Applying the Adaptive Control of Thought-Rational Theory into the Design of Mobile Worked Examples Applications

Ilker Yengin, Ibrahim Furkan Ince

Abstract
Using a direct problem solving method creates confusion and cognitive load for the novice learners. To address the issue, we applied basics of “Adaptive Control of Thought-Rational Theory (ACT-R) and designed a mobile learning application that uses worked examples using simulation. As an alternative method, this mobile application supports teaching production rules and procedural knowledge in a step-by-step manner. This paper discusses the basics implementations of ACT-R Theory into the design of mobile application that uses worked examples as the main delivery method. The paper would be interesting for educational technology practitioners that would like to have examples of the practical applications of ACT-R Theory into mobile learning application design.

Keywords: Adaptive Control of Thought-Rational Theory (ACT-R), Simulations, Worked Examples, Mobile Learning, Mobile Application Design

1. Introduction

According to Adaptive Control of Thought-Rational (ACT-R) Theory of Anderson [1], in the initial levels of problem solving process, students need examples and analogies that presents solutions to a problem where students can use declarative knowledge (the examples and the presented solution steps itself) and can create production rules. Same theory implies that using declarative knowledge and creating production rules enable learners to gain procedural knowledge that can be used later (because of many practices) without a need of any reference examples.

ACT – R theory mostly interested in rule related learning events that are based on procedural knowledge building such as problem-solving tasks. ACT-R’s relationship with problem solving skills has been investigated widely in many different research studies [2-5]. Shorty, ACT-R theory asserts that the declarative knowledge is a direct encoding of the environment and the procedural knowledge is the direct encoding of observed and synthesized transformations of the environment [6], [7].

ACT – R theory borrows ideas from human associative memory (HAM) theory [8]. Similar to HAM, ACT-R explains mechanisms of human cognition to make a clear distinction between declarative and procedural knowledge by showing how procedural knowledge is derived from declarative knowledge [9], [10]. Accordingly, when students are presented into the situations where there is a heavy need of applications of learned knowledge bits in the lesson, there should also be production rules for further knowledge building. The productions rules are integrated in a procedural form of knowledge. Procedural form of knowledge has to be built due to the occurrence of declarative knowledge in students’ Long Term Memories (LTM) [11]. Another form of presenting the heavy knowledge is to present it in the form of declarative knowledge. Declarative knowledge is encoded as forms and functions in schema like structures which are named as PUPS structures [6]. ACT-R uses production rules, which shape the procedural knowledge. A specific production rule can be applied into situations whenever there are conditions matched for designated declarative knowledge in LTM. Also a production rule can result into actions that create new declarative knowledge [7].

Production rules originate from the encoding of chunks which are visual mappings of environment due to the synthesis of recognized objects [7]. Once an object is synthesized, it is ready for chunks in the working memory [7],[12]. Production rules indicate the transformations of chunks encoded from environment such as the examples.

When a problem presented to a student, the first thing she must do is to recognize a mapping between given structures of the problem. According to ACT-R, student should directly map the identical structures. After the mapping, she will search for a chunk structure that makes the mapping. After finishing the mapping, she will form a production rule to connect structures to each other.
Anderson asserts, “This is how students learn problem-solving by mimicking examples of solutions” [7]. When students face a problem, they tend to try eliminating the difference between the current state and the goal state of the problem. This problem-solving method is called as “means ends analysis”. When students employ techniques for eliminating and reducing the difference, they are subtotaling the operators. When students get a situation where there is no problem-solving operator, they will search for an identical problem-solving state and try to solve the problem using this identical state as an example [13]. There is a serious issue in this approach of problem solving, especially for novice students. Such analogical reasoning could create a cognitive load to compute mapping and it would work only if there are example steps [14-16].

According to results of initial research studies, the use of examples as a guide affects the earlier stage of problem solving of students [17]. In the earlier stage of problem-solving, students need to recall examples and analyze which also require getting declarative knowledge from LTM [18-22]. In the analyses and interpretation stage, students encode the rules of problem-solving procedurally [1]. This stage of moving from interpretive stage to procedural stage is called as “knowledge compilation” [1]. The knowledge compilation analyzes the core of the analogical example and generates a production rule, which also will eliminate the blind search for a solution that would create a great amount of cognitive load [1]. It is very important that students create some knowledge of procedures (or steps of procedures) by encoding how the operators achieve their successful function [23].

According to ACT-R theory, when we present examples to students, rather than presenting situations requiring a means end analysis using generalization and discrimination; we should present the critical features allowing construction of the production rules that results in knowledge compilation. As Anderson [12] adds; “this is not to argue that examples are not important, but they should be annotated with information about what they are supposed to illustrate”. Thus, it is clear that we need another methodology that illustrates the procedures when we present examples to the students. Worked examples are one candidate for such a purpose.

### 1.1 Worked Examples and ACT-R

A worked example is a step-by-step problem solving methodology that is designed to teach students how to approach models of problem solving in steps [24]. Worked examples are very effective teaching methodology in illustrating solution steps of problems, epically when the students have no prior knowledge about the problem solution [25]. Application of worked examples may vary. For instance, they can be like sequential steps that might demonstrate the results and solutions of problems in diagrams or other visual forms [26]. Another use of worked examples is to illustrate a pattern or principle such as components of equations [27]. Relatedly, literature on the worked examples basically discusses the use of the methodology in domain areas (such as chess, athletics, mathematics, music or programming etc.) where learners are required to gain cognitive skills in steps [28-31].

The use of analogy, declarative rules, guidelines and steps leading to solutions can be used as templates to guide students to novel solutions [21]. When students apply repeated practices related to the examples, general rules are formed and the specific examples and analogies tied to these rules [1]. Accordingly, the analogies may not be needed to access for reference. This means that knowledge transitions from declarative knowledge are based on encoding of examples to a procedural knowledge. After this transitions process the production rules are formed [1].

If we present a solution in an example form, which results in a mapped production, then knowledge compilation will form and the new solution can be applied later without a need of recall of the example [32]. In this sense of usage of worked examples, the problem solving is much related to the ACT-R theory [33].

Worked examples are one of the suitable methods in order to activate the mechanisms which enable students to create production rules [34-37]. In teaching production rules, worked examples are also superior to the minimal guidance based problem solving method because worked examples create less cognitive load [38]. Thus, worked examples are one of the critical alternative methods for teaching production rules.

Using worked examples as a teaching method has been applied into technology supported learning scenarios such as in inquiry based mobile learning systems [39], intelligent tutoring software application [40-42] and multimedia applications [43].
Although, worked examples are used in the design of few numbers of digital learning applications in research studies, there is no study discussing the application of the ACT-R theory as a guide into the design process of the worked examples in mobile learning applications. Studying the design of mobile applications that uses worked examples within the light of ACT-R theory may show us new insights. To address this need and to explain the use of worked examples on mobile application design within the light of ACT-R theory, we designed a mobile application that uses worked examples as the main teaching method. The mobile application is expected to help students to understand the production rules.

In the design background section, the conceptual model of this application is explained with its connections to the ACT-R Theory. Following that, in the application design section, the design process is illustrated with detailed examples.

2. Design Background

Using the literature above and applying the fundamentals of the ACT-R Theory, we have compiled a design specification list in Table 1. This list enables us to theoretically understand and conceptualize the design of the mobile learning application. The list clearly connects the theory with the proposed application design.

<table>
<thead>
<tr>
<th>Design Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Students should be provided with learning events that enable mimicking examples of solutions. (These events would help students to have problem situations that enable recognition of identical structures and mapping. After the mapping, students should also form a production rule to connect the mapped structures to each other.)</td>
</tr>
<tr>
<td>2. Students should avoid complex tasks analysis</td>
</tr>
<tr>
<td>3. The application should present the critical features in the examples.</td>
</tr>
<tr>
<td>4. The application should support creation of production rules.</td>
</tr>
<tr>
<td>5. Production rules should be compiled from declarative knowledge chunks (Declarative knowledge bits could be presented as different parameters and could be connected and compiled as a production rule)</td>
</tr>
</tbody>
</table>

3. Application Design

The application is intended to be a complementary learning aid where students can practice anytime both in and out of the classroom. Thus, it is designed and developed as a mobile application running on iOS and Android platforms. Table 1 illustrates the basic principles to be applied in our mobile learning application design. Accordingly, we designed and developed the application especially for novice learners where they can use and practice their knowledge to build production rules. The application architecture consists of three main parts, which are worked example based simulations module, free form simulations module and profiling module.

Worked examples based simulation module provides examples that enable students to form and map production rules from declarative chunk bits. Critical features of problem solving procedures will be given in form of sequential steps in worked examples. Moreover, this module may help student to have less cognitive load by not presenting problems directly. Direct presentation of the problems may require the students to engage complex analysis before they master all the problem solving procedures. Rather than requiring students to solve problems directly, the worked examples based simulations module will create a platform for students to practice their knowledge by following the examples of sequenced steps that explains the relationships of parameters in the problem. In the simulations, students are presented pre-fixed parameters. The relationships between these parameters are explained in a worked example fashion. Mainly, students are presented a problem where they need to know the rules based on procedural knowledge that is structured according to the relationships of different parameters affecting the event. The worked examples present the step-by-step demonstrations for the relationships between these parameters. For example, in the Figure 1, students are given a problem related to the Kinematics and Inclined Motion topics.
In Figure 1, students are not presented a direct problem question related to the topic or not presented any example that does not provide an opportunity for the students to form productions rules. Instead, the application auto plays and changes the parameters step-by-step to demonstrate the effect of different parameters on inclined motion. After that, the application guide students to form a production rule related to the inclined motion demonstrated by the simulation (Figure 2).
Following the auto played examples in Figure 1; students are asked to form the formula for an accelerating object on an incline in Figure 2. Dragging the parameters and operators on the equation box, the students may form their equations. Once they finish constructing the equation, the application gives feedback about their performance. After that, the application provides a step-by-step solution of problems. Problems are given as sets to enable students to well internalize the production rules as well as enabling them to recognize the connections between the steps. At the beginning of the each problem sets, the worked examples demonstrate all the steps leading to the solutions for similar problems.

As the students successfully continue to solve similar steps in similar problems, the application start taking away some of the steps. This encourages students to complete the missing steps by themselves. This procedure is called as a faded worked examples method. In the final stages of problem sets, students are provided just problems instead of worked examples in the free form of simulations modules (Figure 3).

![Application provides step-by-step worked examples.](image)

Free from of simulations modules provide a platform where students uses different specifications to understand the relationships between different declarative knowledge bits. This module is presented after the students master the worked example based problem sets. In this module, students are not provided any solution steps so they are expected to solve and understand the problems by themselves using the production rules they have learned in the worked examples based module. This module is called as free form, because it lets students to discover the problem situations freely without a pre given worked example. By changing the different specifications effecting on the problem situations, students may observe the changing effects of different declarative knowledge such as processes, parameters and their attributes, and the relations between these processes, parameters and their attributes. Free form of simulation is similar to the example in Figure 1, except instead of the application auto plays and changes the parameters step-by step; students are free to change the parameters by themselves to see the different effects. For self-reminding purposes, they can still activate the auto play option. Profiling module works with user students’ interaction log data to model user learning path. Learning path is calculated according to the students’ each successful completion of the worked examples. The learning path starts with student’s initial start reference point and ends with predetermined mastery point. The mastery levels means students are able to solve all the steps in the worked examples problem sets. Profiling module is just informative to students to help them to figure out their ongoing progress.
4. Discussions

Using the literature about the ACT-R Theory, we compiled a design specification list (Table 1) and developed a mobile application as described in the previous paragraph. According to the ACT-R theory, worked examples should help students to form production rules and reduce cognitive load. We designed the application for the mobile platforms to enable students to use it in and out of the classroom as a learning aid. For instance students may use the application in the physics laboratory session to practice the worked examples.

In a future study, the application may be tested within experimental user studies to evaluate its ability to reduce cognitive load and to help students to learn the concepts by forming production rules as the ACT-R theory suggested.

5. Conclusion

In this paper, we present example design for an alternative mobile learning application that is using worked examples. We applied ideas from Adaptive Control of Thought-Rational Theory (ACT-R) for designing worked examples to implement our mobile application. ACT-R theory guided our mobile learning application design where it enables us to theoretically understand and conceptualize the further development of the mobile learning application. Accordingly, we used worked examples based simulations to provide examples that enable students to form and map production rules from declarative chunk bits. Worked examples based simulations module is expected to create a platform for students to practice their knowledge by following the examples of sequenced steps which explains the relationships of parameters in the problem.

Relying on simulation oriented worked examples; our model is different than other mobile learning applications that rely on direct problem solving which potentially creates confusion and cognitive load for the novice students.

6. References


