscience knowledge in the modeling process, which does not seem a current practice.

C. Borrowing from cognitive and learning sciences

Goal-driven vs stimulus-driven behavior: In order to advance in the understanding of creative problem solving, we

[2]Implemented in OWL DL, a sub-language of OWL
[3]Implemented in the Ludo Ontology: `http://ns.inria.fr/ludo`

propose to observe a specific problem-solving task — in this case, the aforementioned #CreaCube task — for which we aim to consider a functional neurocognitive model of the brain [3] and the specific knowledge related to the task in terms of the subjects' prior knowledge, the technological affordances they might discover and their potential interpretation in the process of problem solving (how these affordances might contribute to the task). The behavior of the participant in relation to her or their concurrent goals during the task (choosing among the sub-goals to be addressed in the task, performing the task as fast as possible, preserving their ego when facing problems, etc.) is considered in relation to two main potential strategies/behaviors to reduce the problem space into the solution-space: (i) a top-down goal-driven strategy (a sub-goal is considered and it is questioned which existing skills might be recruited to solve it) and (ii) a bottom-up stimulus-driven strategy (affordances generate stimuli that are selected and explored, potentially leading to revise the sub-goals). Globally, the behavior can be voluntarily goal-driven when the participant regulates the problem-solving process at the meta-cognitive level. However, we assume that the situation will engage the participant in stimulus-driven behavior in certain situations when the participant is lost or confronted with unknown stimuli (such as the robotic cubelets) that drive their actions to satisfy their curiosity. The stimulus-driven behavior could be considered within a larger goal-directed mechanism (see, e.g., [17]). We then assume that when a participant is engaged in a problem-solving activity, the main goal accepted by the participant is to solve the problem. The sub-goals for advancing towards one of the potential configurations which will satisfy the requirements of the problem, will engage the participant in behaviors which can sometimes be stimulus-driven, alternating with analysis of the stimuli based on the main goal of the task, exploiting previous knowledge and skills and also possibly with discovery and creativity [2].

Related cognitive functions in the brain: In addition, also based on the same neurocognitive model of the brain [3], we consider different ways of specifying and identifying a goal by defining (i) What is this goal (its sensory description), (ii) Why (and how) it answers current motivation or specification, (iii) Where it is located and can be accessed and (iv) How it can be manipulated and more generally addressed by a skill. These different motivational and sensori-motor aspects of information representation in the brain, as described in our cerebral architecture, represent an operational way to deal with all aspects of a goal to be considered in problem solving.

Exploration vs Exploitation: Having made the distinction between goal-driven and stimulus-driven behavior, we can now discuss the notions of exploration and exploitation. While this pair is well known in cognitive science and in machine learning at the level of reinforcement learning, they are two distinct conceptualizations that we will now make explicit. In problem-solving activities, subjects alternate between two main strategies: exploration, which aims at experimenting with new alternatives or at acting on the environment in order to generate new stimuli [13], and exploitation, which is the use of existing knowledge (declarative, procedural) in a given situation. At this level of description, we can draw parallels with these notions in reinforcement-based machine learning, but in this second domain exploration is mostly random, whereas in problem solving, exploration is seen as a strategy for generating new stimuli or new ideas or behaviors by recombining pre-existing knowledge or behaviors, or by manipulating the environment. These are therefore two quite distinct notions, one corresponding to an internal process of generating new internal representations, and the other to a behavior that is only defined in relation to a direct interaction with the environment.

III. Results

We implemented the ontology[4] using the software Protégé[5]. First, we have formalized, on one hand the classes corresponding to the material environment of the task, on the other hand the classes referring to the learner's cognition and behavior. Then, we have extensively instantiated these classes, listing their individuals based on the task design itself and the observations collected during the experiments. Finally, we have defined some properties to express the relationships between these instances in order to describe the task process from both a goal-driven and a stimulus-driven point of view.

A. Modeling the task and the material environment

The task involves several *objects* (cubelets, points), which have characteristics or *features* (colors of the cubelets, visual and functional *affordances*, *functions*, etc.) and *states* (assembled/disassembled cubelets, wheels facing the ground/not facing the ground, switch on/off, complete/incomplete cubelet configuration, etc.). These concepts allow to a class hierarchy that characterizes the task material as shown on Fig. 2.

The material states represent the problem space of the task in the sense of [19]: indeed, the goal states of the task are those where the cubelets are in a configuration that satisfies the guideline, i.e. in an "assembled" state, "stable" and "moving" "into the right direction", with the switch "on", and the wheels "on the ground". The learner may interact with the material and change its state through their actions, as described in the next section.

The objects have different functions (e.g the white cubelet rolls, the black cubelet detects nearby objects, etc). The learner may guess these functions, or at least some of them, by looking for clues provided by affordances. These affordances may be either perceptible (i.e. suggested by sensitive information obtained through the discovery of objects), or functional (i.e. discovered through the interaction of one object with others). In the #CreaCube task, the perceptible affordances are all visual. For example, the switch of the blue cubelet, or the wheels of the white cubelet are visual perceptible affordances. Conversely, the red cubelet has no specific visual affluence : there is nothing on its surface that shows its usefulness for the

[4]This ontology is publicly available here: `https://gitlab.inria.fr/line/aide-group/creaonto`.

[5]`https://github.com/protegeproject`

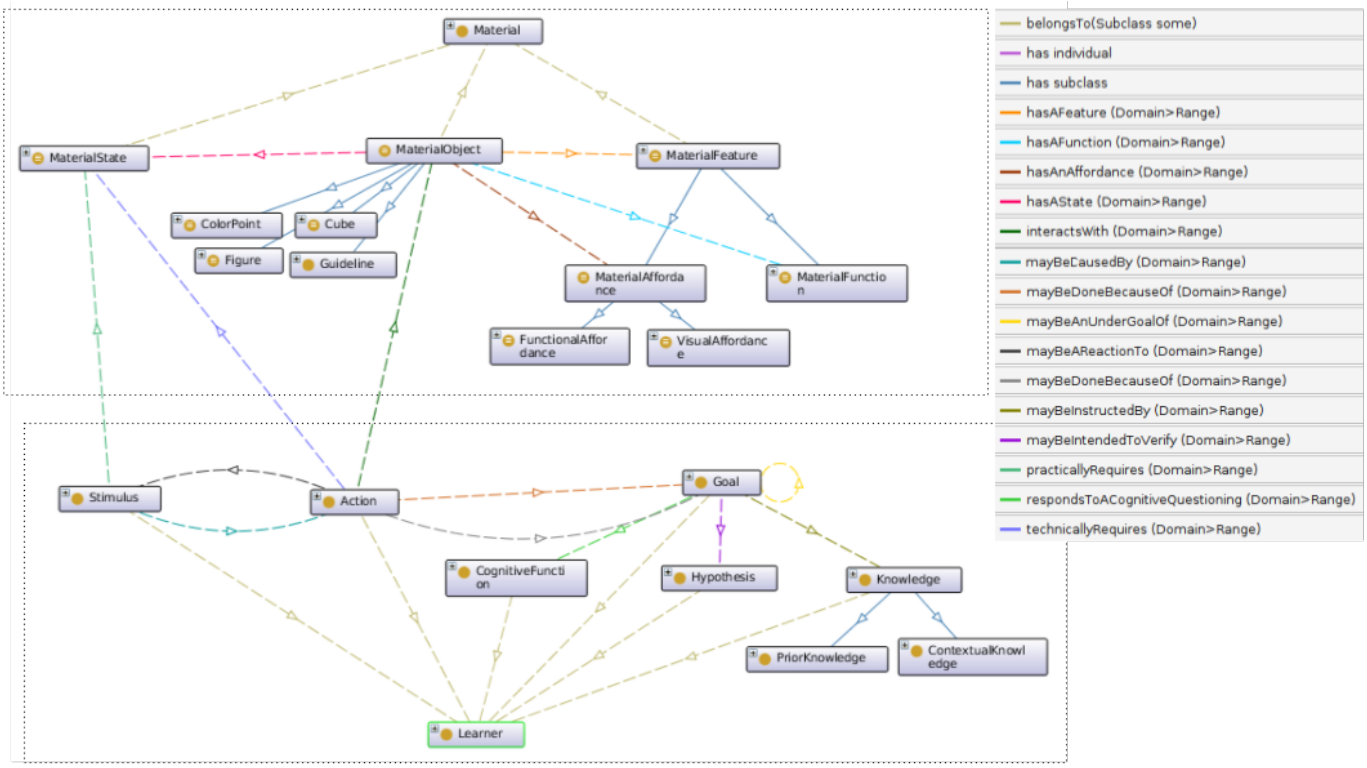

Fig. 2: Main classes defined in our ontology. Classes related to the task material are encompassed in the upper frame, while the lower frame includes classes that characterize the learner. Some of the properties are also represented through their domains and ranges, to show how those classes are related to each other. (Figure edited using the Protégé OntoGraph plugin; lower-level classes and individuals are not shown)

task at hand. However, by making it interact with the other cubelets, the learner may understand its functionality and its operating principle, which is inverting the sensor's behaviour,: hence disclosing a functional affordance. Affordances participate in the construction of memory traces, since they allow the learner to discover, little by little, their material environment and its logic, in order to know how to use it to solve the problem.

B. Modeling the learner's cognition and the knowledge construction

Stimuli and actions: Throughout the task, the learners discover and manipulate the material environment, from which they receive various stimuli, i.e. events that cause them organic reactions. In #CreaCube, the stimuli are sensory in nature and give the learner new information about the material environment of the task; this information can then be processed by the learner to constitute traces of stimuli that are kept active during the task, before being consolidated as knowledge (as we will described further).

In the #CreaCube task, stimuli occur when the learner proactively interacts with their environment; for instance, when they flip the switch on the blue cubelet, a diode lights up as a response to their action. The only exception is at the very beginning of the task when the material is revealed (visual stimulus) and the guideline is played (auditory stimulus), without any previous action from the learner.

Some actions are only available to the learner in some particular states of the material environment; those states define the preconditions of the action. Similarly, post-conditions define changes of state that an action results in. These pre- and post-conditions (specified by the ontology properties, see III-C) also constitute knowledge to acquire.

Knowledge retrieval and acquisition: In order to complete the task, the learner needs to retrieve *prior knowledge* that has been acquired in the course of past experiences and learning. Most of the time, this prior knowledge is activated through the discovery of affordances: for instance, after finding of the wheels, the learner might retrieve the fact that wheels on a vehicle must be in contact with the ground to allow it to move. However, prior knowledge may not always be transferable to the situation at hand: for example, the fact that wheels automatically start to turn after the user has pulled the vehicle backwards is only applicable in certain situations - most often, this mechanism is found in children car toys. However, in this task, trying to press down on the white cubelet by manually rolling it backwards will not cause it to move forward autonomously. This piece of knowledge might even hinder the completion of the task, if it is too strongly anchored (i.e., if it considered to be a general truth when it is actually valid only in a limited set of situations): as long as the learner keeps on trying to make the wheels work this way, they will not go on their research any further.

Confronting their prior knowledge to stimuli perceived

through their interactions with the material, allow the learner to maintain a representation of their environment. We refer to this representation as *contextual knowledge* in our ontology, i.e. facts that are related to the specific context of the task. Some of this contextual knowledge is acquired when the learner listens to the instruction and is thus directly related to the goal; the other part is built up during the task and relates to the resources available to complete the task.

Hypotheses and goals: In order to refine their representation of the environment, the learner needs to understand which pieces of prior knowledge are effectively applicable in the context of the task and relevant to solve it. For example, the learner might want to test the hypothesis that the wheels on the white cubelet can be activated by pull-back action, because they have previously seen such a mechanism on other toys. The outcome of the pull-back action will allow them to either discard this hypothesis or confirm it and store it as contextual knowledge. Testing hypotheses thus define subgoals to acquire knowledge that will help solve the task. As previously described in section II-C, goals are a way for the participant to organize their understanding of the task and plan their actions hierarchically. Global goals (essentially responding to the cognitive questionings *What* and *Why*, in link with high-level executive functions [3]) may be instructed by an external direction — in this case, the guideline — or a metacognitive reflexion. These high-level goals yield new questionings, inducing (*Where*, *How*) behavior mechanisms [3] thus resulting in sub-goals at a more local level, in link with sensori-motor behavior.

C. Modeling the task process

Instantiating the concepts: Having defined the previous concepts, we have instantiated them by extensively (although not exhaustively) listing all possible hypotheses, actions, stimuli, goals... of the learner, as well as all states, affordances, functions... of the material. This instantiation was based on the task design itself, e.g. the cubelet operating modes, and on the expected behaviors predicted by the theory, but also updated during the course of the experiments to account for newly observed behaviors. These instances (also known as individuals, see II.B) are linked to each other through relationships that describe the interactions between the material and the learner, the knowledge construction and the goal hierarchy, all of these contributing to better understand the task process.

Describing relationships: Relationships between individuals are represented by properties for which we specify the domain and the range, i.e. the classes they link, as shown on Fig. 2. These relationships allow us, for example, to identify preconditions and post-conditions of an action (respectively represented by the properties `technicallyRequires` and `changesTheStateOf`), describe the goal intricacy (`mayBeAnUnderGoalOf`) or the knowledge dependence (`operationallyRequires`, `mayBeInducedBy`). These properties have been defined with regard to the goal-driven and stimulus-driven behaviors described in section II-C, as shown on Fig 3. It is worth noting that we are using modal verbs when the relationship may exist while not being systematic. This representation could be improved using a numerical weighting of RDF statements based on the notions of possibility and necessity (in the sense of [8]) as discussed in [16].

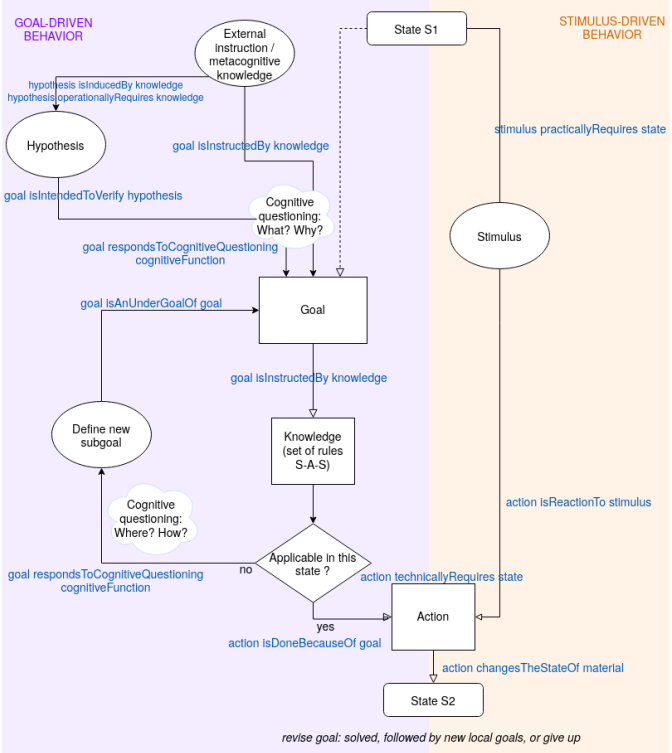

Fig. 3: Goal-driven behavior on the left versus stimulus-driven behavior on the right. Accordingly relevant properties defined in our ontology are represented in blue.

IV. Application

A. Modeling as an interdisciplinary tool

This work allowed us to confirm the multiple interest of working with ontological modeling when we bring together three disciplines (educational, cognitive and computer sciences) to tackle a problem:

- terminological interest to begin with: beyond periphrases and phenomenological descriptions in each discipline, constraining us to define things through lexical choices and fully specified properties and relations between these key words, forces us to clearly posit what we are talking about ;
- interest in formalization: this approach allows, without even using algorithmic reasoning skills at this stage, to take stock of what can be defined rigorously, to formalize completely.

Defining a well-formulated ontology is therefore already by itself a structuring exercise, an epistemological tool in a way, before even using it. As with any modeling, it gives an exhaustive and explicit view of what is to retain compared to what is neglected. Once the specification has been set, the use of available reasoners such as Pellet or Hermit allows to check

Snap SPARQL Query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/mnemosyne-line/aide/creacube#>

SELECT ?action ?stimulus ?goal ?supergoal ?knowledge ?hypothesis ?a
        WHERE {
                ?action :changesTheStateOf :Switch .
                OPTIONAL{
                        ?action :mayBeAReactionTo ?s .
                        ?s rdfs:comment ?stimulus .
                }
                OPTIONAL{
                        ?action :mayBeDoneBecauseOf ?goal .
                        OPTIONAL {
                                ?goal :mayBeAnUnderGoalOf ?supergoal .
                        }
                        OPTIONAL {
                                ?goal :isInstructedBy ?k .
                                ?k rdfs:comment ?knowledge .
                        }
                        OPTIONAL {
                                ?goal :mayBeIntendedToVerify ?h .
                                ?h rdfs:comment ?hypothesis .

                        }
                }
        }
```

(a) An example of SPARQL query

?action	?stimulus	?goal	?supergoal
:SwitchOnF		:GSwitchOnF	:GMakeFigureMoveAutonomously
:SwitchOnF		:GSwitchOnF	:GMakeFigureMoveAutonomously
:SwitchOnF		:GSwitchOnF	:GMakeFigureMoveAutonomously
:TrytoFlipSwitch	The switch of the blue cube is visible	:GFlipSwitch	:GFindFunctionCubeBe

?knowledge	?hypothesis
Switches are used to activate an electronic device	
Switches must be switched on to activate an electronic device	
The blue cube has a switch	
Switches must be switched on to activate an electronic device	The blue cube is a switch and permits to activate the vehicle

(b) SPARQL results returned for the query (a): we enquired about which actions change the state of the blue switch, what stimuli may have provoked them, or what goal they might be aiming at, and if that goal might be the sub-goal of another one, as well as what knowledge they are using and what hypothesis might be tested. We used the Protégé Snap-SPARQL plugin [6](which allows us to perform logical inference and answer queries not only based on asserted data but also inferred data).

Fig. 4: Example of inferences performed by the Hermit reasoner and retrieved by SPARQL.

its consistency and query the data on the consequences that may or may not be drawn from it. It also allows us to evaluate the order of magnitude of such development, showing that it is quite a huge work but feasible.

This being done, we also have an immediate check of the model properties, in particular its logical coherence and soundness. It not only allows us to verify the absence of contradictory statements, but we also experimented that if our definitions are not sufficiently specified, we can easily deduce spurious or odd consequences (e.g., mixing categories of concepts). We thus have an operational tool to verify the completeness of our formalization.

B. Reasoning about the model

1) Inferring internal observable of the learner from outer observable:

Having stored these behavioral processes in an ontology allows to retrieve them to interpret new data. A query language, such as SPARQL in the case of RDF data, may fulfill that purpose. For example, we may observe a new action which changes the state of the switch, and wonder what motivated the learner to execute this action. As illustrated by Fig. 4, a SPARQL query on the model helps us retrieve that the action may have been a simple reaction to the stimulus of seeing the switch, in the case of a stimulus-driven behavior, or, in a goal-driven mode, the result of a more elaborate reasoning such as an attempt to understand the function of the blue cubelet (thus testing the hypothesis that it is indeed a switch) or, this hypothesis being verified, to activate the movement of the vehicle.

Characterizing whether the learner is rather in a goal-driven or stimulus-driven mode for a given sequence of observables might be inferred from the timestamps recorded in the data, thus helping us to decide which statements of the ontology are more appropriate to describe the observed situation. For example, we could assume that once the learner has identified a goal and the means and resources at their disposal to fulfill it, the flow of their actions might be smoother, whereas in an exploration situation, there might be more pauses corresponding to the wait between new stimuli. However, this link between timestamps and switch of strategy is not well understood yet and needs further investigation.

2) Predicting the learner's behavior:

Instead of querying the model using backward chaining as in the previous example, we can also use forward chaining in order to simulate the learner's deduction, hence predicting their behavior. For example, we might wonder how the learner would react to the visual stimulus of seeing the switch. In a stimulus-driven mode, they would likely touch it or try to flip it (assuming that the knowledge relative to switches being flipped has become a procedural knowledge, that is to say automated into an almost reflex behavior). In a goal-driven mode, we may infer that the learner will elaborate the hypothesis that the affordance is indeed a switch and can be used to activate the vehicle, which will result in planning goals (flip the switch, connect the cubelet to the figure) and sub-goals (they have to grasp the cubelet before connecting it to the others).

Again, this ontology provides a subset of plausible behaviors, however we still need a way to decide which of these behaviors to choose to simulate the learner. Such mechanisms are still to be developed to explain what the learner's attention is more focused on in a given situation. Moreover, we need to take into account inter-individual differences: some participants might be more interested in achieving a performance goal, thus appealing for a predominant exploitation strategy, whereas others might pursue a mastery goal and try to understand as much as possible about the environment of the task, displaying more exploratory behaviors.

V. Discussion and Conclusion

This study introduces the operationalization of a creative problem-solving task. Within this type of task, we need to consider first extrinsic motivation (e.g., task performance)

[6]https://github.com/protegeproject/snap-sparql-query

and intrinsic motivation (e.g., increase knowledge), i.e., the "Why". Then consider the "What", i.e., both exploration and exploitation strategies directed toward different concurrent goals (performance goal versus mastery goal in relation to the knowledge related to the task). Our ontology formalizes these behaviors involved in the creative problem-solving process to model the learner's behavior in interaction with the educational robots, with regard to a cognitive model of the brain. Through this process, we take into account the stimuli received during the interactions with the modular robots, the perception and interpretation of the different technological affordances that will support the problem-solving activity through concrete actions to be developed. The learner's prior knowledge is mobilised to understand current perceptions and transform knowledge into a way that makes sense within the context of task [18], formulating and testing in-task hypotheses.

The key point is that the description does not remain phenomenological, that is to say, guided by a theoretical framework expressed in human language to guide the analysis of the data, while we hereby reformulate these notions at a computational level to give oneself the means to process them in a more systematic way using an algorithmic approach, as discussed here, formalizing in a precise language the concepts invoked. We also consider, based on this formalism, biologically plausible implementations of internal reasoning mechanisms in a companion work [16] in order to contribute to cognitive neuroscience modeling of related mental processes.

This is only a first step, which already shows that formalizing such a learning task is definitely a rather huge, but attainable work. Next steps will be to complete this formal description, and develop reasoning mechanisms to better infer the subject internal motivation and behaviors. We will also complete the linking between this task and learner modeling and experiment observables (e.g., learner state observation, post-activity evaluation).

Though we have the feeling that it is a rather disruptive approach with respect to learning science, we are really aware that it is still a very preliminary work. Eventually, the key point is to consider a triple multi-disciplinary approach from learning science, cognitive neuroscience and computer science, something that might be called computational learning science [20].

Acknowledgments

Hugo Chateau-Laurent and Lola Denet are highly acknowledged for their contribution to scientific discussions in link with this work.

REFERENCES

[1] B. Albero and J. Guérin. L'intérêt pour l'activité en sciences de l'éducation. Vers une épistémologie fédératrice ? TransFormations : Recherches en éducation et formation des adultes, 11:11–45, 2014. Publisher: Institut CUEEP, Lille 1.

[2] F. Alexandre. Creativity explained by Computational Cognitive Neuroscience. In ICCC'20 - International Conference on Computational Creativity, page 4, Sept. 2020.

[3] F. Alexandre. A global framework for a systemic view of brain modeling. Brain Inf., 8(1):3, Dec. 2021.

[4] F. Alexandre, J. Becker, M.-H. Comte, A. Lagarrigue, R. Liblau, M. Romero, and T. Viéville. Open Educational Resources and MOOC for Citizen Understanding of Artificial Intelligence, July 2020.

[5] D. Allemang, J. Hendler, and F. Gandon. Semantic Web for the Working Ontologist. ACM, June 2020.

[6] S. Barnabé, L. Denet, M. Manrique, D. Menon, E. Pascual, M. Romero, and T. Viéville. A low-cost tabletop set-up to collect learning analytics during computational thinking unplugged or tangible activities, Nov. 2020. In preparation.

[7] M. T. H. Chi and R. Wylie. The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. Educational Psychologist, 49(4):219–243, Oct. 2014. Number: 4.

[8] T. Denœux, D. Dubois, and H. Prade. Representations of Uncertainty in AI: Beyond Probability and Possibility. In P. Marquis, O. Papini, and H. Prade, editors, A Guided Tour of Artificial Intelligence Research: Volume I: Knowledge Representation, Reasoning and Learning, pages 119–150. Springer International Publishing, Cham, 2020.

[9] Y. Engestrom. Activity theory as a framework for analyzing and redesigning work. Ergonomics, 43(7):960–974, July 2000.

[10] H. Eriksson, Y. Shahar, S. W. Tu, A. R. Puerta, and M. A. Musen. Task modeling with reusable problem-solving methods. Artificial Intelligence, 79(2):293–326, Dec. 1995.

[11] G. George and A. M. Lal. Review of ontology-based recommender systems in e-learning. Computers & Education, 142:103642, Dec. 2019.

[12] W. Huang and C.-K. Looi. A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. Computer Science Education, 0(0):1–29, July 2020. Publisher: Routledge _eprint: https://doi.org/10.1080/08993408.2020.1789411.

[13] A. Kaplan and H. Flum. Achievement goal orientations and identity formation styles. Educational Research Review, 5(1):50–67, Jan. 2010.

[14] A. Lebis, M. Lefevre, V. Luengo, and N. Guin. Recherche intelligente de processus d'analyse de traces d'e-learning via des inférences sémantiques. In Journées francophones d'Ingénierie des Connaissances (IC), Toulouse, France, July 2019.

[15] M. Lodi. Introducing Computational Thinking in K-12 Education: Historical, Epistemological, Pedagogical, Cognitive, and Affective Aspects. phdthesis, Dipartimento di Informatica - Scienza e Ingegneria, Alma Mater Studiorum - Università di Bologna, Apr. 2020.

[16] C. Mercier, H. Chateau-Laurent, F. Alexandre, and T. Viéville. Ontology as neuronal-space manifold: towards symbolic and numerical artificial embedding. In CIKM'2021, Nov. 2021. submitted.

[17] A. Moors, C. Fini, T. Everaert, L. Bardi, E. Bossuyt, P. Kuppens, and M. Brass. The role of stimulus-driven versus goal-directed processes in fight and flight tendencies measured with motor evoked potentials induced by Transcranial Magnetic Stimulation. PLoS ONE, 14(5):e0217266, May 2019.

[18] M. Musial and A. Tricot. Précis d'ingénierie pédagogique. PEDAGOGIES EN DEVELOPPEMENT. De Boeck Supérieur, 1st edition, Feb. 2020.

[19] A. Newell and H. A. Simon. Human problem solving. Prentice-Hall, Englewood Cliffs, N.J., 1972. OCLC: 622041645.

[20] M. Romero, F. Alexandre, T. Viéville, and G. Giraudon. LINE - Mnémosyne : Des neurosciences computationnelles aux sciences de l'éducation computationnelles pour la modélisation du cerveau de l'apprenant et du contexte de l'activité d'apprentissage. Bulletin de l'Association Française pour l'Intelligence Artificielle, (108), Apr. 2020.

[21] M. Romero, D. David, and B. Lille. CreaCube, a Playful Activity with Modular Robotics. In M. Gentile, M. Allegra, and H. Söbke, editors, Games and Learning Alliance, volume 11385, pages 397–405. Springer International Publishing, Cham, 2019. Series Title: Lecture Notes in Computer Science.

[22] M. Romero, A. Lepage, and B. Lille. Computational thinking development through creative programming in higher education. Int J Educ Technol High Educ, 14(1):1–15, Dec. 2017. Number: 1 Publisher: SpringerOpen.

[23] M. Romero, T. Vieville, and L. Heiser. Analyse d'activités d'apprentissage médiatisées en robotique pédagogique. In Traité de méthodologie de la recherche en Sciences de l'Éducation et de la Formation. B. Alberto and J.Thievenaz, 2021. Publisher: Unpublished.

[24] S. Tang and M. Hanneghan. Game Content Model: An Ontology for Documenting Serious Game Design. In 2011 Developments in E-systems Engineering, pages 431–436, Dec. 2011.

[25] J. M. Wing. Computational thinking. Commun. ACM, 49(3):33–35, Mar. 2006.