LEAP: A Model for Learning Engineering Practices Through Abstraction at Progressive Levels

Aditi Kothiyal[a,c,*], Shitanshu Mishra[b,c], Sahana Murthy[c], Sridhar Iyer[c]
[a]Department of Humanities and Social Sciences, Indian Institute of Technology Gandhinagar,
[b]UNESCO MGIEP
[c]IDP in Educational Technology, Indian Institute of Technology Bombay
*aditi.kothiyal@iitgn.ac.in

Abstract

In recent years the emphasis of engineering education has shifted to the learning of disciplinary practices such as design, estimation, troubleshooting and so on. These practices are challenging for undergraduate students to learn and it is important to support students in developing expertise in engineering practices. Therefore, there is a need for a model of how expertise in engineering practices is developed, that researchers, teachers and instructional designers can use to design effective technology-enhanced learning environments to support learners in acquiring expertise in engineering practices. This paper presents a model for how expertise in engineering practices may be acquired, called the LEAP model, developed using a design-based research methodology. The model is based on the theoretical foundations of the nature of engineering practices and the role of metacognition in engineering practice. The model proposes that a learner develops expertise by going through four levels of cognitive processing and iterating between the levels using monitoring and control processes. The model was applied to design two technology-enhanced learning environments for engineering practices. Evaluation studies showed that learners developed expertise in the respective engineering practices through the learning pathway proposed in the LEAP model. We discuss how to apply this model in design and implications for research and practice.

Introduction

Engineering practice has been defined as the “complex, thoughtful and intentional integration of these [problem solving process and specialized knowledge] towards some meaningful end” (Sheppard et al, 2006). Engineers solve different types of ill-structured problems in the workplace in which the path of action is not clear or pre-defined, multiple solutions exist, multiple criteria need to be applied, uncertainties exist, self-regulation and judgement are required, and the goal is to find meaning or structure in disorder (Jonassen, 2000). The nature of the goal or the “meaningful end” determines which specialized knowledge and problem solving processes should be applied, and how knowledge and processes should be integrated (Vincenti, 1990; Sheppard et al, 2006), thus giving rise to different engineering practices. Thus, practices are the tools of the engineering profession. Some examples of engineering practices are engineering estimation, troubleshooting, design inquiry and conceptual design (Dym et al, 2005; Rivera-Reyes & Boyles, 2013; ABET, 2014; Pahl & Beitz, 2013).

As elaborated in the literature of engineering practices such as design thinking, troubleshooting and engineering estimation (Dym et al, 2005; Rivera-Reyes & Boyles, 2013; Hay et al, 2017; Kothiyal et al, 2016), different practices require the application of different cognitive processes, such as mental simulation, concept identification, comparison and generating questions (Ball & Christensen, 2009; Hay et al, 2017; Kothiyal et al, 2016; Dym et al, 2005). These cognitive processes are applied to create artefacts such as functional models and design concepts, in order to achieve the goals of the problem (Vincenti, 1990). In addition, research on engineering practices has shown that they require metacognitive processes in order to monitor and control the cognitive processes such that the goals are successfully met (Schon, 1983; Ball et al, 1997). These metacognitive processes include knowing when to apply the cognitive processes and how to evaluate the artefacts created (Mayer, 1998; Holton & Clarke, 2006; Efklides, 2009). Thus engineering practices require an application of cognitive and metacognitive processes, and so learning any practice involves the use of multiple cognitive and metacognitive processes and switching between these processes appropriately.

Engineering educators and practitioners agree that the goal of engineering education has changed from learning engineering knowledge to the development of engineering practices (Rugarcia et al, 2000; Sheppard et al, 2006; Froyd et al, 2012). Expertise in these practices is the hallmark of expertise in engineering (ABET, 2014) and hence it is important for students to learn these engineering practices. However, research suggests that these practices are challenging for undergraduate engineering students and not automatically learned (Linder, 1999; Rivera-Reyes & Boyles, 2013; Dym et al, 2005; Atman et al, 2007; Moore et al, 2013). Therefore, it is important to explicitly support students in developing expertise in engineering practices. This has led to a recent emphasis on engineering practices in the engineering curricula (Woods et al, 2000) and triggered extensive research on supporting the development of engineering practices among students. Several teaching and learning environments for design and problem solving have been designed and empirically evaluated for their effectiveness (for instance, Woods et al, 1997; Dym et al 2005). These environments broadly support both the cognitive and metacognitive processes of the practice while students are learning. However, research on professional practices such as design and therapy suggests that cognitive and metacognitive processes interleave intricately in any practice and the practice becomes a "reflective conversation with a unique and uncertain situation." (Schon, 1983). Therefore, learning environments for engineering practices must take novices through a learning pathway wherein cognitive and metacognitive processes interleave intricately during the learning of a practice.

In this paper, we propose a learning pathway for engineering practices which includes different kinds of cognitive and metacognitive tasks, explicitly interleaving the two. We conjecture that the frequent interleaving of cognitive and metacognitive processes supports the learning of practices by taking learners through progressively more abstract aspects of the practice. In the next section, we review literature of the teaching-learning of engineering practices, identify the gaps within this literature that necessitates a new learning design and situate our work within this literature. In the subsequent sections we describe the theoretical foundations on which this work
is based, detail the learning design and present illustrative case studies of the design and evaluation of two such learning environments. Finally, we conclude by synthesizing the findings from our studies to propose a learning pathway for the development of engineering practices. We discuss the implications of this pathway for educational technology researchers, teachers and instructional designers.

Related Work: Teaching and Learning of Engineering Practices

Researchers agree that there are practices such as design, troubleshooting and estimation which are employed across domains in engineering (Vincenti, 1990; Sheppard et al, 2006; Rugarcia et al, 2000; ABET, 2014). One of the central debates in engineering education therefore, has been about whether practices should be taught directly, without emphasizing the domain content or whether the learning of practices should be embedded in the domain content (Woods et al, 2000). Further, there are two perspectives on engineering practices which dictate how instruction is designed: the perspective of practices as tools for learning engineering concepts vs practices as tools of work (Sheppard et al, 2006). In the former perspective, practices are taught as means to better learn domain content, while in the latter practices are taught as means to better work as an engineer. In this paper, we adopt the perspective that practices are tools for work in engineering. Additionally, we consider that the learning of engineering practices should be necessarily embedded in the domain content because learning is known to be more effective when anchored or situated in authentic engineering contexts (Johri & Olds, 2011).

The problem of designing instruction for developing a practice within an engineering domain has been extensively researched for engineering design. There has been a lot of research to characterize design thinking and its cognitive processes (for instance, Dym et al, 2005; Atman et al, 2007; Ball et al, 1999, 2004, 2009; Hay et al 2017). The findings from research on design cognition have been used in order to identify teaching-learning principles for engineering design (Crismond & Adams, 2012). The main principle followed in engineering design education has been "learning while designing" and so design has been taught through project-based learning (Perrenet et al., 2000) at the undergraduate (for instance, Dutson et al, 1997; Dym et al 2005; Benjamin & Keenan, 2006) and K-12 levels (for instance, Chiu et al, 2013; Moore et al, 2014). Students learn by engaging collaboratively on real-world design tasks within a project-based learning environment. Appropriate strategies and scaffolds are used to support students' engineering design processes such as, making students do product teardowns or reverse engineering (Sheppard & Jenison, 1997), teaching design ideation strategies (Daly et al, 2011), supporting visualization and simulation through computational tools (Crismond, Howland, & Jonassen, 2011; Mandinach & Cline, 1996), making the process of informed engineering design explicit (Chiu et al, 2013) and reflecting through the use of diaries (Kolodner et al, 2004). The focus has been on helping students learn design processes practised by informed designers (Crismond & Adams, 2012) with the assumption that by learning the processes of informed designers, novices would acquire expertise. Thus these interventions have been based on studying and understanding expertise, and fostering the productive processes of design that characterize expertise.

Apart from engineering design, there has been widespread research on developing students' generic engineering problem solving practice (for instance, Bozic et al., 2014; Kalnins et al., 2014; Stojcevski, 2008; Wankat & Oreovicz, 2015; Woods et al., 1997). The focus is on student learning of a general engineering problem-solving strategy (Woods, 2000; Wankat & Oreovicz, 2015) or "structured procedures" (Vincenti, 2000). These strategies typically have five or six stages, namely, define the problem, explore, plan, execute and reflect, applied iteratively. In addition, students learn and practice component problem solving skills (Woods et al, 1997), such as "define the problem" and "getting unstuck" within a domain and apply these across domains via bridging activities, constant monitoring and reflection, gradually building up to solving ill-structured engineering problems. These interventions are typically grounded in problem and project-based learning (De Graaf & Kolmos, 2003; Perrenet et al., 2000) and include variations depending on whether students receive instruction in improving component problem-solving skills along with, or prior to, attempting the problems (Pimmel, 2001; Woods et al., 1997).

The teaching-learning interventions discussed above largely focus on supporting students in learning a general problem-solving strategy (for instance, Woods, 2000). While strategies are effective to ensure systematicity and are a hallmark of expertise (Woods, 2000), what is largely missing in these interventions is the emphasis on the cognitive processes of a practice that experts use within each phase of a strategy. Another issue is that strategies make a practice seem modular, wherein metacognition is relegated to certain phases. Research however suggests that metacognition is very integral to expert practice and happens intermittently and frequently (Schon, 1987; Litzinger et al, 2011; Adams et al, 2008; Ball et al, 1997; Davidson, Deuser and Sternberg, 1994). Further, while it may be true that there are certain component skills or "ways of thinking" (Vincenti, 1990) in engineering which are widely applied, it is known that these are not applied in the same manner in each practice. As literature on engineering practices such as estimation, troubleshooting and design suggests, each practice requires certain cognitive and metacognitive processes applied together in specific ways to meet the goals of the problem (Rivera-Reyes & Boyles, 2013; Kothiyal & Murthy, 2016; Atman et al, 2007). Thus, the general problem-solving interventions (for instance, Woods et al, 1997; Wankat & Oreovicz, 2015) may be too broad to be useful for all engineering practices. Apart from engineering design for which the component cognitive processes have been identified and their learning supported among students, this has not been done for other engineering practices.

A framework for instructional design of complex tasks called the "Ten Steps to Complex Learning" (Kirschner & van Merriënboer, 2008) provides a blueprint for designing a course or curriculum involving complex learning, defined as "the integration of knowledge, skills and attitudes; coordinating qualitatively different constituent skills; and often transferring what was learned in school or training to daily life and work". The approach is based on identifying the cognitive strategies for doing tasks and mental models of the underlying domain in order to provide supportive and procedural information to students as they practice doing whole and part tasks. This framework can be used to design instruction for engineering practices because they are a type of complex learning, especially because of the emphasis within the framework on identifying the cognitive strategies underlying the tasks. However, the emphasis in this framework remains on the cognitive strategies involved in the tasks, without explicitly designing for the metacognitive processes and the intertwining of cognitive and metacognitive processes that is required for expertise in engineering practices (Schon, 1987; Litzinger et al, 2011; Adams et al, 2008; Ball et al, 1997; Davidson, Deuser and Sternberg, 1994). Literature has extensively discussed the importance of metacognition and scaffolding for metacognition in learning (for instance, White & Frederiksen, 2005; Etkina et al, 2010; Holton & Clarke, 2006).

In our previous work, we designed, developed and evaluated technology-enhanced learning environments (TELEs) for the practices of structuring open design problems (Murthy, Iyer and Mavinkurve, 2016) and macro-micro thinking (Kenkre & Murthy, 2017) by analysing expert cognitive and metacognitive processes and identifying learning activities to support these processes. Empirical studies showed that the TELEs supported learners' in developing expertise in each of the practices (Murthy, Iyer and Mavinkurve, 2016; Kenkre & Murthy, 2017). Reflection on the learning mechanism revealed that while the TELE supported the cognitive and metacognitive processes involved in the practice, the nature of the cognitive-metacognitive interleaving needs to be further unpacked. As an example, consider the practice of structuring open design problems (Murthy, Iyer and Mavinkurve, 2016) which includes the cognitive processes of decision making, concept integration and synthesis. The metacognitive processes need to make learners aware that they need to apply a particular cognitive process, i.e. whether they need to take a decision or connect concepts to specifications or synthesize the problem specifications with procedures and concepts. In our studies, we observed that this interleaving was not explicitly supported in the TELEs. As a result, the nuances of the learning mechanism of the TELE were not well-understood.

Together our literature review and previous work suggests two issues within the research on teaching and learning of engineering practices. Firstly, it is not clear whether current teaching and learning interventions support the learning of all the cognitive and metacognitive processes that characterize expertise in a practice. Secondly, the interleaving of cognition and metacognition is supported at a much coarser scale, if at all, in these interventions, compared to actual practice, and so it is not evident whether and how learning happens. Together, these two issues point to the fact that there is the need for a learning design for engineering practices that explicitly support all the cognitive and metacognitive processes of a practice, and the frequent interplay between the two. Such a learning design could suggest how the learning of engineering practices such as troubleshooting (Ross & Orr, 2009) and engineering estimation (Nachtmann & Lehrman, 2002) might be supported, because these practices have received little attention in education and learning sciences literature. Teachers and instructional designers can use this learning design to design learning environments for engineering practices of their interest.

Building on our previous research (Murthy, Iyer and Mavinkurve, 2016), we incorporate literature on the role of metacognition in the doing and learning of engineering practices (Schon, 1987; Litzinger et al, 2011; Adams et al, 2008; Ball et al, 1997; Davidson, Deuser and Sternberg, 1994; White & Frederiksen, 2005; Etkina et al, 2010; Holton & Clarke, 2006; Efklides, 2009) in order to propose a learning design using which the development of expertise in engineering practices via the interplay between cognitive and metacognitive processes might be supported. We elaborate on the characteristics and purposes of metacognitive processes in engineering practices, and how they must be intertwined with cognitive processes such that novices develop expertise in engineering practices. In the next section we describe the theoretical foundations of our learning design.

Theoretical Foundations

Engineering practices require executing a set of cognitive processes and the application of these cognitive processes must be coordinated and integrated in order to achieve the goal of the task; this coordination and integration is done via the metacognitive processes of monitoring and controlling the progress towards the goal. For instance, conceptual design requires cognitive processes such as mental imagery, concept generation, retrieval and association of representations and evaluation and synthesis of concepts (Hay et al, 2017). However, these are interleaved with the processes of intermittently monitoring whether the generated concepts are appropriate for the design goal and monitoring the progress towards the final design (Hay et al, 2017; Crismond & Adams, 2012). Below we elaborate on the nature of the cognitive and metacognitive processes of an engineering practice, and what literature tells us about how these might be interleaved in expert engineering practice.

Cognitive Processes Underlying Engineering Practices

Engineering practices require the integration of problem-solving processes and disciplinary knowledge in order to accomplish a particular task (Sheppard et al, 2006). Each of the problem-solving processes, in turn, require application of multiple cognitive processes and creation of artefacts which lead to achieving the goal of the task. For instance, consider the practice of conceptual design, which requires cognitive processes such as retrieving from long-term memory, mental imagery processing, concept generation, synthesizing concepts, evaluating concepts and decision-making (Hay et al, 2017). Creating a conceptual design that meets the given specifications happens via retrieving from long-term memory and mental imagery processing to generate design concepts, evaluating the concepts to ensure that they satisfy the design requirements, choosing appropriate concepts and synthesizing them to create the design. The designer must be able to apply each of these cognitive processes of conceptual design in order to complete the design task successfully. In summary, an engineering practice requires the application of each of the cognitive processes necessary for that practice, evaluating the artefacts resulting from applying the cognitive processes to ensure they are aligned with the goals and integrating these cognitive processes such that the goal of the task is achieved (Litzinger, 2011).

Metacognitive Processes Underlying Engineering Practices

Lipman (2003) argues that reflection is necessary for the improvement of thinking. Vygotsky (1978) described how the “inward speech” of an individual during a problem solving task is a problem solving tool that allows the solver to analyze, plan and reflect on possible paths to a solution. This inward speech thus plays the roles associated with metacognition, such as awareness and control of the cognitive processes (Fox & Riconscente, 2008). In problem solving, for example, metacognitive processes are needed in every stage, namely identifying and defining the problem, building mental representations of the problem, planning how to proceed and evaluating what you know about your performance (Davidson, Deuser and Sternberg, 1994). It is known that experts seamlessly interleave metacognitive processes with cognitive processes in order to achieve the goal of the task (Schon, 1987; Litzinger et al, 2011; Adams et al, 2008; Ball et al, 1997; Davidson, Deuser and Sternberg, 1994). Returning to the example of conceptual design, the designer must use their internal speech to constantly monitor the generated concepts and ask if these concepts will meet the specifications. The designer must recognise that to answer the question they need to perform the cognitive process of evaluating the concepts, execute the process, obtain the result of the evaluation and then decide on the next suitable cognitive process to apply. Thus, in order to acquire expertise in an engineering practice, novices must learn to leverage this inward speech to monitor their work, identify what cognitive processes are needed to solve the tasks, and when and how to apply those (Fox & Riconscente, 2008).

We draw on the extensive literature regarding models of metacognition (for instance, Flavell, 1979; Nelson & Narens, 1996 and reviews in Efklides, 2008, 2009) in order to understand the roles of metacognition in engineering practices. This leads us to conjecture how metacognitive processes must be interleaved with cognitive processes during a task. We begin by summarizing the salient ideas about metacognition that we apply in this paper.

Broadly, metacognition refers to an individuals’ knowledge concerning their own cognitive processes and products (Flavell, 1979). Metacognition is thus a model of cognition that serves two purposes, namely, monitoring and control of cognition (Flavell, 1979; Nelson & Narens, 1996). Nelson & Narens (1996) described cognitive processes as splitting into two interrelated levels, namely the “object” level and the “meta” level”, which has a model of the object level. The “meta” level is informed by the object level, and the information flowing from the “meta” level to the “object” level changes the state of the object-level process or changes the object-level process itself. Thus there is a directionality to the flow of information between these two levels with the “meta” level “monitoring” and “controlling” the object level (Figure 1). This view of metacognition can be extended to more than two levels, where every level is the “object” for the higher level. The “meta” level consists of metacognitive knowledge (MK), metacognitive experiences (ME) and metacognitive skills (MS) (Figure 1, Efklides, 2008). MK is declarative knowledge which includes models of cognitive processes, while ME are an individuals’ online feelings, judgements and task-specific knowledge. MS are the skills deliberately used by an individual to control his/her cognitive processes and include orientation strategies, planning strategies, strategies for regulating cognitive processing, strategies for checking the implementation of actions, strategies for evaluating outcomes and self-regulation strategies (Efklides, 2009). Efklides (2008) proposed a multilevel and multifaceted model of metacognition, consisting of the non-conscious, personal awareness and social levels of metacognition. Engineering practices require individuals working at the personal awareness level of metacognition since they require conscious application of cognitive processes.

Taken together, literature on metacognition suggests that as an individual works on a task (object level, Figure 1), monitoring of the cognitive processes being executed is reflected in the individuals' ME (Efklides, 2009). This ME takes the individual to the "meta" level where, based on their MK, the goals of the task and the cognitive processes, they take decisions such as whether and what changes need to be made. These decisions are then implemented through appropriate MS (controlling the cognitive processes). For example, in conceptual design a solver retrieves from memory or applies mental imagery to generate new concepts. Online monitoring of the concepts (via inward speech, i.e. asking themselves, "Is this concept applicable to this design?") takes them to the meta level, where they apply their knowledge of quality criteria for design concepts and the goals of the task to evaluate the usability of the generated concept. Depending on the result of this evaluation, they may modify the concept or generate a new concept or move on in the design task. In the next subsection, we describe how we built on this literature of metacognition, shown in Figure 1, to design learning environments for engineering practices.

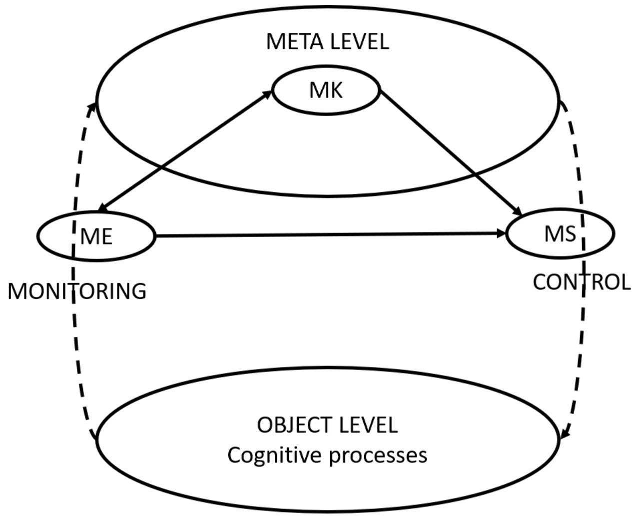

Figure 1: Model of Metacognition (Adapted from Efklides, 2008)

Learning of Engineering Practices by Interleaving Cognitive and Metacognitive processes: Progressive Abstraction

Synthesizing and extending the above literature, we argue that an engineering practice can be considered as one particular way of applying certain cognitive processes, and monitoring and controlling the cognitive processes in order to complete a task. The exact way will depend on the specific task - its goal, the domain knowledge and the cognitive processes involved - all of which constitute the "object" cognitive level. However, there are certain mechanisms which are common across engineering practices, such as, regularly monitoring the created artefacts to evaluate whether the appropriate cognitive processes are being correctly applied and in the right sequence so that the solution is aligned to the goal. Based on this monitoring, appropriate strategies must be applied to control the cognitive processes in the "object" level in order to improve the solution. As an example, consider the practice of engineering estimation wherein, as explained in Kothiyal et al (2016), the main cognitive processes are mental simulation, creation of external representations and semantic knowledge retrieval to build models. The practitioner creates a model, asks herself (monitoring) whether the model is useful for the given estimation task and evaluates this by checking whether the model includes all relevant functions and structures for the estimation. Depending on the result of the evaluation, the practitioner either revises the model or creates another model. Intermittently, the practitioner also asks herself whether she is moving in the right direction (monitoring) and synthesizes the cognitive processes done until now to choose the next cognitive process. Thus estimation practitioners work at three cognitive levels – the modelling level, the model evaluation level and the model synthesis level.

As literature suggests, the “meta” cognitive level contains an abstraction of the “object” cognitive level. In the case of engineering practices, this abstraction varies depending on the progress towards the task goal and serves different purposes in the practice. The first “meta” cognitive level contains an abstraction of the “object” cognitive level for evaluating whether the cognitive processes are correctly applied to create an artefact that is aligned with the goal of the task (Figure 2). The second, more abstract “meta” cognitive level contains an abstraction for synthesizing the cognitive processes done until now, to choose the one that must be applied next to progress towards the goal. Finally, while learning or developing expertise, novices must work at a third more abstract “meta” cognitive level, which involves abstracting a generalizable sequence of cognitive processes necessary to achieve the goal. This last level allows the novice to do the practice in novel but similar situations. Combining these four levels, we propose the LEAP model to develop expertise (Figure 2) according to which a novice must be taken through the cognitive processes at these four levels, and the monitoring and control processes which take them between levels. We argue that this periodical movement between the progressively more abstract cognitive levels via the monitoring and control processes supports the development of expertise. In the next section we describe two cases of learning environments designed to take learners through a sequence of tasks at progressively more abstract levels, interleaved with monitoring and control processes.

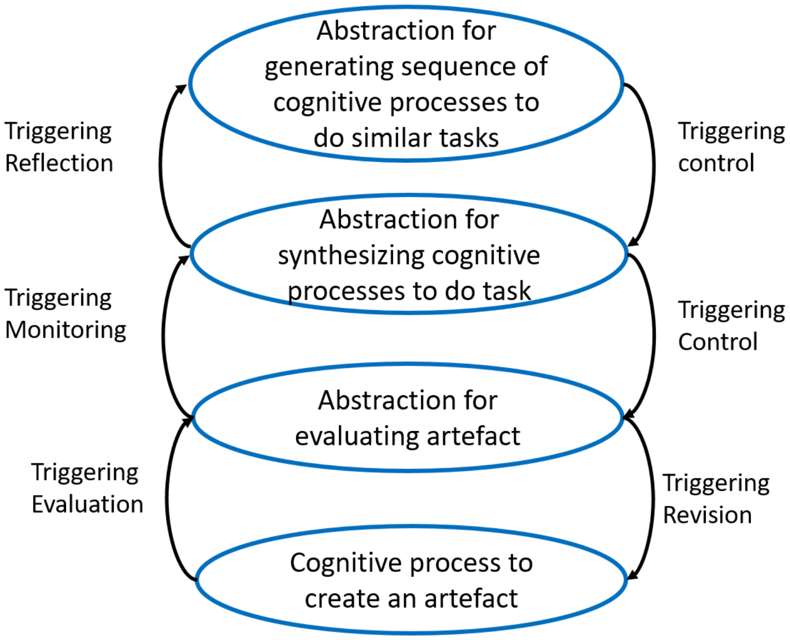

Figure 2: Overview of LEAP model

Illustrative Cases

We illustrate the process of development of practices via the LEAP model through the examples of the practices of engineering estimation and knowledge integration.

Engineering Estimation

Engineers routinely make estimates of physical quantities such as power before they begin designing or making (Dym et al., 2005). Estimation is an important problem in engineering education that has not been extensively addressed (Kothiyal & Murthy, 2018). However, the characteristics of expertise in estimation have been studied and it has been identified that the cognitive process underlying estimation is modelling (Kothiyal et al, 2016). To estimate a quantity, experts make three models of the problem system, namely functional, qualitative and quantitative models, and integrate them in order to obtain an estimate (Kothiyal et al, 2016). Each of the three models (functional, qualitative and quantitative) serves a particular purpose in the estimation and integrated together lead to the final estimate.

Based on the understanding of expertise and the LEAP model (Figure 2), we conjectured that in order to learn the practice of estimation, novices need to do each modelling process of estimation multiple times and synthesize them to learn estimation. We instantiated this conjecture into the Modelling-based Estimation Learning Environment (MEttLE), which supports the tight interplay between cognitive, monitoring and control processes necessary to make an estimate, by regularly providing novices triggers to move between different levels of abstraction (Figure 3). First, the novice builds models for solving estimation problems and then they evaluate whether the models satisfy the criteria for obtaining good estimates. Next, the novice synthesizes the role of all three models and obtains a good estimate. When a novice reflects on the goodness of their model for estimation (monitoring), they apply the methods of the domain to evaluate their model. During evaluation, they identify the changes that need to be made to the model in order to meet the goals of estimation and then revise the model (control). There are triggers for monitoring processes which help the novice in synthesizing the role of each model and choosing which model to build next (control) in order to obtain a good estimate. Finally, there are triggers for reflecting and generalizing a process of estimation which novices can apply to new problems.

MEttLE supports learners to create the three models for solving estimation problems, by providing them explicit model-building sub-goals and affordances such as simulations, a causal mapping tool and an equation builder (Figure 4). Learners are also provided guidance regarding expert estimation practices to make comparisons and judgments, choose values and evaluate their estimates. MEttLE has appropriate and regular (after every modelling activity) planning tasks, feedback and expert guidance, along with a process management feature and overall reflection activity. Thus MEttLE was designed to support all four levels of abstraction of model-building, model evaluation, planning and reflection about the practice of estimation (Figure 3).

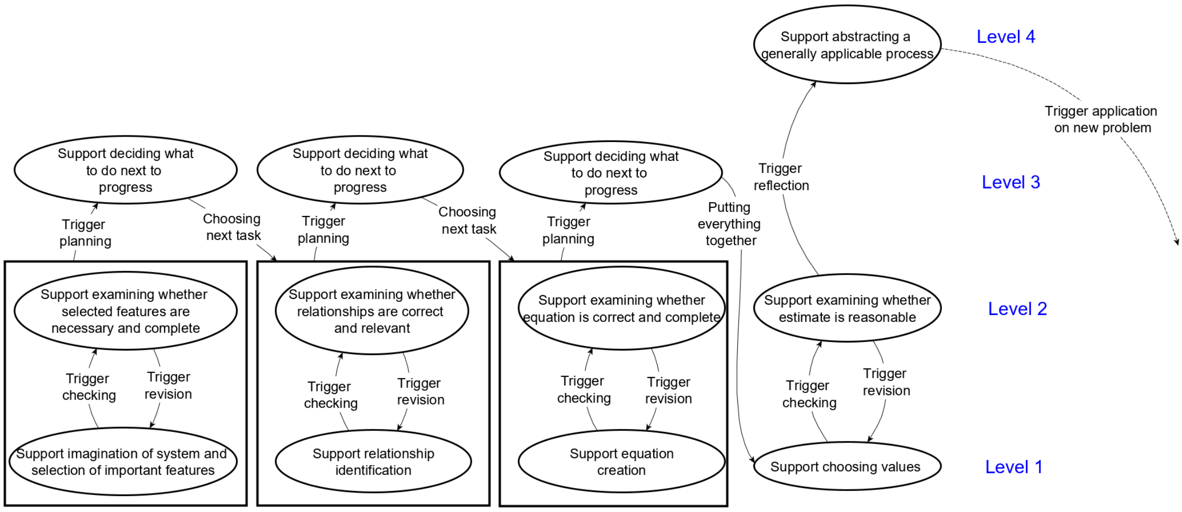

Figure 3: LEAP model for the design of MEttLE

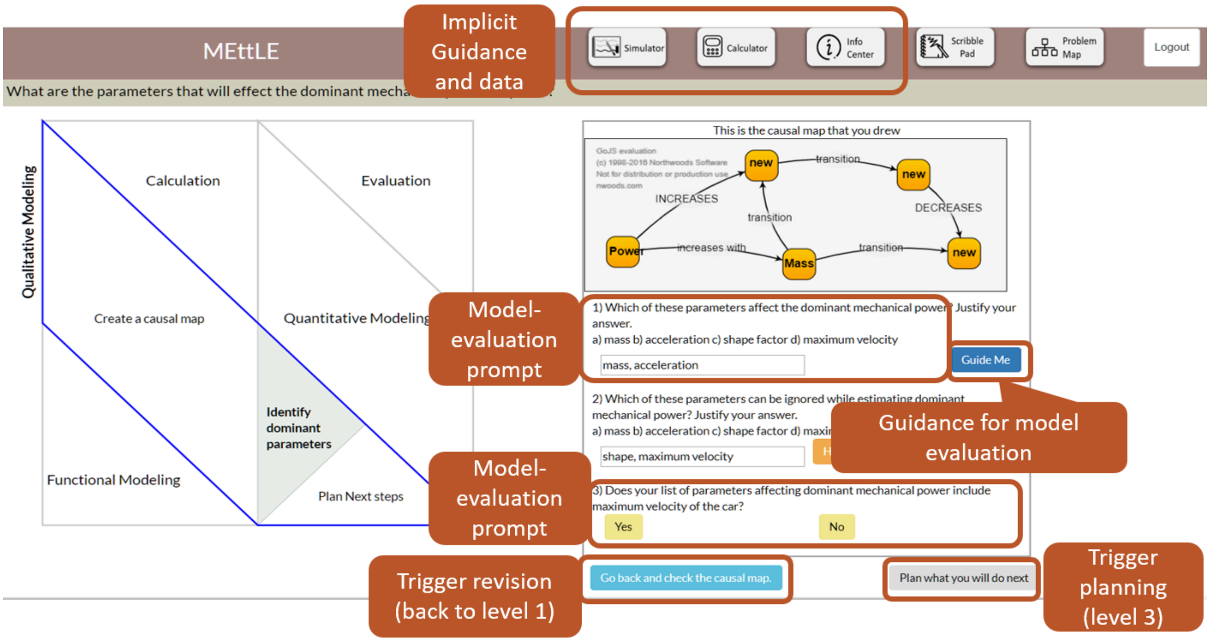

Figure 4: Sample learning activity in MEttLE

We performed two studies with undergraduate engineering students and employed interaction analysis (Jordan & Henderson, 1995) of learners' interaction in MEttLE, along with their reasoning as reported in interviews, to infer their learning process (Kothiyal & Murthy, 2020). We found that when novices obtained good estimates, they used the designed features of MEttLE in different ways to build and evaluate functional, qualitative and quantitative models, and used the planning tasks and guidance in MEttLE to recognize and synthesize the role of all three models in the estimation process, thus attaining the three levels of execute, evaluate and synthesis of the LEAP model. Finally, our results showed that good-performing novices understood the three-phased modelling-based process as a systematic way to solve estimation problems that they can apply to other problems as well, thus attaining the fourth level of generalize as well. Thus we concluded that when learners interact with MEttLE to execute the model-building processes, take action based on the prompts and expert guidance to evaluate and revise their models, reflect on the role of each model and synthesize them, generalize the model-based reasoning process and have the opportunity to practice this process, they go through the four levels of the LEAP model (Figure 5). Further, going through these four levels of abstraction, intertwined with the monitoring and control processes at appropriate times in MettLE, supports novices in developing their estimation expertise by progressive abstraction of the model-based reasoning process.

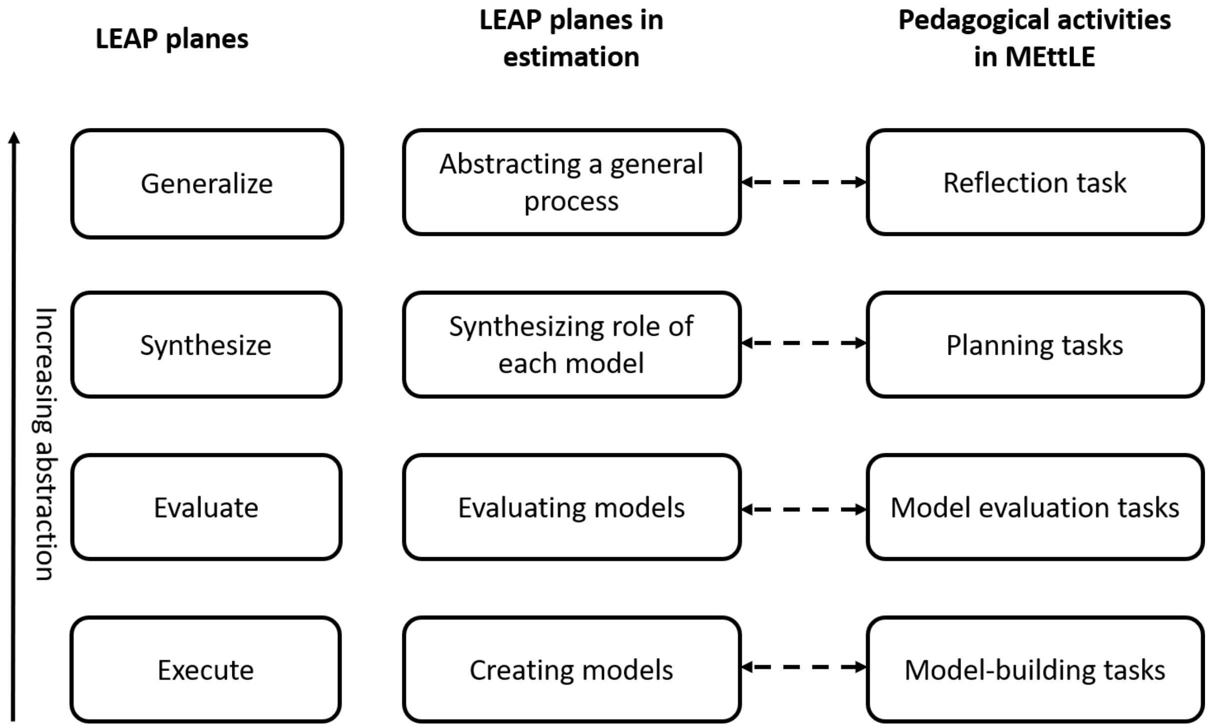

Figure 5: LEAP levels in developing Engineering Estimation expertise

Knowledge Integration (KI)

Novices need to develop the practice of integrating new knowledge with prior knowledge while doing many tasks such as design and problem solving. KI has been defined as, "the process by which learners sort out connections between new and existing ideas to reach a more normative and coherent understanding of science” (Linn and Eylon, 2011). It is recommended that instruction should support at least the following cognitive processes: (i) Eliciting prior knowledge that may be related to the new knowledge; (ii) Focusing on the new knowledge; (iii) Distinguishing ideas - identifying conflicts, inconsistencies, and gaps. A learner with good KI is expected to execute these cognitive processes better than a learner with weaker KI. Supporting the development of KI requires that the learning environment should be able to elicit these cognitive processes in the learners’ mind, make learner reflect on when any of these cognitive processes are needed, what is the role of each of the cognitive processes and how these cognitive processes synthesize together and lead to better knowledge integration, understanding or learning.

To support the development of KI practices among learners, our solution, based on using question-posing (QP) as a cognitive tool, is a TELE called Inquiry-based Knowledge Integration Training (IKnowIT) (Mishra, 2018). IKnowIT employed the LEAP model to help learners move between different levels of abstraction by interleaved execution of cognitive and metacognitive processes of various nature. The pedagogy uses QP-based activities to trigger cognitive and metacognitive processes as per the LEAP model. The first set of activities that a learner does in IKnowIT is to watch a video lecture (~15 minutes long) and type and submit questions in the system as and when they pop up in their mind. The act of posing questions makes the learners implicitly execute the cognitive processes pertinent for questioning, and therefore for knowledge integration. All the processes of QP and KI are executed by the learner without any prior knowledge about them, unknowingly and infrequently. Thus, in this phase, the learner performs "implicit execution" of the cognitive processes of KI.

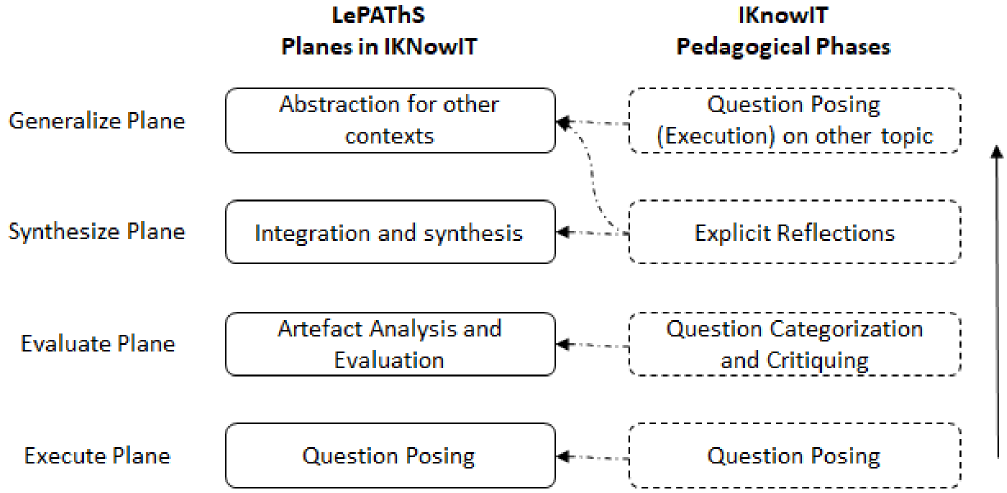

Figure 6. LEAP levels in developing KI

The second set of activities make the learner analyze and evaluate the artefacts emerging out of the previous activity, i.e., questions generated by the learner and her/his peers on the same topic. The learner is supposed to categorize the questions by qualitatively examining and identifying what new knowledge (concepts from the video lecture) and prior knowledge (concepts which were not in the video lecture) (s)he used to generate the questions, and how the prior knowledge and new knowledge were related in the questions. This enables her/him to access, understand and reflect on the roles of cognitive processes of KI that (s)he had implicitly performed during question posing. We can see that in this stage, the learner monitors and evaluates the quality of the artefacts generated. This third set of activities involves the learner to answer a series of reflection questions to ensure that: (1) the learner recognizes the KI thinking processes that (s)he executed; (2) the learner recognizes that these processes were performed while (s)he posed exploratory questions; and (3) the learner recognizes that these processes are essential for their learning. Learners reported that the large role of reflection activity was to help them in "concluding and consolidating their learning about QP." This stage of learning made the learner synthesize the cognitive processes needed for better KI. However, this stage also, partly, made the learner reflect on the overall process by identifying the general processes which (s)he can apply to do KI for other topics (video lectures) in future. The last set of activities make the learner repeat the question posing activity similar to the first set but on a different topic. However, this time, the situation is entirely different for the learner because (s)he has already traversed through the three levels of learning about the cognitive processes of KI. This repetition of the QP activity not only makes the learner execute the cognitive processes again but also makes her/him to evaluate, synthesize and generalize the set of cognitive processes that could be useful for future KI. Several qualitative and quantitative studies (Mishra, 2018) have shown that the IKnowIT successfully leads to the development of KI in learners when they go through four levels of abstraction proposed in the LEAP model, and the monitoring and control processes that take learners between these four levels (Figure 6).

Discussion: Learning engineering practices through abstraction at progressive levels (LEAP): A learning pathway for developing expertise in engineering practices

As described in previous sections, an engineering practice includes a set of cognitive processes that underlie the task, intertwined with a set of monitoring and control processes, applied periodically to meet the goals. In addition, a set of metacognitive processes are needed for abstracting the overall sequence of cognitive processes necessary for doing the practice in novel, but similar situations (Etkina et al, 2010; Litzinger et al 2011; White & Frederiksen, 2005), thus developing expertise in the practice. So, as explained in the theoretical foundations, we extend the two-level cognitive processing model shown in Figure 1 to the four levels in Figure 2. We propose that in order to learn engineering practices, novices must navigate four increasingly abstract levels of cognitive processing (an "object" level plus three "meta" levels). The conceptual foundation of the LEAP model is that novices develop expertise in a practice through the frequent interplay between these levels of cognitive processing via different types of monitoring and control processes. The LEAP model (Figure 7) proposes that novices develop expertise when they are supported, using appropriate scaffolds, to work at all four levels and to frequently move between the levels via the monitoring and control processes. This allows novices to reach the progressively more abstract "meta" cognitive levels and generalize the overall process of doing a practice, and in doing so, develop expertise. The levels of abstraction are *execute, evaluate, synthesize* and *generalize* and novices progress towards more abstract levels via monitoring and towards more concrete levels (i.e., applying the "learned" abstraction to a concrete situation) via control processes. Further, novices spending time at each level of abstraction, leads to growing sophistication in the cognitive processes at that level with each cycle. Below we describe each level and transitions in greater detail.

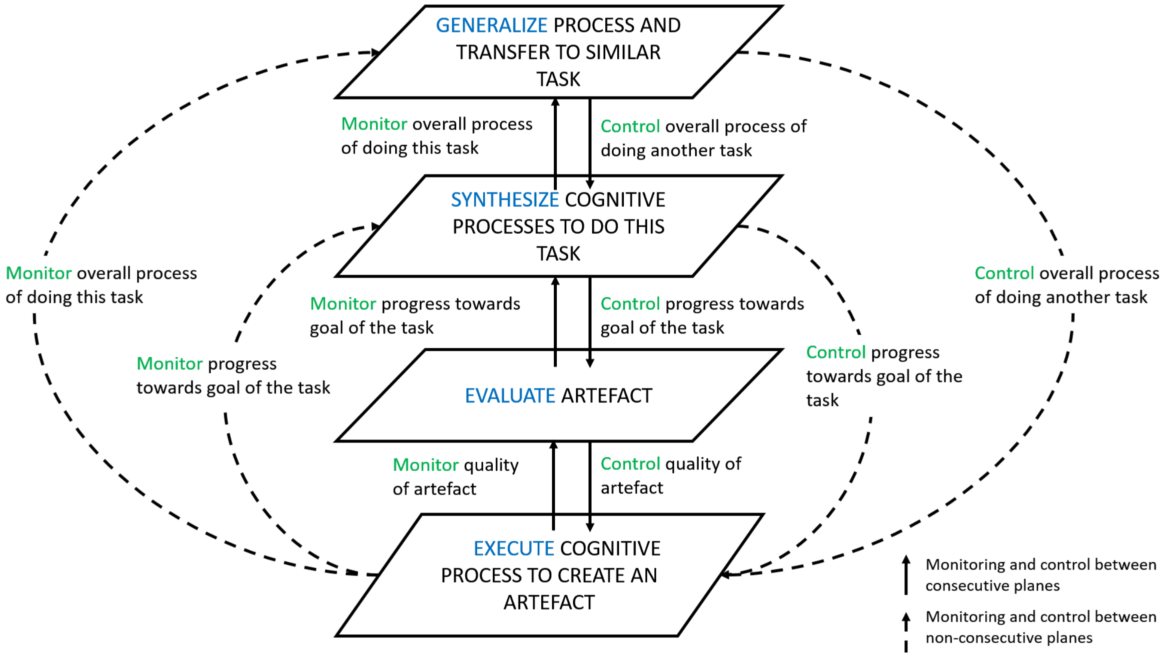

Figure 7: Detailed View of LEAP model

Execute Level

While doing a practice, the expert practitioner applies certain cognitive processes repeatedly to create a series of artefacts which together lead to the goal. For instance, for the practice of conceptual design, designers apply cognitive processes such as semantic retrieval, semantic processing and mental imagery processing to create a design concept (Hay et al, 2017). Further designers repeat these cognitive processes multiple times at various points during the practice in order to create a comprehensive and complete set of design concepts. The LEAP model proposes that novices apply these underlying cognitive processes and create artefacts that lead towards the goal of the task, and that they must do this several times in order to learn the cognitive processes of artefact creation. Learning to do these cognitive processes constitutes the *execute* level of LEAP.

Evaluate Level

At the *evaluate* level, a novice must learn to evaluate the artefacts emerging from the execute level to ensure that they meet the goals of the task. That is, the novice must check whether the created artefacts meet the standard of quality of the discipline and the practice, and whether the artefact will enable moving towards the goal. Novices must repeat this evaluation with every artefact in order to understand the standards of quality and learn the decision process of choosing appropriate artefacts. For instance, in the case of conceptual design, designers assess the concepts against design requirements, constraints and other criteria to assess whether the concepts meet the criteria, and decide which concepts should be taken forward for further development of the design (Hay et al, 2017).

The transition from execution to evaluation is triggered by monitoring the quality of the artefact. In experts, this monitoring is seen in their “inward speech” (Vygotsky, 1978) or “reflection in action” (Schön, 1983) during the task. The results of the evaluation then trigger the processes controlling the quality of the artefact which returns the practitioner to the execute level of doing the cognitive processes again and revising the artefact. For novices, these monitoring and control processes must be intermittently triggered by way of scaffolds which take the novice back and forth between the two levels.

Synthesize Level

Once an artefact satisfies the quality criteria, practitioners decide which artefact to create next in order to move further towards the goal of the task, and hence what cognitive process should be done next. To make this decision, practitioners first synthesize the artefacts created or the cognitive processes done and compare them against the goal of the task. Based on this, they choose the next artefact to be created or the next cognitive process to be done. The LEAP model proposes that at the *synthesize* level, the novice must learn to synthesize the set of cognitive processes applied until a point and choose the next cognitive processes needed to attain the goal. In the example of conceptual design, this would mean combining the design concepts already created and deciding what other concepts are needed to meet the given design requirements and constraints. Accordingly, the designer would create a new design concept.

As before, the process of monitoring progress towards the goal of the task moves the practitioner to the synthesis level. Then they use the result of this synthesis to control which cognitive process to apply next in order to accomplish their goals. Novices have to be frequently triggered to step back and think about what they have accomplished (monitoring) and what they ought to do next to move towards the goal (control).

Generalize Level

At the *generalize* level, novices must reflect on their entire process and identify the sequence of cognitive, and monitoring and control processes which they perceive to be useful to do similar tasks.

The LEAP model recommends that this be done after completing the task and achieving the goal, and novices use their experience of solving a task to identify a general process for the practice which they can apply to do other, similar tasks. In the case of conceptual design, this would mean the designer would examine the sequence of cognitive processes they applied to create the set of design concepts that meet the design requirements and abstract an overall process that they would apply to a novel conceptual design task.

Novices must be triggered to monitor their overall process of doing the task which moves them to the *generalize* level. When presented with a similar task, the novice can then apply this generalized process via a control process of planning the next task or directly choosing a cognitive process to begin the next task.

Monitoring and Control

The novice will need to iterate between all the levels of abstraction via appropriate monitoring and control processes multiple times in order to develop expertise. The number of times this iteration between levels happens depends on the complexity of the practice. This may happen over the course of solving one problem or multiple problems depending on the pedagogy chosen by the instructional designer or teacher. Even though we believe that beginning with the *execute* level and progressing towards more abstract levels consecutively via monitoring their actions might be easier for novices, we do not mandate the order. We do mandate, however, that novices must learn to work at all four planes of abstraction. Novices may, with developing expertise, choose to skip levels as shown in Figure 7 (dotted arrows), i.e., novices may skip explicitly evaluating when they are able to evaluate the artefacts implicitly, and instead directly move to the synthesize level and choose the next cognitive process to do. Also as seen from Figure 7, the monitoring and control processes that novices must apply to move between two consecutive vs non-consecutive levels depends on the ending level of abstraction. What is important for developing expertise is that novices spend sufficient time working and practising the cognitive processes at each level of abstraction and apply the monitoring and control processes to move between all four levels.

Implications and Conclusions

In this paper, we present a model that describes a learning pathway for how practices in science and engineering can be developed among novices. The main features of the model are practising cognitive processes at progressively higher levels of abstraction namely execute, evaluate, synthesize and generalize, interleaved with the metacognitive processes of monitoring and control to periodically move between the levels. In studies with learners working on TELEs for two practices of engineering estimation and knowledge integration, we found that the pedagogical design of the TELEs triggered learners to go through the four levels of abstraction. When learners go through cognitive processes at these four levels, interspersed with monitoring and control processes, they develop the respective practices.

The LEAP model can be used by researchers studying learners' development of practices in science and engineering, as the model provides a lens through which to trace the pathway among learners. Models of student thinking have been conceptualized in terms of learning progressions, ie, characterizing successively sophisticated ways of reasoning about a topic as students learn (National Research Council, 2012). More recent research on learning progressions have focused on describing how students gain expertise in inquiry science practices (Schwarz et al, 2009). In a similar vein, the LEAP model can be used to analyze novice learners' performance in their learning environments: they can seek evidence for learners working at the four levels (execute, evaluate, synthesize and generalize) of abstraction, and evidence of learners moving across the levels by monitoring and controlling their cognitive processing. They can then track how this interplay between the four levels of cognitive processing via frequent monitoring and control leads to the development of the thinking skill such that novices are able to apply the skill to new tasks.

The LEAP model can be applied by teachers and instructional designers who want to design effective TELEs for developing learners' practices in science and engineering. Existing frameworks provide guidelines to design learning activities for specific practices such as scientific inquiry (Quintana et al 2004), integrating computational thinking into a science curriculum (Sengupta et al 2013), and solving engineering problems (Woods et al 1997), and provide scaffolds for specific purposes such as reflection prompts for ill structured problem solving (Ge & Land, 2004). The LEAP model complements the above frameworks by providing a model for student learning of practices, which can be used as the guiding structure to make decisions about the pedagogical design. In order to begin the pedagogical design, one would have to analyze the constituent cognitive and metacognitive processes of the practices, and what needs to be monitored and controlled at each of the four levels of abstraction. This can be done through literature analysis as well as studying experts to identify the cognitive and metacognitive processes. The next step is to design appropriate learning activities and scaffolds that will allow learners to learn the component processes and the overall process of the thinking skill, which can be drawn from the above frameworks and guidelines. Finally, the LEAP model suggests the types of learning activities in each level that align with the requirements of that level. For example, the activity at the execute level should require the learner to apply the corresponding cognitive process to create an artefact (such as building a model via a causal map), while the evaluate level activity should be a sense-making activity or a self-assessment task (such as, evaluating if the causal map includes all relevant parameters).

Some examples of practices where the LEAP model has been applied include science inquiry practices, engineering design thinking, engineering estimation and troubleshooting. We expect that the LEAP model can be applied to other practices such as computational thinking, however the scope of its application is limited to practices relevant to science and engineering. The LEAP model may be applicable to broader related practices such as ill-structured problem solving and critical thinking, however this needs to be examined through further research.

It is known that volition and affective factors such as engagement and interest play an important role in novices' perseverance in a task (Presseisen, 1991) and their metacognitive awareness (McWhaw & Abrami, 2001). In our model, however, we delimit ourselves to a learning mechanism focusing on the cognitive and metacognitive aspects of practices and assume that the novice has high volition, interest and engagement towards the task so that these factors do not affect their persistence or metacognitive awareness as they work on the task.

While the LEAP model details out the interplay between cognition and metacognition, a key limitation is that it does not include other factors such as learners' personal epistemologies and epistemic cognition (Kuhn & Weinstock, 2002), affect and conation, that are known to play a role in learning. The interaction between some of these factors have been well studied, such as, cognitive-affective models of learning that include conceptual change (Pintrich, Marx & Boyle, 1993) or a meta-review on interest as a predictor for learning (Schiefele, Krapp & Winteler, 1992). The LEAP model in its current form is delimited to cognitive and metacognitive aspects. Another limitation is that the LEAP model considers only individual processes in the learning pathway. The effects of collaboration and social processes in learning, and mechanisms of creation of shared understanding have been part of many theories (for example, Stahl, 2006). Including these factors in a learning model will help in both understanding learners' development of practices as well as designing effective learning environments to support learners'.

The LEAP model offers instructional designers a way to think about creating learning activities and scaffolds, based on learners' progression between concrete and abstract levels. In order to facilitate the instructional design process, specific guidelines are needed on how to choose and design learning activities that can trigger the cognitive and metacognitive processes on each level and transition of the LEAP model. In order to create such guidelines, future work includes designing, developing and evaluating learning environments to understand if these processes are indeed being triggered, if the progression between levels is being seen, and if different practices are being learnt. This reflective iteration between design, development and evaluation will lead to design guidelines for learning activities that can support learning of practices via the LEAP model.

Funding Information

This research was partially supported by a grant from Next Education Research Lab at Educational Technology, IIT Bombay

Conflict of Interest

The authors declare that they have no conflict of interest

References:

- ABET (2014). Criteria for Accrediting Engineering Programs http://www.abet.org/eac-criteria-2014-2015/
- Adams, J. P., Kaczmarczyk, S., Picton, P., & Demian, P. (2008). Problem Solving and Creativity in Engineering: findings of interviews with experts and novices. In International Conference on Engineering Education ICEE 2008.
- Atman, C. J., Adams, R. S., Cardella, M. E., Turns, J., Mosborg, S., & Saleem, J. (2007). Engineering design processes: A comparison of students and expert practitioners. Journal of engineering education, 96(4), 359-379.
- Ball, L. J., St. B. T. Evans, J., Dennis, I., & Ormerod, T. C. (1997). Problem-solving Strategies and Expertise in Engineering Design. Thinking & Reasoning, 3(4), 247–270.
- Ball, L. J., Ormerod, T. C., & Morley, N. J. (2004). Spontaneous Analogising in Engineering Design : A Comparative Analysis of Experts and Novices. Design Studies, 25, 495–508.
- Ball, L. J., & Christensen, B. T. (2009). Analogical reasoning and mental simulation in design: two strategies linked to uncertainty resolution. Design Studies, 30(2), 169–186.
- Benjamin, C., & Keenan, C. (2006). Implications of introducing problem-based learning in a traditionally taught course. Engineering education, 1(1), 2-7.
- Božić, M., Čizmić, S., Pavlović, D. Š., & Tramullas, M. T. E. (2014, April). Engineering practice: teaching ill-structured problem solving in an internship-like course. In 2014 IEEE Global Engineering Education Conference (EDUCON) (pp. 721-726). IEEE.
- Chiu, J. L., Malcolm, P. T., Hecht, D., Dejaegher, C. J., Pan, E. A., Bradley, M., & Burghardt, M. D. (2013). WISEngineering: Supporting precollege engineering design and mathematical understanding. Computers and Education, 67, 142–155.
- Crismond, D. P., & Adams, R. S. (2012). The Informed Design Teaching and Learning Matrix. Journal of Engineering Education, 101(4), 738–797.

- Crismond, D. P., Howland, J., & Jonassen, D. (2011). Designing with technology. Meaningful learning with technology, 72-90.
- Daly, S. R., Christian, J. L., Yilmaz, S., Seifert, C. M., & Gonzalez, R. (2011, June). Teaching Design Ideation. In 2011 ASEE Annual Conference & Exposition (pp. 22-1382).
- Davidson, J. E., Deuser, R., & Sternberg, R. J. (1994). The role of metacognition in problem solving. Metacognition: Knowing about knowing, 207-226.
- De Graaf, E., & Kolmos, A. (2003). Characteristics of problem-based learning. International journal of engineering education, 19(5), 657-662.
- Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. Journal of Engineering Education, 94(1), 103–120.
- Dutson, A. J., Todd, R. H., Magleby, S. P., & Sorensen, C. D. (1997). A review of literature on teaching engineering design through project‐oriented capstone courses. Journal of Engineering Education, 86(1), 17-28.
- Efklides, A. (2008). Metacognition: Defining Its Facets and Levels of Functioning in Relation to Self-Regulation and Co-regulation. European Psychologist, 13(4), 277–287.
- Efklides, A. (2009). The role of metacognitive experiences in the learning process. Psicothema, 21(1), 76–82.
- Etkina, E., Karelina, A., Ruibal-Villasenor, M., Rosengrant, D., Jordan, R., & Hmelo-Silver, C. E. (2010). Design and reflection help students develop scientific abilities: Learning in introductory physics laboratories. The Journal of the Learning Sciences, 19(1), 54-98.
- Flavell, J. H. (1979). Metacognition and Cognitive Monitoring: A New Area of Cognitive-Development Inquiry. American Psychologist, 34(10), 906–911.
- Fox, E., & Riconscente, M. (2008). Metacognition and Self-Regulation in James, Piaget, and Vygotsky. Educational Psychology Review, 20, 373–389.
- Froyd, J. E., Wankat, P. C., & Smith, K. A. (2012). Five major shifts in 100 years of engineering education. Proceedings of the IEEE, 100(Special Centennial Issue), 1344-1360.
- Ge, X., &Land, S. M. (2004). A Conceptual Framework for Scaffolding Ill-Structured Problem-Solving Processes Using Question Prompts and Peer Interactions. Educational Technology Research and Development, 52(2), 5–22.
- Hay, L., Duffy, A. H., McTeague, C., Pidgeon, L. M., Vuletic, T., & Grealy, M. (2017). Towards a shared ontology: A generic classification of cognitive processes in conceptual design. Design Science, 3.
- Holton, D., & Clarke, D. (2006). Scaffolding and metacognition. International Journal of Mathematical Education in Science and Technology, 37(2), 127–143.
- Kalnins, S. N., Valtere, S., Gusca, J., Valters, K., Kass, K., & Blumberga, D. (2014). Cooperative problem-based learning approach in environmental engineering studies. Agronomy Research, 12(2), 663-672.
- Kenkre, A. B., & Murthy, S. (2017). MIC-O-MAP: a technology-enhanced learning environment for developing micro-macro thinking skills. Research and Practice in Technology Enhanced Learning, 12(1), 23.
- Kirschner, P. A., & Merriënboer, J. Van. (2008). Ten Steps to Complex Learning A New Approach to Instruction and Instructional Design. In T. L. Good (Ed.), 21st Century Education: A Reference Handbook, 244–253.
- Kothiyal, A., Murthy, S., & Chandrasekharan, S. (2016). Hearts pump and hearts beat: Engineering estimation as a form of model-based reasoning. In Proceedings of the 12th International Conference of the Learning Sciences (pp. 242-249). Singapore: International Society of the Learning Sciences
- Kothiyal, A., & Murthy, S. (2018). MEttLE: a modelling-based learning environment for undergraduate engineering estimation problem solving. Research and Practice in Technology Enhanced Learning, 13 (1), 17.
- Kothiyal, A. (2019). A model to understand and scaffold novices in estimation problem solving using a technology-enhanced learning environment. PhD Dissertation, IIT Bombay.

- Kothiyal, A. & Murthy, S. (2020). Disciplinary Model-Based Reasoning and Metacognition Underlies Good Estimation Performance by Engineering Undergraduates. In Gresalfi, M. and Horn, I. S. (Eds.), The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020, Volume 2 (pp. 1095-1102). Nashville, Tennessee: International Society of the Learning Sciences.
- Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., & Ryan, M. (2004). Promoting deep science learning through case-based reasoning: Rituals and practices in learning by design classrooms. Curriculum, plans and processes of instructional design: International perspectives, 89-114.
- Kuhn, D., & Weinstock, M. (2012). What is epistemological thinking and why does it matter?. In Personal epistemology (pp. 121-144). Routledge.
- Johri, A., & Olds, B. M. (2011). Situated engineering learning: Bridging engineering education research and the learning sciences. Journal of Engineering Education, 100(1), 151-185.
- Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice. The Journal of the learning sciences, 4(1), 39–103.
- Lipman, M. (2003). Thinking in Education (Second). Cambridge University Press.
- Linder, B. M. (1999). Understanding estimation and its relation to engineering education, (Unpublished doctoral dissertation). Massachusetts Institute of Technology.
- Linn, M. C. & Eylon, B.-S. (2011). Science learning and instruction: Taking advantage of technology to promote knowledge integration. Routledge.
- Litzinger, T. A., Lattuca, L. R., Hadgraft, R. G., & Newstetter, W. C. (2011). Engineering Education and the Development of expertise. Journal of Engineering Education, 100(1), 123–150.
- Mandinach, E. B., & Cline, H. F. (1996). Classroom dynamics: The impact of a technology-based curriculum innovation on teaching and learning. Journal of Educational Computing Research, 14(1), 83-102.
- Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. Instructional Science, 26, 49–63.
- McWhaw, K., & Abrami, P. C. (2001). Student goal orientation and interest: Effects on students' use of self-regulated learning strategies. Contemporary educational psychology, 26(3), 311-329.
- Mishra, S. (2018). Fostering Cognitive Processes of Knowledge Integration through Exploratory Question-Posing. PhD Dissertation, IIT Bombay.
- Moore, T. J., Miller, R. L., Lesh, R. A., Stohlmann, M. S., & Rae, Y. (2013). Modeling in Engineering: The Role of Representational Fluency in Students' Conceptual Understanding. Journal of Engineering Education, 102(1), 141–178.
- Moore, T. J., Glancy, A. W., Tank, K. M., Kersten, J. A., Smith, K. A., & Stohlmann, M. S. (2014). A framework for quality K-12 engineering education: Research and development. Journal of pre-college engineering education research (J-PEER), 4(1), 2.
- Murthy, S., Iyer, S., & Mavinkurve, M. (2016). Pedagogical framework for developing thinking skills using smart learning environments. Learning, Design, and Technology: An International Compendium of Theory, Research, Practice, and Policy, 1-49.
- Nachtmann, H., & Lehrman, A. (2002). Teaching Engineers to Handle Uncertainty. In Proceedings of the 2002 ASEE Annual Conference and Exposition.
- National Research Council. (2012). A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. National Academies Press.
- Nelson, T. O., & Narens, L. (1994). Why investigate metacognition. Metacognition: Knowing about knowing, 1-25.
- Pahl, G., & Beitz, W. (2013). Engineering design: a systematic approach. Springer Science & Business Media.
- Perrenet, J. C., Bouhuijs, P. A., & Smits, J. G. (2000). The suitability of problem-based learning for engineering education: theory and practice. Teaching in higher education, 5(3), 345-358.
- Pimmel, R. (2001). Cooperative learning instructional activities in a capstone design course. Journal of Engineering Education, 90(3), 413-421.

- Pintrich, P. R., Marx, R. W., & Boyle, R. A. (1993). Beyond cold conceptual change: The role of motivational beliefs and classroom contextual factors in the process of conceptual change. Review of Educational research, 63(2), 167-199.
- Presseisen, B. Z. (1991). Thinking skills: Meanings and models revisited. Developing minds: A resource book for teaching thinking, 56-62.
- Quintana, C., Reiser, B. J., & Davis, E. A. (2004). A scaffolding design framework for software to support science inquiry. The Journal of Learning Sciences, 13(3), 337–386.
- Rivera-Reyes, P., & Boyles, R. E. (2013). Training in Troubleshooting Problem-Solving: Preparing Undergraduate Engineering Students for Industry. 120th ASEE Annual Conference and Exposition, (May 2000).
- Ross, C., & Orr, R. R. (2009). Teaching structured troubleshooting: integrating a standard methodology into an information technology program. Educational Technology Research and Development, 57(2), 251-265.
- Rugarcia, A., Felder, R. M., Woods, D. R., & Stice, J. E. (2000). The future of engineering education I. A vision for a new century. Chemical Engineering Education, 34(1), 16-25.
- Schön, A. D. (1983). The Reflective Practitioner: How professionals think in action. Basic Books
- Schön, D. A. (1987). Educating the reflective practitioner.
- Schwarz, C. V., Reiser, B. J., Davis, E. A., Kenyon, L., Achér, A., Fortus, D., ... & Krajcik, J. (2009). Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learners. Journal of research in science teaching, 46(6), 632-654.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. Education and Information Technologies, 18(2), 351-380.
- Schiefele, U., Krapp, A., & Winteler, A. (1992). Interest as a predictor of academic achievement: A meta-analysis of research.
- Sheppard, S., & Jennison, R. (1997). Freshman engineering design experiences and organizational framework. International Journal of Engineering Education, 13, 190-197.
- Sheppard, S., Colby, A., Macatangay, K., & Sullivan, W. (2006). What is Engineering Practice?*. International Journal of Engineering Education, 22(3), 429–438.
- Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences (pp. 409-426). Cambridge, UK:
- Stojcevski, A., & Fitrio, D. (2008, December). Project based learning curriculum in microelectronics Engineering. In 2008 14th IEEE International Conference on Parallel and Distributed Systems (pp. 773-777). IEEE.
- Vincenti, W. G. (1990). What engineers know and how they know it (Vol. 141). Baltimore: Johns Hopkins University Press.
- Vincenti, W. G. (2000). Engineering Experiment and Engineering Theory: The Aerodynamics of Wings at Supersonic Speeds, 1946–1948. In Atmospheric Flight in the Twentieth Century (pp. 157-180). Dordrecht: Springer Netherlands.
- Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Harvard University Press.
- Wankat, P. C., & Oreovicz, F. S. (Eds.). (2015). Teaching engineering. Purdue University Press.
- White, B. Y., & Frederiksen, J. R. (2005). A theoretical framework and approach for fostering metacognitive development. Educational Psychologist, 40(4), 211–223.
- Woods, D. R., Hrymak, A. N., Marshall, R. R., Wood, P. E., Crowe, C. M., Hoffman, T. W., ... & Bouchard, C. K. (1997). Developing problem solving skills: The McMaster problem solving program. Journal of Engineering Education, 86(2), 75-91.
- Woods, D. R., Felder, R. M., Rugarcia, A., & Stice, J. E. (2000). The future of engineering education III. Developing critical skills. Chemical Engineering Education, 34 (2), 108-117.
- Woods, D. R. (2000). An Evidence-Based Strategy for Problem Solving. Journal of Engineering Education, 89(4), 443–459.