HIROSHIMA UNIVERSITY

Retaining the Information Structure in the Open Information Structure Approach During Redesign of Learning Applications: Two Study Cases (オープン情報構造アプローチを用いた学習アプリケーションの再設計における情報構造の保持：2 つの事例)

by

Pedro Gabriel Fonteles Furtado

A thesis submitted in partial fulfillment for the
Doctor of Engineering degree

in the
Graduate School of Engineering
Department of Information Engineering

2020 September

Declaration of Authorship

I, Pedro Gabriel Fonteles Furtado, declare that this thesis titled, 'Retaining the Information Structure in the Open Information Structure Approach During Redesign of Learning Applications: Two Study Cases' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Yori ii mono ni suru tame, jinsei mo manga mo onaji. Jibun nara yarerutte unubore ya un mo hitsuyou dakedo, ichiban taisetsu na no ha, doryoku."

Tsugumi Ohba

HIROSHIMA UNIVERSITY

Abstract

Graduate School of Engineering
Department of Information Engineering

Doctor of Engineering

by Pedro Gabriel Fonteles Furtado

The open information structure is a promising approach to design learning applications. One challenge with learning applications is redesigning the activity for new learning contexts. The open information structure, however, is based on interactable information structures. There is a possibility that the activity can be redesigned for the new contexts without changing the underlying information structure. This should make for a more effective redesign process. This study explores redesigning learning applications that use the open information structure approach. It explores two case studies of two systems that were redesigned and analyses data from multiple experiments. The redesign process was successful and the applications appropriate for their new context. This study also provides insight on how it has affected learning gains and on how the redesigned portions affect how the application should be used.

Acknowledgements

Gratitude to my family, for their love and support. Without them, I would not have came so far.

Gratitude to my two brilliant professors, Professor Hirashima and Professor Hayashi, for their guidance.

Professor Hirashima was the one who made me want to come to Hiroshima University. He was also the one who made me want to continue working in research. A genius of discussions, who can consider the opinions of others better than I ever could, always trying to find an agreement and deepen everyone's knowledge. He may not believe me, but I'm definitely a fan. He often helped me think about things deeper and from different angles. I'm truly thankful.

Professor Hayashi offered me needed insight and advice into my research, helping me find and deal with various problems. He also helped me adapt to the various cultural differences between Japan and my home country, an adaptation which is still a work in progress. Our various conversations have helped me mature better as a person and for that I'm grateful.

Deep thanks to the Japan's Ministry of Education, Culture, Sports, Science and Technology(MEXT), for having chosen me to receive a scholarship. If not for their continued support, I would not have had the chance to come to Japan. I hope I can contribute further to my laboratory and to Japan, as to honor their choice.

Of course, all my lab-mates. Always warm, friendly, accepting and glad to help. In particular Yamamoto, Hirota, Yoshimura, Nachan, Iwai, Motokawa, Lia and Kitamura, who helped me multiple times and were very useful in giving me feedback on my research.

To the various people I met in Japan. Everyone who makes my life here better, friends, university staff, everyone who has helped me. I would like to mention Abu, Jonathan, Kai, Li, Chizuka, Tomato, Karina and Misako in particular, for having played a big role in my life in Japan.

To Samuel, Rodrigo, Mari and Bea for entertaining me in Tokyo. To my friends in Brazil. There are too many to cite but I would just like to say that the "prostitutos" group is "doidera".

And thank you, reader. If you're going to read everything, I definitely hope you enjoy.

Contents

List of Figures

List of Tables

Abbreviations

LAH **L**ist **A**bbreviations **H**ere

Dedicated to Ana Fonteles, my mother

Chapter 1

Introduction

No application can be used in every context. Learning applications are designed and optimized for specific situations and learners. Whether or not they will keep the same learning gains in different contexts is something that has to be verified. Not just learning gains but the application might not even be usable in a different context. For instance, an application in Japanese cannot be used in English without translating the text in the application. However, this problem can be more complicated depending on the context. What if the students cannot read ANY text? What if the students instead have little time and the activity needs to be shortened? What about a context where students experience high cognitive load and that might be diminishing their learning gains? How to redesign activities for these tasks? Before we address this issue on how to redesign activities, let us first focus in one way to create them in the first place.

One approach for designing learning applications is the Open Information Structure Approach[1, 2]. It is grounded on multiple theories:

1) Some researchers in cognitive science have pointed out mental representation structures as being relevant to learning Pitt states that thought can be modeled through mental structures [3].

2) Vosniadou proposed conceptual models for learning physics[4].

3) Furthermore, researchers have theorized that people learn through interactions with a context, through activities[5].

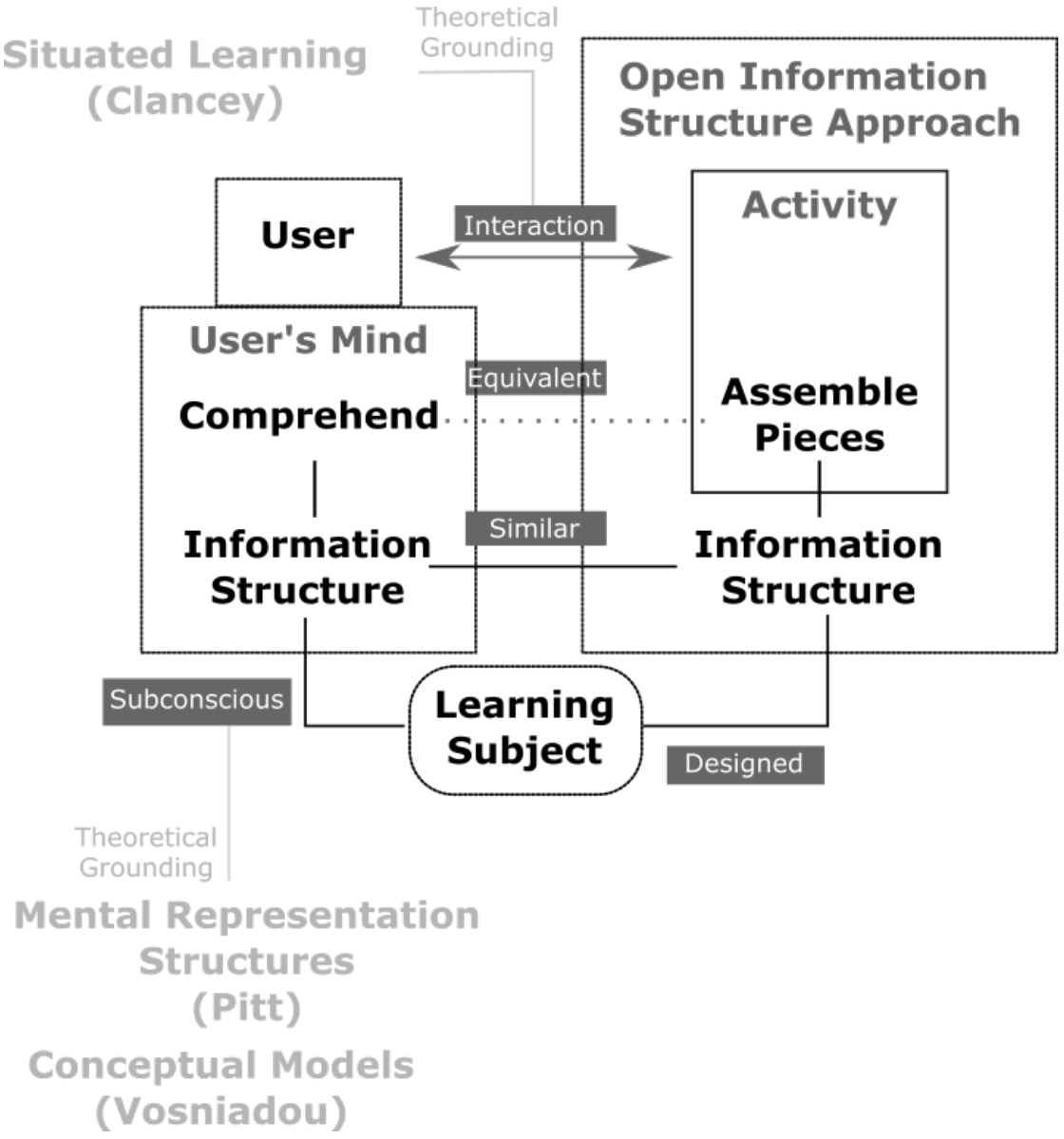
Combining contextualized interactions, mental representation structures and information science, The open information structure method was proposed[1, 2]. Various software have been developed under the open information structure method and have been shown to improve learning [6, 7, 8, 9]. In the open information structure approach, the

learning subject is modeled as an information structure. Then, this information structure is broken down into pieces and manipulated by the users. Users can then use these pieces to construct complex structures. We mentioned before that there are multiple theories that ground the Open Information Structure Approach. Figure 1.1 illustrates how the theories relate to the information structures. Vosniadou and Pitt theorized representing the thoughts and understanding of students as mental structures. This provides theoretical grounding on the link between the information structure and the learning subject as seen in the Figure. By mapping the learning subject and making it interactable, the approach taps into the situated learning theory of Clancey, allowing users to learn by manipulating those structures. This is illustrated by the ”interaction” link in the figure. Since the structures being manipulated are meant to mirror the structures that are forming in the minds of the users, the activities aim to externalize the thought process. The representation of the learning subject in the mind of the user is represented by the ”subconscious” link. The similarity between the information structures can be seen in the ”similar” link. The struggles users face while trying to assemble together the pieces aims to mirror the struggle to understand the learning subject. This mirroring is illustrated by the ”equivalent” link in the figure. This helps bring those struggles to the surface, making them accessible to educators and for use in feedback functions.

This is similar to the Model-View-View (MVC) model of software engineering[10]. However, instead of it being at the software level, it is an MVC at the activity level. One advantage of MVC is that you can change the controller and the view without having to change the model. One could argue, though, that any learning application can be designed using MVC, so it is not an advantage of the Open Information Structure approach. However, we are talking about redesigning learning activities, not just software. Open Information Structure approach can used MVC at a conceptual level but not use it at a software level. Often if you change the controller and view of the software, you might greatly affect how the users learn using the application. What we are proposing here is not changing the controller and view of the software, but of the activity, regardless of the actual data structures in the implementation.

These information structures should be transferable between different contexts. As such. it should be possible for only the way the interaction happens to be redesigned. This way, the information structure itself would remain unchanged. This can be visualized in figure 1.3. This research aims to redesign open information structure based applications and verify the effectiveness of the process. While the redesign is at the activity level, there are other benefits related to maintaining the information structure, specially related to log data analysis, which is often dependant on the information structure of the subject. This way we can apply similar analysis across the redesigned applications.

FIGURE 1.1: Open Information Structure approach related to the grounding theories



On Figure 1.2, the possible benefits of the redesign are visible. Those include the expansion of the userbase by allowing more people to ue the application and also how use of the application can be improved for the previous userbase. By improved use we mean reduction of cognitive load or of time-on-task. Furthermore, the possibility to apply similar log data analysis techniques is also illustrated in the figure.

In this study two systems were redesigned for new contexts as use cases. In one case a arithmetic study system was redesigned for Kindergarten, where not all students can read. In the second case a concept map building application was redesigned for contexts where long activity times and high cognitive load are undesirable. Multiple experiments

FIGURE 1.2: MVC model redesign showing off possible improvements regarding improved use and higher accessibility

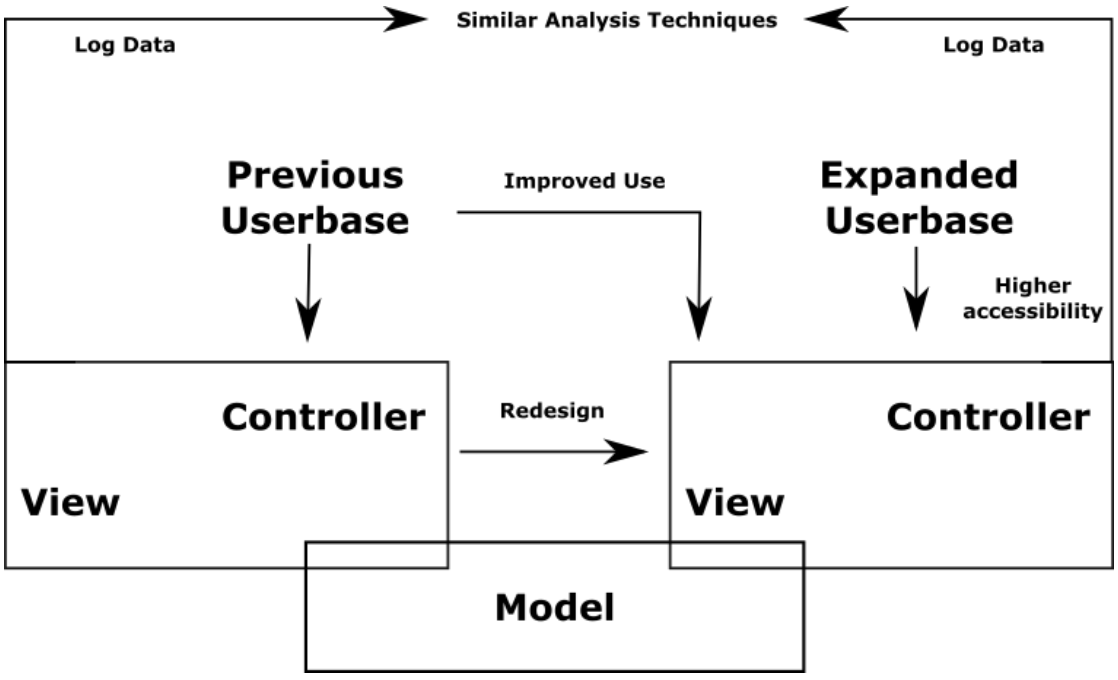
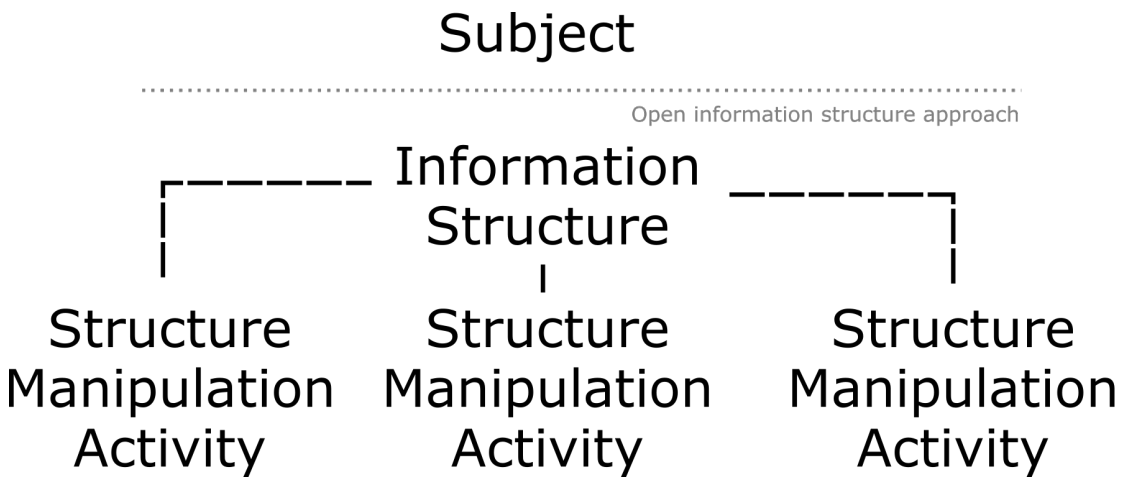


FIGURE 1.3: Expanded Open Information Structure Application for Redesigned applications



are done using the systems to answer the research questions. The research questions are listed in their own chapter.

Chapter 2

Related Works

2.1 Kindergarten Interventions on Math Skills

Past research has shown that proficiency in mathematics in the early years, such as number sense, can predict mathematical performance in later school life [11]. This has been verified all the way to the third year of elementary school. It suggests that building mathematical skills before entering elementary school can greatly impact how the student deals with mathematics in the coming years. Hence, interventions at an early age could be successful in improving students' understanding of contextual problems.

Interventions to strengthen mathematical skills for both kindergarten and pre-kindergarten students have been implemented previously. The work of [12] focuses on cognitive developmental theory to define number sense and teach it through a series of activities and games. It has found success in increasing students' understanding of number sense. The intervention of [13] also saw success, with activities based on various mathematics curricula designed for kindergarten students. The work by [14] focused on pre-kindergarten children of around four years of age. Their study focused on more than just number sense, also considering subjects such as geometric reasoning, and their results were positive.

Computer-based interventions have also been researched. The work by [15] used adaptive software that models the progress of the children and regulates problem difficulty for each child, resulting in software that can be used without supervision. The software is focused more on specific skills and on fast and accurate access, so the authors state that what is developed in the software is number sense access instead of number sense itself, describing the software as a complementary activity for children who are lagging behind in class. [16] used a software called "building blocks". Building blocks focuses

on many aspects of Math teaching, including number sense, and targets children from pre-kindergarten to grade 2. This method has found good success.

These interventions, in general, focus on whole number understanding. However, they do not focus on the conceptual understanding of contextual problems, which is a big part of how students interact with mathematics after kindergarten.

Computer-assisted interventions focused on contextual problems

With regard to contextual problems, interventions using computers have been researched before. The work of [17] provides a review of the literature on the use of computer-assisted interventions (CAI) focused on contextual problems for children with math difficulty. The review suggests that CAI are useful for supporting students with math difficulty. It also suggests that CAI can be more effective than paper-based approaches. However, the review suggests that there is not enough research to answer if computer-mediated learning is more effective than teacher-mediated learning.

The study conducted by [18] approached the problem by using algebraic thinking in the early grades. The study argued that analyzing the problems through this type of thinking required a structural understanding of the problems. The analysis of how students use the software shows that students were gradually shifting to an algebraic way of thinking while solving and understanding the problems. However, the tool deployed elements that would be hard to adapt for Kindergarten students, such as text and diagrams.

The work of [19] focuses on a tool based on graphs to help students solve algebra contextual problems. The tool is mainly used to plot graphics, to give more visual context to the equations and to approximate values. The author states that it would be useful for the tools to become a bridge between the contextual problems and the algebraic symbols. While targeting contextual understanding, an approach using graphics would be too complex for young children. In the study by [20], which also considered algebraic problems, a cognitive tutor attempted to model students and provide specific tasks for each of them. The study analyzed the problem-solving strategy variants of a computerized tutor and explored the differences between the strategies. The cognitive tutor focused more on guiding students in the various steps of problem-solving instead of focusing on understanding the problem itself. The work of [21] had elemental school students with disabilities interact with math content through PowerPoint. Users would revise the content and solve problems while receiving feedback through the system. The score of system users was significantly higher than the control group. The test used for evaluation included addition, subtraction, multiplication, and division. While the content of the system has not been detailed, it is implied that students mostly solved

word problems and received feedback on the correctness of their answer. The system used relies on reading and might not have a step focused on contextual understanding.

The work of [22] describes a problem-solving software based on dividing the problem into phases and assisting the students in solving each phase. One of the phases is dedicated to conceptually understanding the problem and another to planning and carrying out the calculations. Experimental tests were also performed. The experimental group showed better scores than the control group and participants showed a positive affective response towards the system. While the system deployed does have a step focused on conceptual understanding of the problems, Kindergarten students who are unable to read would not be able to use it.

The above research examples show various degrees of success in using computers to help students with contextual problems. Some of the work even have a focus on structural understanding. However, all of them rely on text and on procedures that would be too complicated for kindergarten students.

The main contribution of our study is that the system we propose focuses on the conceptual understanding of contextual problems while being usable by Kindergarten students. Furthermore, we have analyzed collected data from the software to investigate whether students are engaged in thinking about the structure of contextual problems. Whether or not there were improvements in the conceptual understanding of the students is also investigated. Such a study, as far as we know, has not been done before.

2.2 Cognitive Load, Learning, Interfaces and the Flow State

Cognitive load refers to the total amount of effort used in the working memory. Cognitive load theory suggests that instructional activities which are not focused on schema acquisition and automation frequently require more processing capacity than what is available to learners[23]. Cognitive load theory defines three types of load[24]. The first, intrinsic load, is a load which cannot be changed, relative to understanding the target. The second, extrinsic load, is a load caused by cognitive processes that are not related to understanding the target. The third is the germane load, which refers to the use of resources in memory to deal with the intrinsic load. As such, instructional methods should focus on reducing extraneous load while maximizing germane load.

When designing interfaces with consideration to cognitive load, one aspect is eliminating multitasking. Multitasking can cause attention splitting, which increases cognitive load[25]. One example is when two sources of information refer to each other and must

be understood together, but are presented separately. In this case, there is an additional task of keeping items in working memory because of their separation. Integrating the two pieces of information together would eliminate this additional task and free resources used to actually understand the relationship between the pieces[23]. One study found that different interface component grouping schemes affected the amount of effort users needed to use the interface[26]. When users have to spend time finding where interface components are, they are multitasking. That is, they are trying to accomplish something by using the interface and at the same time they are trying to find out where the interface components are. And while they are searching around for the interface components, they need to keep in mind what they are trying to accomplish. This is what affects cognitive load. By simplifying interface component search, users can focus on their goals. In terms of learning, that means users free up resources to deeply process what they are trying to study.

As discussed in the introduction, motivation and the flow state affect Cognitive Load. Cognitive load theories are not usually concerned with motivation, despite the problem that a learner who experiences the flow state may report low cognitive load, while in truth he experienced high cognitive load. This is a limitation of using self-reported subjective ratings to measure cognitive load[27]. In this study, flow state metrics are measured and cognitive load is measured by both objective metrics and self-reported subjective metrics. This makes it possible to see how our changes affect cognitive load and how cognitive load interacts with the flow state.

2.3 Computer-based Concept Map Tools

Computer-based concept mapping tools have been used successfully to improve learning in general[28, 29, 30] and reading comprehension[31, 32, 33]. Past studies have pointed advantages of computer-based concept mapping, such as ease of correction and construction[34], the capability to add behavior-guiding constraints[35], creation process personalization, and frustration reduction [36]. Another possibility of computer-based tools is the automation of diagnosis. One way to perform this diagnosis is by using semantic web technologies[37]. Another study used word proximity data to score the concept maps[38]. Another option is by comparing the student constructed map with an expert constructed map. Concept map tools that provide automatic diagnosis by expert map comparison are Cmapanalysis[39], Kit-build[40, 41], CRESST[42], KAS[43], and ICMLS [44]. The expert map comparison is made possible because the maps are closed concept maps. This means that the number of map possibilities is limited. The tool provides the links and concepts, so the student only has to assemble the map. Since

they are built from the same pieces, it is possible to display exactly in which ways the student map differs from the expert map. This type of automatic diagnosis was found to correlate with standard science tests[45] and was found to be reliable when compared to traditional map scoring approaches[46, 47]. It has also been used to measure changes in interdisciplinary learning during high school[48]. With the diagnosis information, teachers can revise their lessons and give more precise feedback. This approach has shown good results in retention when compared to traditional teaching, especially when the teacher uses the map to give the feedback[49]. This type of automatic diagnosis also allows for automated feedback, which has been effective for improving reading comprehension[44].

Closed Concept map assembling (CMA), from pre-existing labeled concepts and links in tools such as Kit-build, is a different process than traditional concept map creation(TMC). By TMC, we mean creating a map from scratch by creating, labeling and connecting every concept and link. By CMA, we mean connecting pre-existing, pre-labeled concepts and links to form a map. Since CMA only requires connecting the pieces, it may seem like a simpler task than TMC. However, in TMC, to build a proposition, the user has to remember relevant information, translate a portion of it into a proposition and then translate that proposition into two concepts and a link. In CMA, to build a proposition, the user has to find two related nodes, access his memory to find a relationship between them and find, among the provided links, the one which best describes that relationship. TMC involves free recalling of information and describing that information in terms of freely created concepts and links. CMA involves cued recalls, a constant search of pieces, and trying to fit one's knowledge into the concepts and links that another person made.

Results from a study that compared TMC to CMA showed no significant differences in immediate comprehension but CMA had significantly higher scores than TMC after a two-week retention period[33]. In this study, users built the map while looking at the text, so the differences in recall mechanisms were not present. The explanation given for this difference in retention was that CMA challenges students to understand the entire text in order to be able to use every concept and link provided, which doesn't help in TMC since the student creates the concepts and links himself. Furthermore, it was said that CMA requires high memory access and deeper processing of the meaning in the text. This could be interpreted as CMA having a higher cognitive load than TMC. However, some questions remain unanswered. What about the load incurred by having to manage the layout for all those concepts and links that the student did not create himself? And the visual load incurred by having so much information on the screen as soon as the building process starts? And the constant search for relevant pieces? Is that contributing to this deeper access to memory?

The changes to cognitive load proposed in this study could amplify the gains in CMA by reducing the extraneous load in the activity. On the other hand, it could diminish the gains by reducing the germane load. Since TMC and CMA have so many differences, it is hard to know what parts of CMA consist of germane load and what parts consist of extraneous load by comparing it to TMC. Since this study compares different CMA interfaces with different cognitive loads, it should shed light on this issue.

2.4 Learning Analytics

Learning Analytics (LA) is an application of analytics to learning. LA techniques have been used for various purposes, such as discovering patterns that occur only in a small number of students or rarely[50, 51], investing how different learning resources are used and the resulting outcomes[52], and investigate phenomena over a long period of time[53]. One concern of LA is finding a relationship between different variables in a dataset. This can be in the form of prediction methods, where a group of variables is used to predict one aspect of the data. One study used discussion data from students to predict the final grades of students by using latent semantic analysis and hierarchical latent Dirichlet allocation[54]. It can also be in the form of relationship mining, where the data is examined to find which variables have a strong relationship[55]. One work used relationship mining to find patterns of successful students in an engineering simulation with the goal of making suggestions on how students could improve[56]. Another work found correlations between various intelligent tutor features and non-contributive behavior by learners[57].

Another aspect of LA is processing the data for human judgment by using visualization methods. By applying various visualizations methods to the data, humans can manually identify possible relationships between variables. One work used hierarchical cluster analysis to provide visualizations of data to help educators in decision making. The method used could predict student dropout and whether or not students would take a college entrance exam. The present study also relies on visualization methods to examine the data because which aspects of the map should be examined are not clear from the start. The present study uses two visualization methods that rely on Markov chains. With the observations of the first visualization method, the second visualization method is proposed, which led to statistical analysis and to the development of numeric metrics.

2.5 Computer-based Concept Map Tools and Closed Concept Maps

Computer-based concept map tools have been used to improve both general learn in general[28, 29, 30] and reading comprehension[31, 32, 33]. The advantages of closed concept maps have been cited as ease of correction and construction[34], the capability to add behavior-guiding constraints[35], creation process personalization, and frustration reduction [36]. Computer-based concept map tools also make automated feedback possible. This automated feedback has been done by using semantic web technologies[37]. Another study used word proximity data to score the concept maps[38]. If the maps used are closed concept maps, then it is trivial to compare the student-built maps to the expert map. Multiple concept map tools used this approach, such as Cmapanalysis[39], Kit-build[40, 41], CRESST[42], KAS[43], and ICMLS [44]. Since they are built from the same pieces, it is possible to display exactly in which ways the student map differs from the expert map. This type of automatic diagnosis was found to correlate with standard science tests[45] and was found to be reliable when compared to traditional map scoring approaches[46, 47]. It has also been used to measure changes in interdisciplinary learning during high school[48]. With the diagnosis information, teachers can revise their lessons and give more precise feedback. This approach has shown good results in retention when compared to traditional teaching, especially when the teacher uses the map to give the feedback[49]. This type of automatic diagnosis also allows for automated feedback, which has been effective for improving reading comprehension[44].

2.6 Learning Analytics Applied to Concept Maps

Learning analytics has been applied to educational software which uses concept maps in the past. One work compared learners' concept maps to expert maps and to lists of misconceptions, to diagnose students' learning and to find their misconceptions[58]. It uses tabletops coupled with CMapTools. The teacher in the study could visualize students information by using a tablet, at realtime. The analytics used in this study present the current status of the map to find misconceptions and show them to the teacher. It does not attempt to relate that data to external information or to correlate them to other learning metrics.

Another study used concept maps to evaluate how well students learned and coupled that with analytic methods to redesign a learning environment. In this case, the concept map is an evaluation method that was manually scored[59].

Data from two intelligent tutoring systems that feature concept maps were examined in past research to design and evaluate knowledge tracking variables[60]. The number of concepts was a significant predictor of knowledge in one of the two systems. For closed concept maps, however, a complete map always has the same number of concepts, since the pieces are fixed. As such, this result is hard to apply to closed concept maps.

Another study uses concept maps as a visualization method for students performance by generating a concept map based on the tests performed by the learners[61]. In this case, the concept maps are artifacts generated by the analytical methods, instead of the concept maps being the input.

Betty's Brain is a learning by teaching environment where students construct the knowledge of an agent by using concept maps[62]. Automated analysis of learners' actions has been coupled with discourse analysis in a past study on Betty's Brain[63]. Different collaborative behaviors were associated with different learning performances. Some of the modeled concept map constructing actions include adding nodes, adding links, removing nodes, removing links, and highlighting parts of the map. However, the map actions are coupled with discourse to check how students were collaborating. The actual map building process in isolation and how it relates to learning gains was not observed. One work used hidden Markov models to analyze log data in Betty's Brain[64], to check the effects of metacognitive prompts. Another analysis also used hidden Markov models but coupled with reinforcement learning to generate more data, which is then fed to a new model, alongside the old data[65]. However, those two studies that used Markov models only used data related to the various high-level functions in the system, without delving into the data related to the map building process.

One work evaluated concept maps using the number and depth of the concepts to evaluate students' understanding[66]. It uses manual verification by teachers to check which propositions are correct. It was said that teachers believed the evaluation method use mirrored homework scores, but no correlation was provided. This makes it hard to know how effective the method is. An extension to the method using Markov chains has been proposed, but the Markov chain also relies on depth information and amount of elements in the concept map, not including evaluation of the correctness of the propositions[67].

While learning analytics have been applied in great extent to concept maps, few works target the map building process and closed concept maps. The works that do include closed concept maps focus on higher level functions instead of the actual map building process. Relating content in a concept map to individual questions in externals tests in a depth higher than the correlation of average scores has not been done before, as far as we have researched.

Chapter 3

Research Questions

In this chapter the research questions of the study are introduced.

Can applications built using the open information structure approach be redesigned while keeping the information structure impact?

Do the redesigned applications fit their new context appropriately?

How does the redesign process affect learning gains in the new context?

In the following chapters we introduce use cases to solve these questions. The first use case can only address the first two research questions but the second one addresses all of them.

Chapter 4

Case Study 1: Kindergarten Monsakun

4.1 The System

4.1.1 The Development Process

The development team consisted of four people. All members influenced the design of the application. One member of the team was in charge of programming, designing the screens and creating the graphics. One of the team members had years of experience as a primary school teacher. Given his understanding of how to handle children and of teaching mathematics, he was able to point out various shortcomings in the application. He also had experience working with Monsakun inside the classroom. The other two members contributed by helping design the activities and by suggesting possible solutions to the problems that were pointed out by the team.

Developing the software involved creating features, having the team members test and analyze the current state of the software and then deciding on improvements, additions, and cuts. As such, the development process was iterative in nature[68]. The team would have a meeting and decide on the tasks that had to be completed until the next meeting. The time interval between the meetings was defined based on estimations made by the developer. This process continued until all four team members were satisfied with the application.

Ideally, the constructed activities should have been tested with children during development. However, the tight schedule of Kindergarten in Japan has made this unfeasible. The presence of a person who has experience as a primary school teacher is thought

to mitigate this issue, but not to eliminate it. The use of pictures similar to Japanese textbooks in the application and software demonstrations with Kindergarten teachers also helped mitigate this issue.

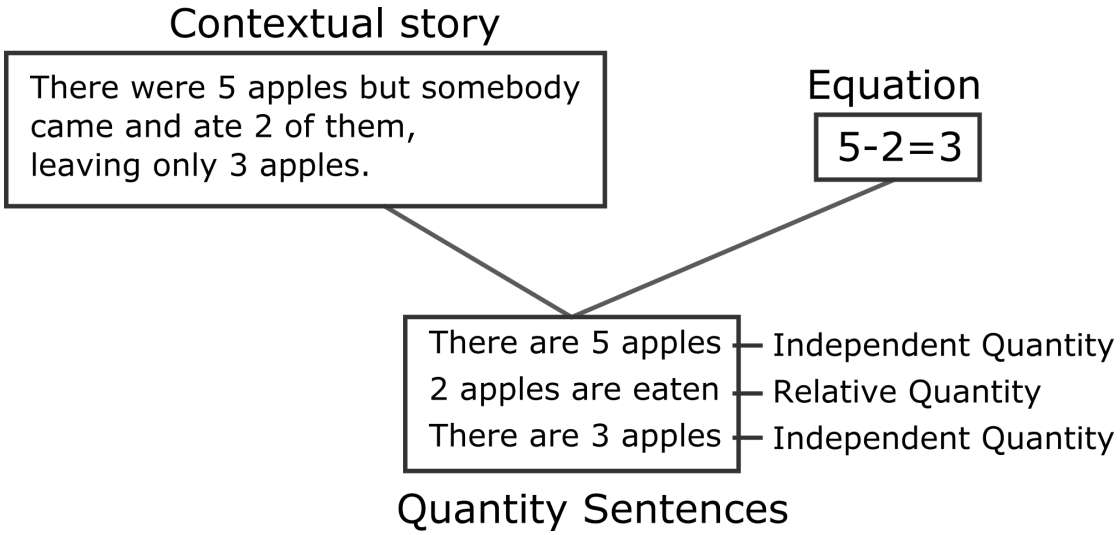


FIGURE 4.1: The Triplet Structure Model. A diagram showing the composition of the triplet structure model

4.1.2 Triplet Structure Model

The Triplet Structure Model binds the three quantities of one arithmetical operation (operand, operant, and result quantities) to contextual story roles by using three quantity sentences. This is illustrated on Figure 4.1. It is usually used to describe arithmetic contextual problems where one of the quantities is unknown.

These quantity sentences can be classified into two types: independent quantity and relative quantity sentences. Independent quantity sentences state the existence of a certain number of objects. For example, 'there are two apples'. Relative quantity sentences depend on the previous existence of a certain number of objects. 'Two apples were eaten' is an example of relative quantity. Relative quantities operate on one or more independent quantities.

The example in Figure 4.1 contains a case of subtraction story. The Triplet Structure Model refers to this type of story as a decrease story. An example of an increase story is given below:

1. There are five apples (independent quantity);
2. Two apples were brought (relative quantity);
3. There are seven apples (independent quantity).

And for combination stories:

1. There are three apples (independent quantity);
2. There are two oranges (independent quantity);
3. Put together, there are five apples and oranges (relative quantity).

Both increase and decrease stories are composed of one independent quantity in the beginning, one relative quantity in the middle, and another independent quantity in the end. The relative quantity shows how much the amount of the object changed (increased or decreased, depending on the story), while the independent quantities show the number of objects before and after the change. All three quantities in increase and decrease stories must refer to the same object, or else the story is invalid (“there are two apples, one apple is eaten, there is one banana” is not valid, for example).

Composition stories have a slightly different structure. In composition stories, the relative quantity comes at the end, with two independent quantities coming before it. The two independent quantities must then refer to different types of objects, while the relative quantity describes the total number of the two objects together.

In decrease and increase stories, the relative quantity created a change in the amount of a certain object. In the composition case, the relative quantity provides a numeric observation of the previously defined quantities without changing their value. Different objects here could be “John’ s apples” and “Mary’ s apples” , while the relative quantity would be “John and Mary’ s apples put together” .

There is one more story type that is outside of the scope of this paper, namely, the comparison story. Due do to the nature of this story, we found it difficult to show this by using pictures and decided to not include it in our design. However, it may be included in future research after a satisfactory representation is found. More information on this and on the Triplet Structure Model can be found in the work of [6].

Application Design

The application design must allow for interactivity with the Triplet Structure Model without relying on text. Our solution has been to use pictures and spoken sound. There are two types of pictures. The first type is rectangular in shape and is called an overall story picture. This type shows the entire problem at once. The second type is the story piece picture. These are small, square and represent each sentence in the Triplet Structure Model. Understanding the relationship between the overall story pictures and

the story piece pictures is like understanding how a problem is structured in the Triplet Structure Model. The design of the pictures will be further introduced below. In the application, we also ask for users to connect the pictures to numbers. This connection brings them a step further to connect the in-context parts of a problem to the out-of-context parts of an arithmetic expression. This type of connection is in-line with the conceptual understanding of contextual problems described in previous sections.

We have divided the application activities into levels. Each level is described in more detail below. Level 3 is critical to the application. It focuses on connecting story piece pictures to big pictures. Performing this requires students to be able to divide the big picture into three small parts, each related to a meaningful quantity.

Level 6, which connects the pictures to numbers, is also worth noting. First, we ask students to connect story piece pictures to numbers. Since each small picture refers to one quantity, connecting one meaningful quantity to one number is not a difficult task. However, later, the application requests users to connect one big picture to three numbers. Since no other help is given to the user, they have no choice but to visualize the three numbers related to the big picture. This is like writing the arithmetic expression of a given problem while being given only the picture of the problem. Students that can perform this task well should be able to meaningfully connect contextual problems to their arithmetic expressions. These levels will be explained further below.

4.1.2.1 Image and sound design in connection with the Triplet Structure Model

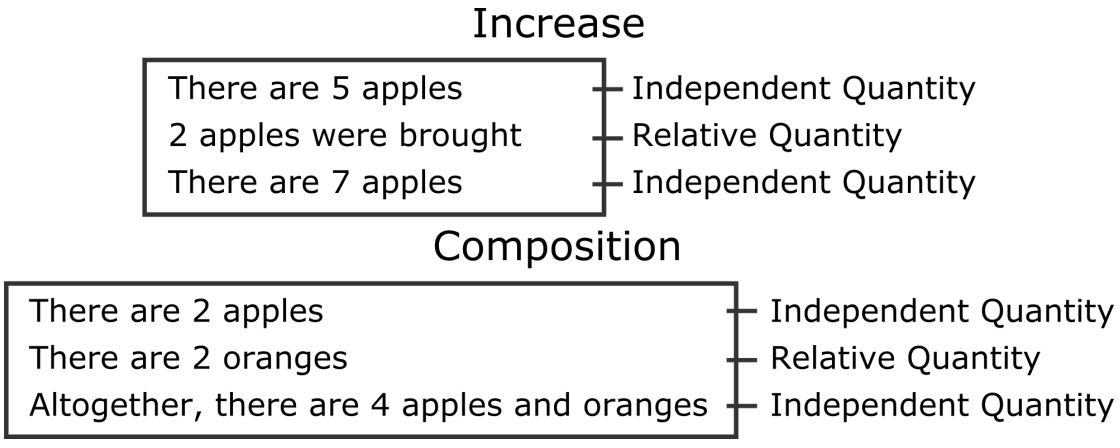


FIGURE 4.2: Increase overall story and story pieces. The two representations of an increase story

The two picture representations of a Triplet Structure Model story can be seen in Figure 4.2. Each picture also has an accompanying sound. On the left, the overall story is

shown, where one picture contains all the information to describe a story. On the right, story pieces are shown, in which three pictures are put together to describe a story.

While in the model we would have, "There are three apples", in the picture's accompanying sound we would have "there are three apples on the shelf". We describe the place that the objects are in to strengthen the connection between what the sound is saying and what the picture is showing. Story piece pictures come in two types, independent and relative pictures. They correspond to independent and relative quantities in the Triplet Structure Model. Independent pictures are usually composed of stationary objects. Relative pictures will usually describe an action, like a boy inserting objects somewhere or an animal entering a place. By describing an action or movement, we can convey the same idea as the relative quantities of the Triplet Structure Model. We can be confident in the children's interpretations of the pictures because very similar designs are used in the textbooks used in Japanese schools.

Overall story pictures represent the entire problem in a single picture. In the overall story picture in Figure 4.2, the three numbers of the problem can be seen, as:

1. The number of watering cans on the shelf;
2. The number of cans the boy puts on the shelf;
3. The total number of cans after the boy places them on the shelf.

The three numbers are mapped to the three-story piece pictures, creating the relationship between the overall picture and the story pictures. The sound related to the overall picture is a simple combination of the sounds of the three-story pieces put together. While connecting the overall story to the three pictures may seem like a trivial task, it is not so simple. While the first two numbers are quite clear in the overall story picture, the third number, which represents how many cans there will be on the shelf, requires more thinking. Students have to understand the described story and then recognize that there will be two watering cans on the shelf after the boy has finished. This number could be calculated by counting or by mental addition, either is valid. What matters is if the student can interpret the story or not. Since not all quantities are explicitly shown, connecting the overall story picture to the three-story piece pictures requires more than simply looking at the photo and trying to match the objects or scenery.

4.1.2.2 Level 1

At Level 1, participants listen to audio describing a picture and then choose, from three pictures, which picture corresponds to the audio. At this level, the pictures are story

pieces and not the overall story pictures. This is an introductory stage to introduce the pictures that can make up a story and their corresponding description.

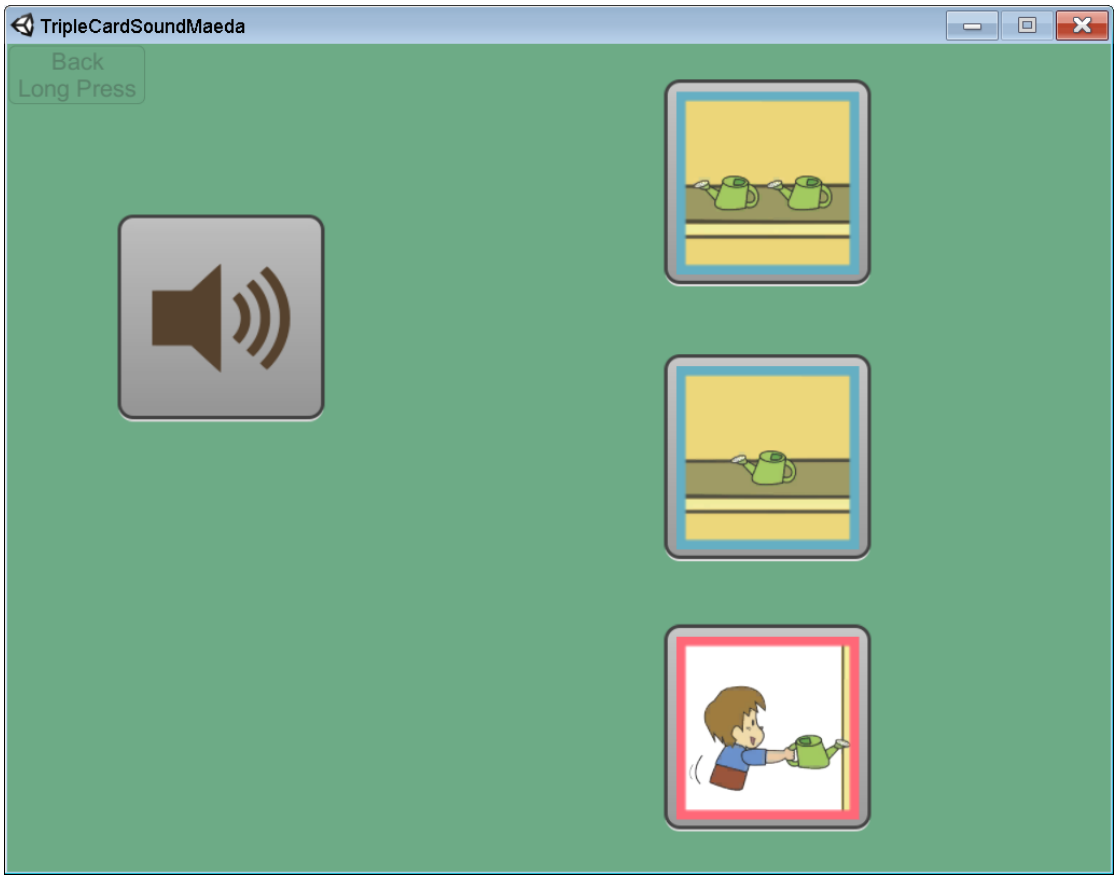


FIGURE 4.3: Example of a problem in Level 1. Screenshot of a level 1 problem

Figure 4.3 shows an example. In this case, the student would hear one of these three phrases spoken out loud:

1. There are two watering cans on the shelf;
2. There is one watering can on the shelf;
3. John puts a watering can on the shelf.

Then the student would have to touch the picture that corresponds to the phrase that they heard. The student can also repeat the sound by pressing the sound button.

4.1.2.3 Level 2

Level 2 focuses on connecting the overall story pictures to their spoken narration, to ease students into understanding the content of the pictures. Unlike story piece pictures, the pictures in this level have a description comprised of three phrases, each one corresponding to one of the quantities described in the story.

Level 2 is made up of two parts, Level 2-1 and Level 2-2.

Level 2-1 This level is based on true or false problems. Participants hear the spoken narration and are shown one overall story picture. They must decide if the picture described in the narration is the picture being shown or not by pressing true or false.

Level 2-2 Part two is like Level 1, where participants are shown three pictures and audio, having to decide which picture corresponds to the audio.

4.1.2.4 Level 3

Level 3 focuses on connecting overall stories to their story piece pictures and it is made up of two parts.

Level 3-1 In this part, we have true or false problems, with participants being shown one overall story big picture and three-story piece pictures. They then must decide if the three-story piece pictures correspond to the same story being shown in the big picture or not by choosing from true or false buttons.

Level 3-2 Participants are shown an overall story picture and given multiple story piece pictures in this part. They are asked to use three of the story piece pictures to make up a single story. The made-up story must correspond to the same story being shown in the overall story picture. Figure 4.4 illustrates this setup. Participants are given five pictures. Since only three pictures make up a story, the remaining two pictures are dummy pictures. Dummy pictures are added to give students more to think about. Problems in Level 3-2 do not all have the same difficulty, the number of blank spaces varies as described below:

1. One blank: Problem 1 to 5. There is only one blank space. The other two spaces are automatically filled. The player can move three of five cards.

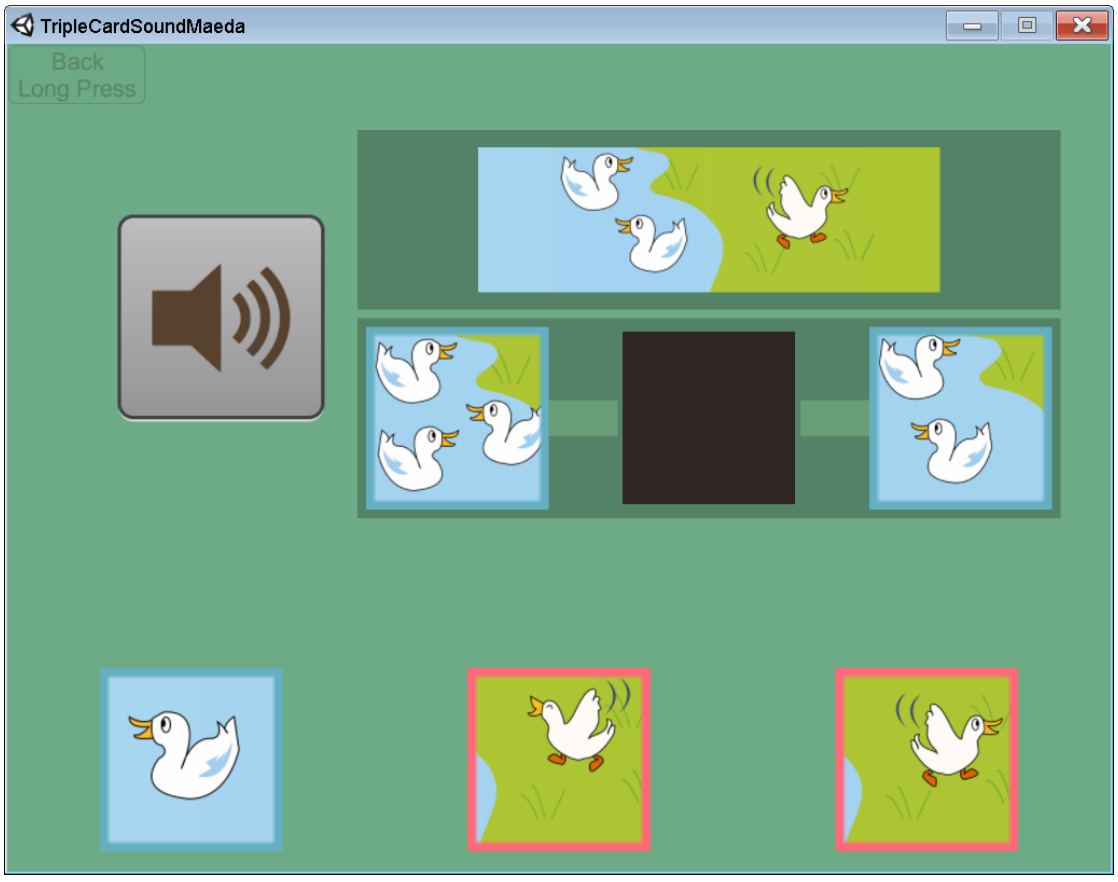


FIGURE 4.4: Screenshot of a level 3-2 problem

2. Two blanks: Problem 6 and 7. Two blank spaces. One space is automatically filled. The player can move four of five cards.

3. Three blanks: Problem 8 to 11. No blank space is filled. The player can move all five cards. This difficulty progression is used to ease the participants into working with Level 3 and constructing a story from three pieces. We stress once again that this is a key skill in the context of the Triplet Structure Model.

4.1.2.5 Level 4

Level 4 requires participants to listen to a narration of a story and then they are tasked with forming the story by using story pieces. This is similar to the second part of Level 3. The difference is that in Level 3 the correspondence was with the overall story picture, while in Level 4 it is with the spoken narration. This level is composed of eight problems, with the first three being easier, only allowing users to move three pictures out of five. The rest of the problems allow the user to move all five pictures.

4.1.2.6 Level 5

Level 5 shows participants an overall story picture and asks if that story belongs to a certain story type (the types being “increase”, “decrease” and “combine”), with the participant having to choose “true” or “false”. This level relates to how well the participant understands the concepts of increasing, decreasing and combining. It also relates to how the participants comprehend the story of each picture.

4.1.2.7 Level 6

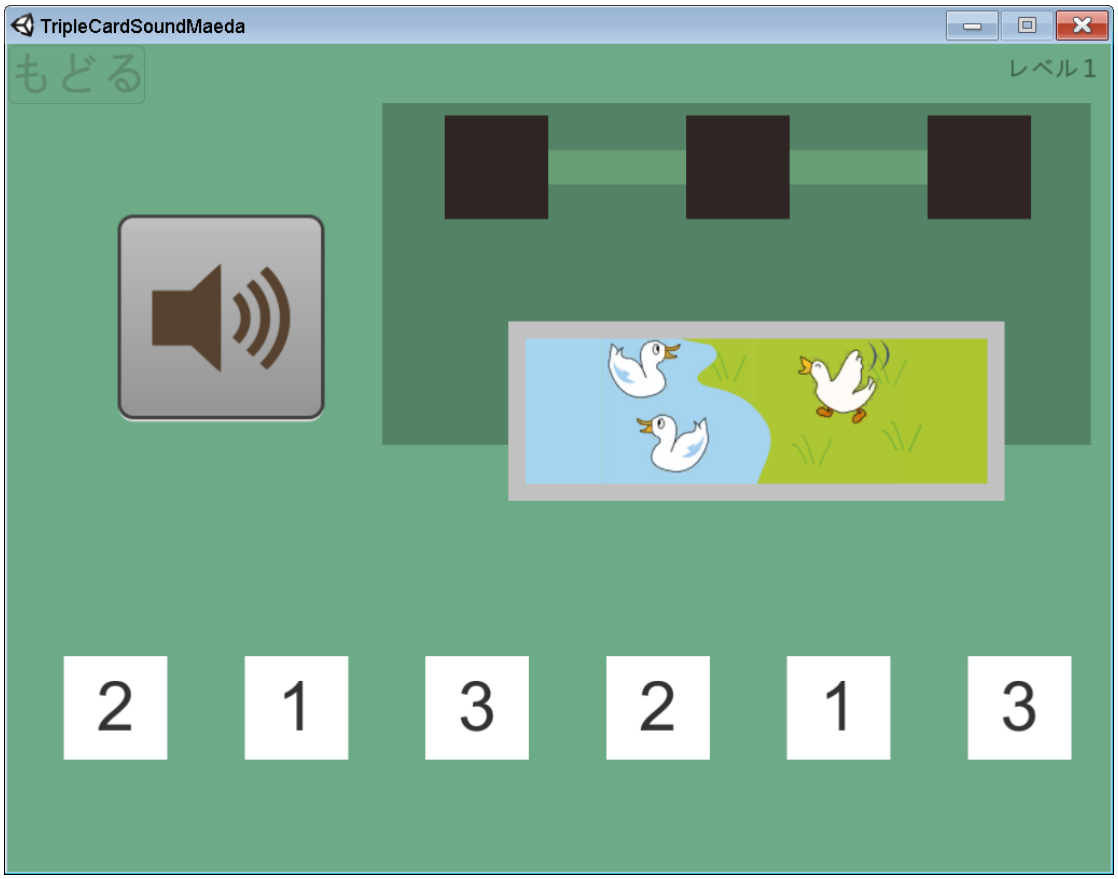


FIGURE 4.5: Screenshot of a level 6 problem

In this level, participants connect numbers to pictures. It is divided into two parts. In part one, students are tasked with connecting numbers to story piece pictures. In part two, students are asked to connect three numbers to a single overall story picture.

The setup that can be seen in Figure 4.5. As stated before, this is a problem that requires a deep understanding of the three quantities that can be interpreted from a single picture. Users that can do this should be able to construct mental models that allow them to be successful problem solvers.

This section introduced the application' s design, sound design, picture design and explained each level of the application. Which levels are critical to the conceptual understanding of contextual problems and why they are critical has also been discussed.

4.2 The Experiment

4.2.1 Methods

100 Japanese kindergarten students participated in the experiment. They were around five years old and they were divided into three classes. All students belonged to the same kindergarten. The application was introduced in around 10 minutes. During this time, four to five problems of the first two levels were shown by using a projector connected to a tablet. The participants were encouraged to give their opinion on the answer while the teacher advanced through the problems. Afterward, students had around 20 minutes to interact with the application by themselves, by using an android tablet that contained the application and headphones. There was one tablet available per student, so no sharing was necessary. The use of the application by the students and the explanations were recorded on video.

Afterward, user log data was extracted from the tablets for analysis. Because of privacy issues, it was not possible to associate student information recorded on video to their respective log data. The data includes various events that can happen during application use. Examples of these events are a list of problems being started or a button being pressed. A snippet of data can be seen below.

```
config choice, LevelConfig, 2016−12−31−12:26:26, 11.6,
level choice, 0, 2016−12−31−12:26:30, 15.16,
problem list start, ProblemList − Level 1, 2016−12−31−12:26:30, 15.16,
problem in list start, 0, unitCardVsSound, 2016−12−31−12:26:30, 15.18,
problem in list start, 1, unitCardVsSound, 2016−12−31−12:26:32, 16.84,
problem solved, 2016−12−31−12:26:32, 16.84,
problem mistake, jouro1boyadd, 2016−12−31−12:26:32, 17.66,
```

This raw data was then processed into an intermediary comma separated value (CSV) format for easier analysis. The main interest of this intermediary data was the number of mistakes that each user made while trying to solve the problems. An example of this data can be seen in Table 4.1.

The measured performance of participants was compared to a calculated "gaming the system" performance metric. Gaming the system (GTS, will also be used as "to game

TABLE 4.1: Example of intermediary CSV generated from raw data

session	list	problem	mistakes	operation	difficulty	probability
5	ProblemList Level 1	0	2	exist	1	0.3333333
5	ProblemList Level 1	1	0	add	1	0.3333333
5	ProblemList Level 1	2	1	exist	1	0.3333333
5	ProblemList Level 1	3	1	exist	1	0.3333333

the system"), refers to the behavior displayed when students attempt to systematically take advantage of the way the system is made. One way users can GTS is by randomly picking options. This definition and further discussion can be found in the work of [69]. We used the probability that students will solve a level on their first try as a performance metric. This probability depends on both the level and the difficulty of the problem.

For example, on Level 4, the probability, by random chance, of solving the first three easy problems in one try is 1/3 (only one option with three choices). The probability of solving the last five problems is 1/60 (five choices on the first card, four choices on the second card, and three choices on the third card). The calculation (3 * 1/3 + 5 * 1/60) / 8 gives us the probability of solving a problem in Level 4 in one try, by random chance. We can use a similar logic to calculate the average number of attempts.

However, the calculated values are based on which problems the students have attempted on each level. While uncommon, students could have stopped midway through a level, restarted and completed the level. This means that the calculated values are based on which problems the participants have completed for this study.

To analyze the data collected from the students we used two metrics. The first was the average number of attempts per problem. That is, the average number of times they try to get to the correct answer. The second metric was the ratio of problems solved in one try to the total number of problems attempted (RPOT). The metrics are calculated for every student/level pair. The first metric showed how much difficulty a participant had with the problem. The second one revealed signs of problem mastery, since solving the problem in one try implies that the student obtained the required knowledge related to the problem. We also analyzed peaks of difficulties. That is, we analyzed when students are first introduced to harder problems and how they coped with the problem when they saw it again, by comparing their performances.

For example, looking at the data in Table 4.1, the student related to session five took three attempts to solve problem one, one attempt to solve problem two, and two attempts to solve problem three and four. If these four problems were the only problems in Level 1, the average for this student for Level 1 would be two tries per problem. Since he only solved one problem in one try, his RPOT would be 0.25.

TABLE 4.2: Number of participants who cleared each level of the application

Level	N
1	100
2-1	90
2-2	89
3-1	20
3-2	14
4	13
5	9
6	2

TABLE 4.3: One sample t-test results comparing measured number of attempts per problem to the equivalent "gaming the system" calculated value. sd stands for standard deviation.

Level	Avg. N. Attempts	Avg. N. Attempts (GTS)	t	df	p
1	1.16 (0.24)	2.00	-34.79	99.00	<0.001
2-1	1.17 (0.23)	1.50	-13.61	89.00	<0.001
2-2	1.24 (0.25)	2.00	-29.31	88.00	<0.001
3-1	1.30 (0.32)	1.50	-2.77	19.00	0.01
3-2	4.77 (3.80)	13.18	-8.28	13.00	<0.001
4	1.44 (0.66)	18.44	-93.11	12.00	<0.001
5	1.19 (0.18)	1.50	-5.38	8.00	<0.001
6	1.31 (0.27)	8.25	-37.00	1.00	0.02

Statistical analyses were run in the group of students as a whole using the metrics defined above. Furthermore, the interactions of students that performed similarly to the calculated GTS metrics were individually analyzed as use cases.

A pre-test and a post-test were not included in our pilot study due to constraints on time in the schedule of the school. Log data was only collected when the user completed a level. This means that our analysis did not include some data from levels that were stopped midway through.

4.2.2 Results and Discussion

4665 lines of raw data were collected. To prepare for the analysis, the data was converted into 2578 lines of intermediary CSV. No data was eliminated.

4.2.2.1 Overall Analysis

Students were quick to progress through Levels 1 and 2 and found difficulty with Level 3. During application use, students were deeply focused on the application. Other than using the software they commented on the progress of each other and asked for help

TABLE 4.4: One sample t-test results comparing measured ratio of problems solved in 1 try to the equivalent "gaming the system" calculated value. Data was constant for level 6 so the test was not performed

Level	Ratio of solved in 1 try	Ratio of solved in 1 try (GTS)	T	df	P
1	0.87 (sd=0.16)	0.33	33.13	99.00	<0.001
2-1	0.84 (sd=0.18)	0.50	18.55	89.00	<0.001
2-2	0.82 (sd=0.18)	0.33	26.11	88.00	<0.001
3-1	0.75 (sd=0.27)	0.50	3.99	18.00	0.001
3-2	0.56 (sd=0.27)	0.21	4.82	13.00	<0.001
4	0.81 (sd=0.24)	0.17	9.43	12.00	<0.001
5	0.81 (sd=0.18)	0.50	5.38	8.00	0.001
6	0.88 (sd=0.00)	0.10	-	-	-

from the teachers and helpers. This has been interpreted as a positive affective response to the software. The number of users who completed each level can be seen in Table 4.2.

The average number of attempts per problem and the RPOT calculated for all levels were calculated by using the collected data. Likewise, gaming the system (GTS) versions of the same metrics were calculated. Tables 4.3 and 4.4 show that the difference between student' s performance and the calculated GTS performance is statically significant for both measures. This result shows that the strategies students employed to solve each level were more effective than GTS. This suggests that students were not blindly progressing through the application without thinking. Furthermore, it is indicative that students displayed the necessary skills to solve each level. This addresses our first research question, that kindergarten students are able to meaningfully interact with the triplet structure model.

It also shows signs that our pictures, and their audio descriptions, fit with the participants' interpretations of the pictures, otherwise, they would not have been able to perform well on the first two levels. Results suggest that students were able to use the software and their observed affective response was also positive.

Between Level 2-2 and 3-1 there was a large drop from 89 students to 20 students. This drop has been accounted to time constraints. Given how Level 3-1 contained only truth-and-false exercises, there is little reason to believe that students would not be able to complete it given more time.

Level 3-2 Figure 4.6 shows the average number of attempts of each problem during GTS for Level 3-2. Students had little trouble with Difficulty 1. The same cannot be said for Difficulty 2 and 3. There are two spikes in the graph which correspond to the difficulty changes. This shows that Difficulty 1 was not so demanding, and as a result, the students were not pushed to understand the meaning of what they were doing. As

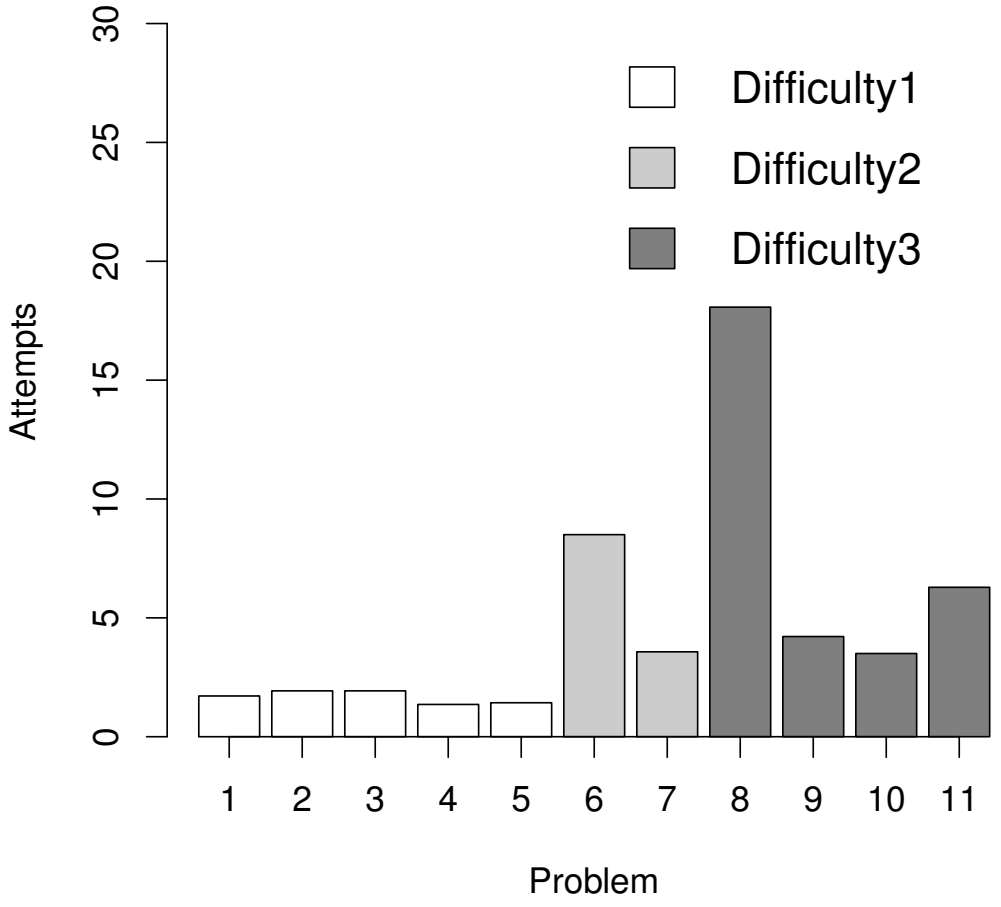


FIGURE 4.6: Average number of tries on Level 3-2 problems

soon as the difficulty went up, students struggled to solve the problem. However, after this first peak, their performance quickly increased. This means that students were quick to understand what was being asked of them after solving it for the first time.

Figure 4.7 shows a similar trend. The figure shows the value of RPOT for level 3-2.

Students had little trouble with difficulty 1 and they had difficulties with the same problems shown in the previous analysis. However, it is worth noting that around 20% of the participants managed to solve Problem 8 in one try. Level 8 is the point at which Difficulty 3 is first introduced. This means that a good portion of the students had a good understanding before the difficulty increased. The effects of the peaks are summarized in Figure 4.8. The only peak that shows a statistical significance ($t(13.83) = 2.21$, $p < .05$) is the difference between the average number of attempts between the first problem ($M = 18.07$, $SD = 22.35$) and the other problems ($M = 4.67$, $SD = 6.89$).

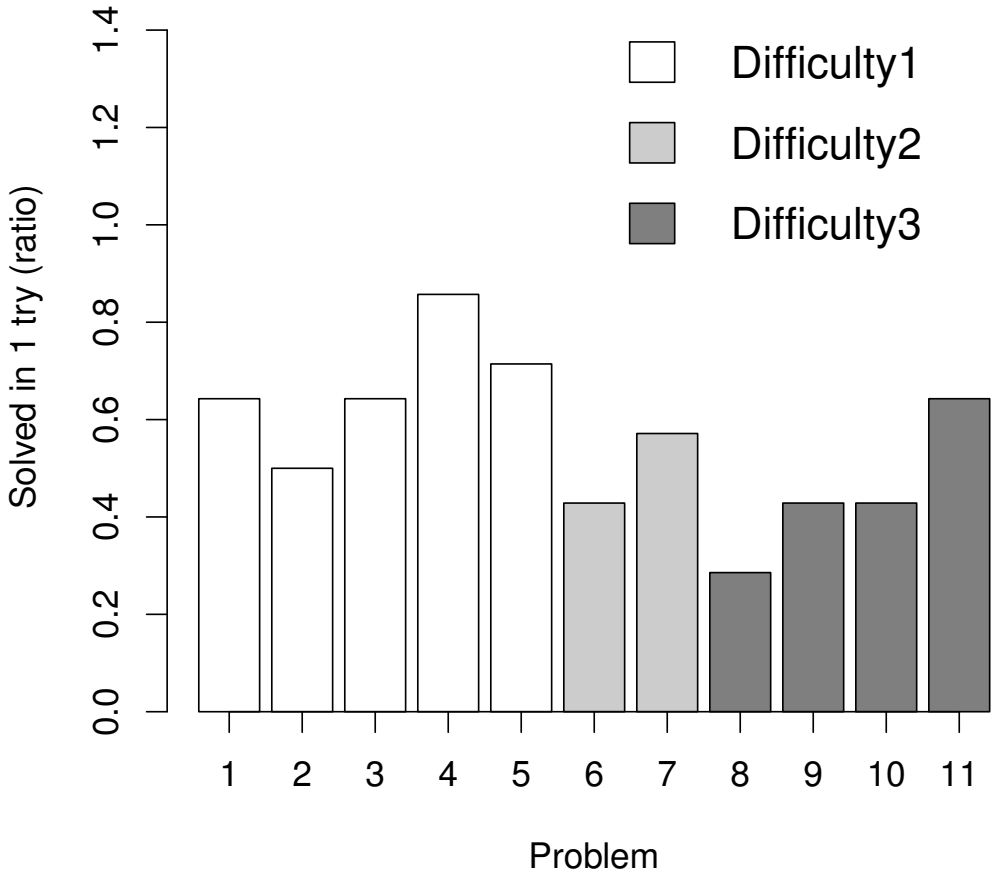


FIGURE 4.7: Ratio of problems solved in 1 try to total number of times attempted on Level 3-2

Although the other peaks that were analyzed di not show statistical significance, they all show the same trend.

These results show that students had trouble when the difficulty goes up in Level 3-2 but, after initially struggling, they could understand what was being asked of them, and showed patterns of growth.

While this trend has been verified for the group in general, it remained unknown if the students who had most trouble also showed similar patterns. Some of these students have shown performance similar to the predicted GTS metrics. To investigate whether these students also took advantage of using the system, we perform case studies for every student whose average number of tries went above 50% of the predicted values.

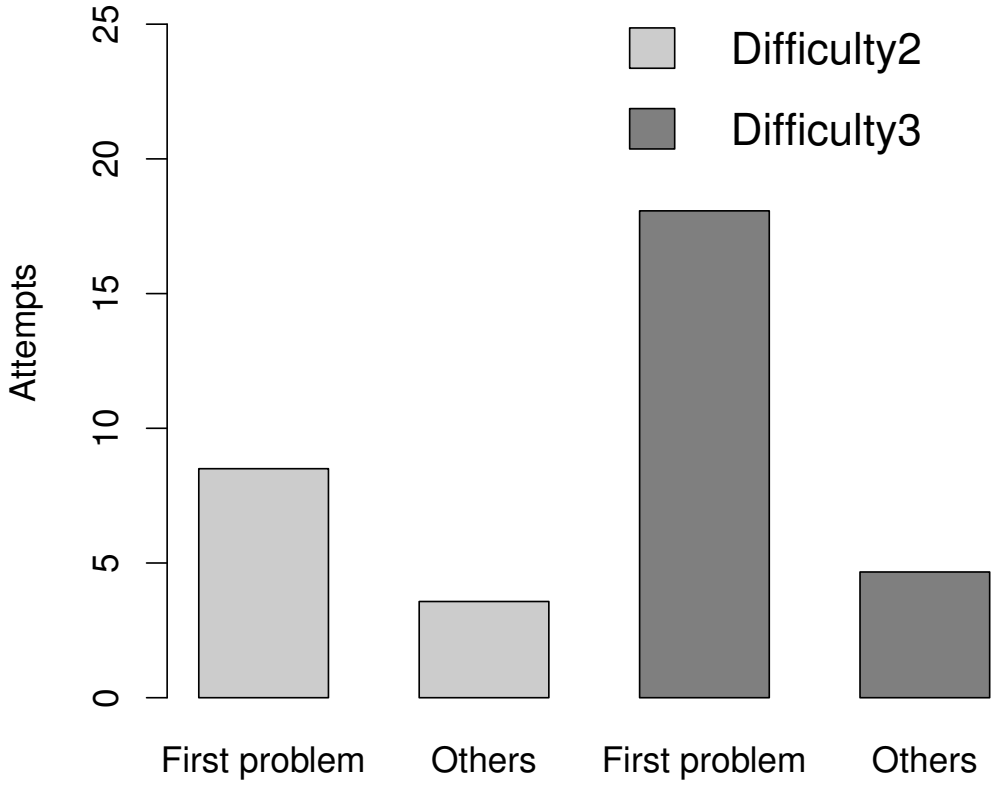


FIGURE 4.8: Comparison between performance on the first problem and on remaining problems for Level 3-2 and Difficulty 2 and 3

4.2.2.2 Case studies

The performance of four participants on Level 3-2 is analyzed in this section. This means that their performance was close to trying to game the system. This could be a sign that they may not have engaged well with the system.

1. User 1: Performance of 1.0 * GTS value. This participant had a lot of trouble in both peaks of difficulty (problems 6 and 8) and in problem 11. The participant quickly solved the other problems in Difficulty 2 and 3;
2. User 2: Performance of 0.52 * GTS value. This participant had trouble from problems 6 to 9. They quickly managed to solve problems 10 and 11;

3. User 3: Performance of 0.54 * GTS value. This participant had similar results to User 1;

4. User 4: Performance of 0.84 * GTS value. This participant had similar results to User 1.

All four students had good performance in previous levels. None of the students completed Level 4. Since these students took time to get through Level 3, the reason they did not complete Level 4 is likely because of lack of time.

Three of the four users had trouble with problems 6, 8 and 11. They had more trouble than the other participants in the pilot study and this pushed their average performance closer to GTS. That does not mean that they were not trying to actively solve and understand the problems since after going through this initial trouble, they managed to solve the other problems fast.

User 2 had a different scenario. The student had moderate trouble with problems 6 to 9, four problems in a row, and then the student' s performance went up in problems 10 and 11. The student did not have as much trouble with problems 6 and 8 as the other participants, having more trouble with problems 7 and 9. Either the student finally understood the activity while trying to solve problem 9 or they received help from another student. We cannot be certain because although it was observed that some of the students were asking for help, it is not possible to know if the student corresponding to the examined log data asked for help or not.

Users had trouble with problem 11. Of the four problems of difficulty 3 in Level 3-2, two of them are "decrease story" type, one is a "combination story" type and the last one is an "increase story" type problem. Problems 8 and 11 are "decrease story" type problems and participants had trouble with both. It could be the case that "decrease story" type problems are more difficult for students.

These results suggest that, even when students struggled at first, they could meaningfully take advantage of the application.

Results suggest that the interaction of students with the system was meaningful and that they showed growth patterns. This remains true even when we isolate the students that struggled at first. The overall analysis combined with the case studies answer our second research question, that the interactions with the triplet structure model improve as the students use the application.

4.2.3 Limitations

The pilot study lacked a control group, a pre-test, and a post-test. This makes it hard to evaluate the learning effects of the application. Time constraints also limited the amount of data collected on the harder levels of the application.

Lastly, the interactions between students, and between teachers and students, may have influenced application use. In the future, one solution to this problem would be to isolate the children, so that data would be unaffected. Another approach would be to collect data on these interactions so that it could be compared to the collected software data. This would allow for a more complete view of application use and of the social implications of the application.

Chapter 5

Case Study 2: Kit-build

5.1 Airmap & The Relation to Kit-build

5.1.1 System Design

This study uses two interfaces for the construction of concept maps. The first one is the interface of the previously mentioned Kit-build. The second one, the Airmap interface that was developed in the course of this study, will be detailed in this section. Both interfaces approach map building by providing pieces to the user. This means that in both interfaces users are not required to write the nodes and links of the concept map. There are three major differences in the two interfaces:

1. Kit-build users have to manage the position of the pieces. In Airmap, the pieces automatically position themselves.
2. In Kit-build, the user is shown all links and nodes in the same area from the very beginning. In Airmap, the user is shown only the concepts in the beginning. Links are added to the same area as the concepts according to the actions of the user.
3. To connect nodes, Kit-build users use connector gizmos. Airmap users use a link selection menu.

These three differences will be further explained below. The study used two variants of the Airmap interface. The difference is that one of the variants only shows the link selection menu when the user connects two nodes. The other variant always keeps the link selection menu visible. Differences in cognitive load between all three interfaces are further explored below.

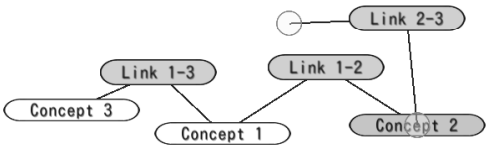


FIGURE 5.1: The Kit-build interface

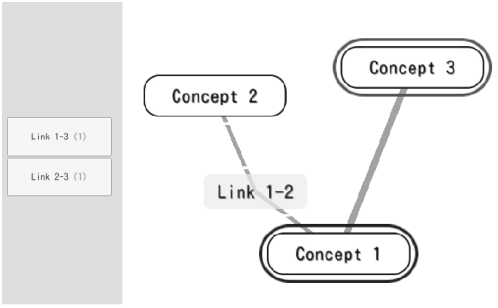


FIGURE 5.2: The Airmap interface

5.1.1.1 Kit-build

A screenshot of the interface of Kit-build can be seen in Fig. 5.1. In this figure, Concept 1, 2 and 3 are the nodes of the map. Link 1-2, Link 1-3 and Link 2-3 are the links. When the activity starts, all nodes and links are displayed in a column. Links and nodes are in separate columns.

In order to connect nodes, users have to drag-and-drop the connector gizmos of each link. The connector gizmos are two circles connect to each link by a line. When the circle comes into contact with a node by drag-and-drop, the circle disappears and the node and link are connected. In Kit-build, a link can be connected to a single node during the building process. By connecting a link to two different nodes we have a node:link:node connection. This is called a proposition.

In order to disconnect nodes from links, the users have to click on a link. By clicking on a link, the circles that disappeared during the connection process reappear. By drag-and-dropping the circles away from the connected node, the link and node become disconnected. The link can then be connected to other nodes.

In order to manage the layout, users are required to drag-and-drop the nodes and links. In no moment do the links or nodes move by themselves.

5.1.1.2 Airmap

The interface of Airmap can be seen in Fig. 5.2. On the right, Concept 1, 2 and 3 are the nodes. Users can click on a node to select it. In the figure, Concept 1 and Concept

3 are selected. When two nodes are selected, a thick line connects them. Link 1-2 is the link. Link 1-2 is connecting Concept 2 and Concept 1. Thin lines connect the link to the two concepts. On the left, there are two buttons on a menu. They show the links available alongside their available quantities.

Two variations of the interface have been built. In the first variation, the button menu on the left only becomes visible when two nodes are selected. In the second variation, the button menu is always visible.

In order to connect nodes, users have to select two nodes. Afterward, they have to click on one of the link buttons in the left menu. Then, the link will appear, connecting the two nodes. Unlike Kit-build, a link always connects two nodes. It is not possible to associate a link with a single node in this interface.

In order to disconnect nodes, users have to click on the links. Clicking on a link destroys it, breaking the connection. That link then becomes available on the left menu.

Users don't have to manage the layout. Both links and nodes move automatically. The user is unable to directly control the positions of the nodes and links. The algorithm used to handle the layout is a type of force-directed graph drawing algorithm[70]. The implementation used can be found in [71].

5.1.1.3 Cognitive load considerations between the interfaces

In this subsection, we analyze possible differences in cognitive load between the Kit-build interface (Kit), Airmap with hideable links (AirA) and Airmap without hideable links (AirB). All interfaces provide users with pre-labeled concepts and links, thus both are CMA activities. They all involve cued recalls, that is, the labeled pieces provided help users recall information. A summary of the analysis can be seen in Table 5.1.

Table 5.1: Cognitive load differences between the interfaces

Use case	Kit	AirA	AirB
Concept search		>Kit, >AirB	>Kit
Link search		>Kit	>Kit
CCL	>Kit	>Kit, >AirB	>Kit
CLC	?AirA, ?AirB	?Kit	>AirA, ?Kit
LCC	?AirA, ?AirB	?Kit	>AirA

To reason about cognitive load, we define possible interface use cases and try to model what kind of information the user has to keep in working memory during the use cases. We will first define the use cases and explain how they work generally. Afterward, we

will explain how the use cases differ between each interface. All use cases define the construction of a proposition formed by concept A, concept B, and link L.

The first use case we consider is the situation where the user picks concept A, searches for a concept B that might be connected to concept A, then searches for the link L which best describes a possible relationship between concept A and B. We will call this use case CCL (concept-concept-link). During the search for concept B, information about concept A is being constantly accessed because the user must find a concept which is likely to be connected to A. When searching for link L, it is assumed that information about both concept A and concept B are kept in working memory. Information about both A and B is constantly accessed to make a decision if the current link being visualized can connect A and B.

The second use case we consider is the situation where the user picks a concept A, searches for a link L believed to involve concept A, then searches for a concept B which is related to concept A through L. We will call this use case CLC (concept-link-concept). Similar to CCL, the user has to record the initial information in his working memory(concept A), then he has to perform two searches, with the second search (concept B) requiring the result from the first search (link L).

The third use case is LCC(link-concept-concept). The only difference to CLC is that the user will first pick a link and then search for the concepts.

The three use cases we have defined all involve searching for the same three pieces of information and only differ in the order of search. However, when analyzing the differences between the interfaces, this order of search becomes important. First, we compare AirA and AirB, the two interfaces which are most similar. In both AirA and AirB concepts get selected after clicking. However, a user cannot select a link. By selecting a concept, the user might decrease the cognitive load associated with keeping the information of a concept in his working memory since it is marked in the interface. Thus, this should facilitate both searches in CCL, the first search of CLC and the second search of LCC. Now, remember that only in AirB the user has constant access to all link labels. In AirA, the user can only see the links after selecting two concepts. Because of this, LCC and CLC are unlikely to happen in AirA. If the user does engage in LCC or CLC while using AirA, he will have further difficulty because if he forgets information regarding the link, he will have to select two nodes in order to recheck link information. However, since link information is hidden, users are less distracted when searching for concepts. Thus, the first search of CCL should be easier in AirA compared to AirB. To summarize, both AirA and AirB have mechanisms to facilitate CCL, with AirA facilitating CCL further. However, AirA has mechanisms that make CLC and LCC harder.

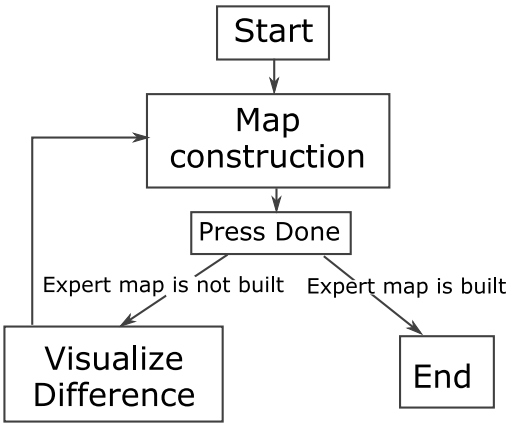


FIGURE 5.3: Flow diagram for building the map and changing it into the expert map

Finally, we compare Kit to both AirA and AirB. In Kit, links and concepts share the same space. In AirA and AirB, they exist in clearly separated spaces. Furthermore, the user may have to engage in organizing the layout during a search, putting more strain in the working memory. Thus, it is assumed that searching, in general, is more difficult in Kit, for both concepts and links. Also, Kit does not allow for the graphical selection of concepts. To compensate for this, users might develop spatial strategies, such as moving a concept or a link to a specific position in order to simulate selection. Such spatial strategies are not possible in AirA and AirB because these interfaces do not allow users to move the pieces. However, these strategies are user-enforced and there is no guarantee that users will use them. One advantage that Kit has over the other two interfaces is that users can connect a concept directly to a link, without specifying the second concept. This connection is visible. As such, during CLC and LCC, the user can connect the first concept to the link, thus reducing the load on the working memory while searching for the final concept to complete the proposition. Since during CCL the link is only identified last, this mechanism is of no help. Thus we can say that Kit has mechanisms that facilitate CLC and LCC. However, it cannot be said that Kit is better than AirA and AirB for CLC and LCC since there are other factors involved, such as differences in visual spaces and layout organization. Finally, AirA and AirB group links with the same name as one element. This is not possible for Kit. This is another element that speeds up the search for links when compared to Kit, in the case of maps with multiple links that have the same label.

5.1.2 Support for Recreating the Expert Map

The general workflow for how the student builds their map and changes it into the expert map can be seen in Fig. 5.3. At first, students build their map using Kit-build. When

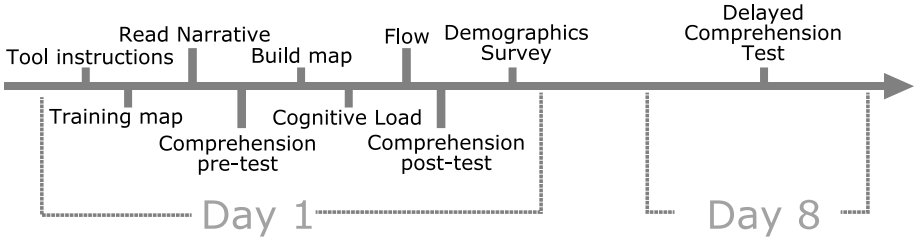


FIGURE 5.4: Timeline for Experiment 1

they are satisfied with their maps, they press the 'DONE' button. At this point, the built map is compared to the expert map by the system. If they are a perfect match, the activity is over. If they are not a perfect match, the student is shown the differences between their map and the expert map. The difference between the maps is made up of two components. The first component is the parts of the expert map which are not present in the student map. The second component is the parts of the student map which are not contained in the expert map. This support system was only used in the first experiment.

5.2 Kit-build Experiment 1

This study used a between-subjects design with three conditions: AirA, AirB, and Kit. The experiment had a main phase and an optional delayed phase. In the main phase, participants were required to:

- Read tool instructions.
- Build the training map using the tool.
- Read a narrative.
- Take the comprehension pre-test.
- Build the text map using the tool.
- Take a cognitive load survey.
- Take a flow measurement test.
- Take the comprehension post-test.
- Take a demographics survey.

The only difference between the conditions was the interface used. AirA used Airmap with hideable links. AirB used Airmap without hideable links, and Kit used Kit-build. The interfaces did not have automatic feedback enabled and users could submit incorrect maps. Participants who completed the main phase were invited to participate in the delayed phase. The delayed phase consisted of the same comprehension post-test used in the main phase, but with a delay of one week. A timeline for this experiment can be seen in Fig. 5.12.

5.2.1 Participants

Participants were recruited through Amazon Mechanical Turk (AMT) between December of 2017 and January of 2018. Participants were required to be residents of the U.S. and were also required to have completed more than 5000 tasks on AMT with an approval rate above 97%. This was done to ensure quality and avoid automated programs from participating in the experiment. Participants were also required to use a computer while participating since the experiment was not optimized for mobile devices. Participants were paid $2 upon completion of the activities. Participants who agreed to take the delayed post-test received an additional $0.15.

5.2.2 Materials

A narrative text involving sled dog racing was used[72]. The comprehension pre-test and post-test both used the same questions. The questions used were the reading comprehension exercises found in [72], but only the multiple choice questions were included. A 7-point Likert scale was used for measuring perceived difficulty and perceived effort. Previous research shows that subjective rating scales like these can be valid measurements of cognitive load[73]. We also collect time-stamped log data of software use. This data is used to calculate time-on-task and the number of actions participants take to build the map. Both are indirect objective measurements of cognitive load[74] and have been used before[75, 76]. The flow-short-scale[77] was used for measuring flow. An additional question was included in the flow short scale to measure the perceived fun of the activity. The map participants were requested to build was based on the text and on the reading comprehension exercises.

5.2.3 Procedure

The experiment was delivered through a website. Participants completed informed consent and then proceeded to read instructions on the map building tool they would use.

Afterward, they would build the training map to get used to the tool. The training map consisted of three concepts and three links. The content of this training map had no relation to the rest of the experiment. The tool instructions and the tool used to build the map was specific to each condition. After building the training map, participants read the narrative and answered the pre-test. After the pre-test, participants had to build the map using the tool respective to their condition. Then, all participants had to answer the cognitive load questions related to the map buildinfg task. This was followed by the flow measurement questions also related to the map building task. Participants then answered the post-test and completed the demographics survey, ending the main phase of the experiment.

All activities in the main phase had a 5-minute limit, with the exception of building the map, which had a 10-minute limit.

One week later, participants were contacted by email to take part in the optional delayed phase. The delayed phase consisted of the same comprehension test taken in the pre-test and post-test. Participants did nothing else other than answer the comprehension test.

5.2.4 Results

66 participants attempted the experiment. 6 users were excluded because of a data collection bug (N = 60). None of the demographics collected (gender, age and educational attainment) were related to the assigned condition under a chi-square test of independence. 28 users participated in the optional delayed phase.

To analyze the cognitive load metrics, we used a non-parametric MANOVA with Wilks's lambda because not all metrics passed a test of normality. The procedure used is described in [78]. A multiple testing procedure using Wilk's lambda was performed for the post hoc comparisons. The procedure controls for type I error and it is also described in [78]. Statistical tests were performed at $\alpha = 0.05$ to address our research questions.

To address our first research question, whether changes in layout management burden and visual load affect the cognitive load during concept map construction, we performed a non-parametric MANOVA on the four dependent variables related to cognitive load, including the number of actions, time-on-task, perceived difficulty and perceived effort when building the map. The predictor used was condition. Table 5.2 shows the sample size, the mean and standard deviation for all four dependent variables tested among the 3 conditions.

The non-parametric MANOVA revealed a significant main effect of condition on cognitive load, $F(8, 108) = 2.942$, $p = 0.005$. Post hoc comparisons between subsets of

condition using a closed multiple testing procedure revealed that the cognitive load for the Kit condition was significantly higher than the cognitive load for the AirB condition ($p = 0.003$). No other subsets of condition showed significant differences. Post hoc comparisons on subsets of the dependent variables revealed a significant difference for the number of actions ($p = 0.014$) between conditions and for every subset that contained the number of actions. No other subsets of dependent variables were significantly different.

To address our second research question, whether changes in layout management burden and visual load affect flow during concept map construction, we performed a Kruskal-Wallis H test on flow scores as the dependent variable. The predictor used was condition. Mean and standard deviation for the flow scores can be seen in Table 5.3. The test did not reveal a significant effect for condition on flow scores. All interfaces scored below flow measures during learning in an obligatory statistics course ($M = 4.60$, $SD = 1.16$) and during playing a computer game on moderate difficulty($M = 4.68$, $SD = 1.18$)[77]. Although not statistically significant, both Airmap interfaces had similar flow scores and they both outperformed flow scores reported during Kit-build use.

We could not address questions 3 and 4 because significant gains in reading comprehension could not be seen in the data for all conditions. There were six questions and pre-test scores had an average higher than five questions answered correctly. It is believed this was caused by the low difficulty of the reading comprehension test.

Our main finding for experiment 1 is that changes in layout management burden and visual load reduced the cognitive load during the map building activity. Although only the Airmap interface without hideable links showed statistically lower cognitive load when compared to the Kit-build interface, the Airmap interface with hideable links outperformed the Kit-build interface in all four cognitive load metrics. Also, there were no significant differences between the two Airmap interfaces. This means that whether the links are hideable or not should not have a large impact on cognitive load.

Given the low flow scores, we can assume that there is no reduction of the perceived cognitive load because of the flow state. As such, objective metrics of cognitive and self-reported subjective metrics should follow the same trends, which does happen for perceived difficulty. Flow scores follow an opposite trend to perceived difficulty scores, which could be interpreted as the Kit-build task being perceived as too difficult by users, thus resulting in a lower flow score by deviating from the ideal flow conditions.

Table 5.2: Cognitive Load metrics

Group	N	N. of actions	Time-on-task	Perceived difficulty	Perceived effort
		M (SD)	M (SD)	M (SD)	M (SD)
AirA	18	57.78 (20.50)	276.17 (132.14)	3.28 (1.13)	4.22 (1.22)
AirB	23	54.91 (16.39)	328.22 (144.12)	3.13 (1.32)	4.78 (1.28)
Kit	19	91.53 (44.98)	361.00 (163.26)	3.79 (1.51)	4.74 (1.05)

Table 5.3: Comprehension and flow measurements

Group	Flow	Pre-test	Post-test	Map score
	M (SD)	M (SD)	M (SD)	M (SD)
AirA	4.19 (0.79)	0.92 (0.16)	0.91 (0.13)	9.44 (4.97)
AirB	4.25 (0.80)	0.85 (0.19)	0.89 (0.16)	8.96 (3.46)
Kit	3.62 (1.14)	0.89 (0.10)	0.89 (0.17)	6.42 (5.06)

5.3 Kit-build Experiment 2

This experiment is similar to the previous experiment, but with a higher focus on reading comprehension gains. The main difference from the previous experiment is that users received automated feedback while using the tool and are required to construct the correct map to proceed. Furthermore, there was an increase in difficulty in the text read and in the content of the pre-test and post-test. These changes made it easier to answer research questions three and four, which are the main concerns of this experiment.

Table 5.4: Collected metrics for all participants in Experiment 2

Group	N	Pre-test	Post-test	Normalized Change	N. of actions	Time-on-task
Air	26	0.60 (0.19)	0.78 (0.21)	0.47 (0.40)	146.52 (69.38)	771.00 (270.41)
Kit	24	0.58 (0.22)	0.82 (0.11)	0.53 (0.30)	280.58 (268.52)	944.50 (259.82)

Table 5.5: Metrics for participants who completed the delayed test

Group	N	Pre-test	Post-test	Delayed test	Delayed Normalized change
Air	20	0.59 (0.19)	0.78 (0.21)	0.59 (0.15)	0.03 (0.26)
Kit	16	0.59 (0.22)	0.82 (0.10)	0.70 (0.17)	0.25 (0.37)

5.3.1 Design

This study used a between-subjects design with two conditions: Air and Kit. The experiment had a main phase and an optional delayed phase. In the main phase, participants were required to:

- Read tool instructions.
- Build the correct training map using the tool.
- Read a text.
- Take the comprehension pre-test.
- Build the correct text map using the tool.
- Take the comprehension post-test.

The only difference between the conditions was the interface used. Air used the Airmap interface without hideable links and Kit used the Kit-build interface. The interfaces had feedback enabled and required participants to redo their maps until they built the correct map. Participants who completed the main phase were invited to participate in the delayed phase. The delayed phase consisted of the same comprehension post-test used in the main phase, but with a delay of two weeks.

5.3.2 Participants

Participants were recruited in the same way as described in subsection 5.2.1 of experiment 1, differing only in monetary compensation and recruitment period. Participants of experiment 1 were not allowed to participate in experiment 2. The monetary differences were due to longer experiment times and to increase the number of participants of the delayed phase. Participants of this experiment were recruited during February of 2018. They were paid $3.10 upon completion of the activities. Participants who agreed to take the delayed post-test received an additional $0.80.

5.3.3 Materials

The text used described various characteristics of the Komodo dragon. It is a modified, shorter version of a text found in Wikipedia[1]. The comprehension pre-test and post-tests contained the same questions. The questions consisted of ten multiple choice questions created to test the content of the text. The map participants were requested to build was based on the text and on the reading comprehension exercises.

[1] `https://en.wikipedia.org/wiki/Komodo_dragon`

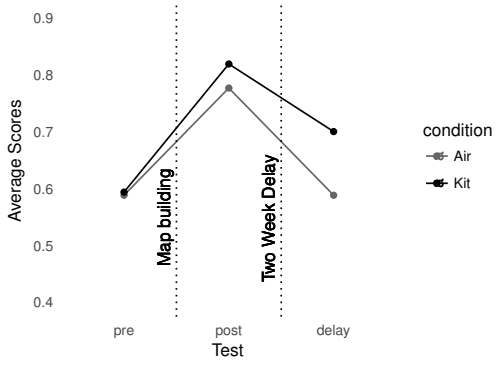


FIGURE 5.5: Pre-test, post-test and delayed-test score averages for participants who completed the delayed test

5.3.4 Procedure

The procedure was similar to the procedure of experiment 1, described in subsection 5.2.3. This section will list the differences between the procedure of the two experiments. Neither flow tests nor cognitive load tests, present in experiment 1, were present in experiment 2. Regarding the construction of the concept map, there was a change of time limit from ten minutes in experiment 1 to twenty minutes in experiment 2. Also, unlike experiment 1, users were given automated feedback by the system and could only proceed to the next task after submitting the correct map. The delay between phases was increased from one week in experiment 1 to two weeks in experiment 2.

5.3.5 Results

50 participants attempted the experiment. Of these, 36 users participated in the optional delayed phase.

To compare gains in reading comprehension, we calculated the normalized change as described in [79]. Pre-test, post-test and normalized change scores for all participants can be seen in Table 5.4. To compare retention, we also calculated a normalized change metric. However, we used the delayed test scores as the post-test instead. We call this metric the delayed normalized change. Pre-test, post-test, delayed test and delayed normalized change scores can be seen in Table 5.5. The data in Table 5.5 is only related to the students who participated in the delayed test. Statistical tests were conducted at $\alpha = 0.05$ to address research questions three and four.

To address our third research question, whether changes in layout management burden and visual load during the concept map building activity affect immediate reading comprehension gains, we analyze the data seen in Table 5.4. There is very little difference in pre-test scores, post-test scores, and normalized change. The same trends can be

seen in pre-test scores and post-test scores seen in Table 5.5. We can conclude that changes in layout management burden and visual load do not have a large effect on reading comprehension gains. That being said, participants in the Air condition display a higher variation in scores. Although this difference is small, it could be interpreted as the Airmap interface having a larger sensitivity to differences between the users. In the Kit condition, which requires users to manage their own layout, users are more likely to engage in the same way with the map, thus being less sensitive to user individualities.

To address our fourth research question, whether changes in layout management burden and visual load during the concept map building activity affect the retention of reading comprehension gains, we performed a Mann-Whitney test with delayed normalized change as the dependent variable and condition as the predictor. The test revealed that normalized change for the Kit condition (Mdn = 0.27) was significantly higher than normalized change for the Air condition (Mdn = 0), U = 94, p = 0.03. Looking at the scores on Table 5.5, both conditions answered six questions correctly on average during the pre-test. This goes up to around eight questions on average during the post-test, for both conditions. Then, after two weeks, users in the Air condition went back to answering six questions on average, while users in the Kit condition answer seven questions on average. This can be visualized in Fig. 5.5. Thus, even after two weeks, users from the Kit condition have a greater performance than before building the map. It is important to stress that although users from the Air condition have delayed post-test scores similar to the pre-test scores, the pre-test has been performed after reading a text. Since after two weeks it is expected that they would forget about the content, delayed test scores would be lower than pre-test scores. As such, both conditions are believed to have had gains in retention.

Moving beyond normalized change, Fig. 5.6 gives us better insight on how test scores improved after building the map for each condition. We have examples of users going from answering only one question before building the map to answering seven questions afterward. The score of only one user went down after building the map. Most of the users answered between eight and ten questions in the post-test. Fig. 5.7 gives us insight on how users fared two weeks later when compared to their performance before building the map. A lot of the users did not change in this comparison. Some users in the Kit condition have improved their scores by four questions, while only one user of the Air condition got an improvement of three questions. In the Air condition, we have a drop in performance of as much as four questions. For the Kit condition, performance only drops as much as two questions. Finally, Fig. 5.8 gives us insight on how users fared two weeks later when compared to their scores after building the map. We have abnormal improvements in both conditions. That is, after the two-week retention period, we have users who perform better in the test scores. It cannot be said if this is due to luck

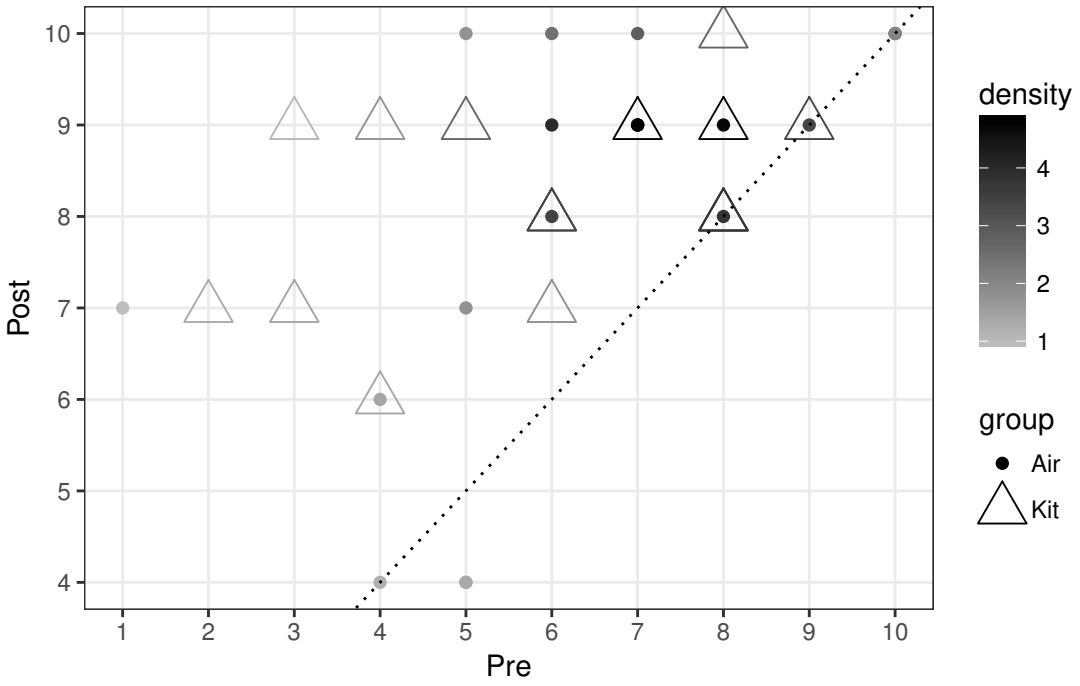


FIGURE 5.6: A scatter plot of pre-test and post-test scores

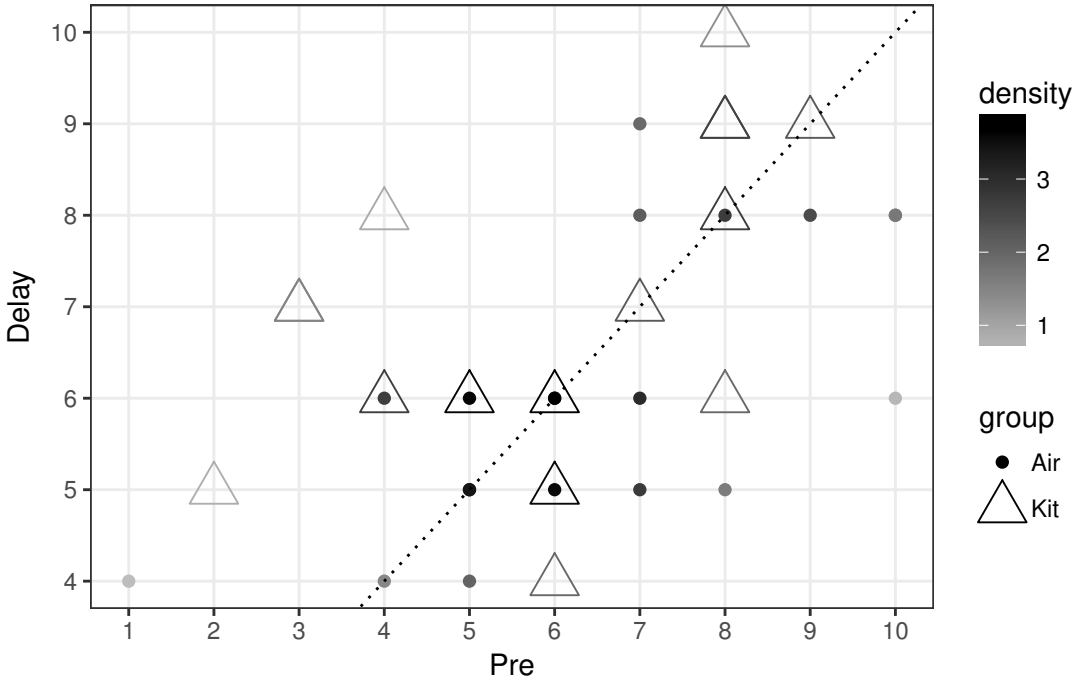


FIGURE 5.7: A scatter plot of pre-test and delayed test scores

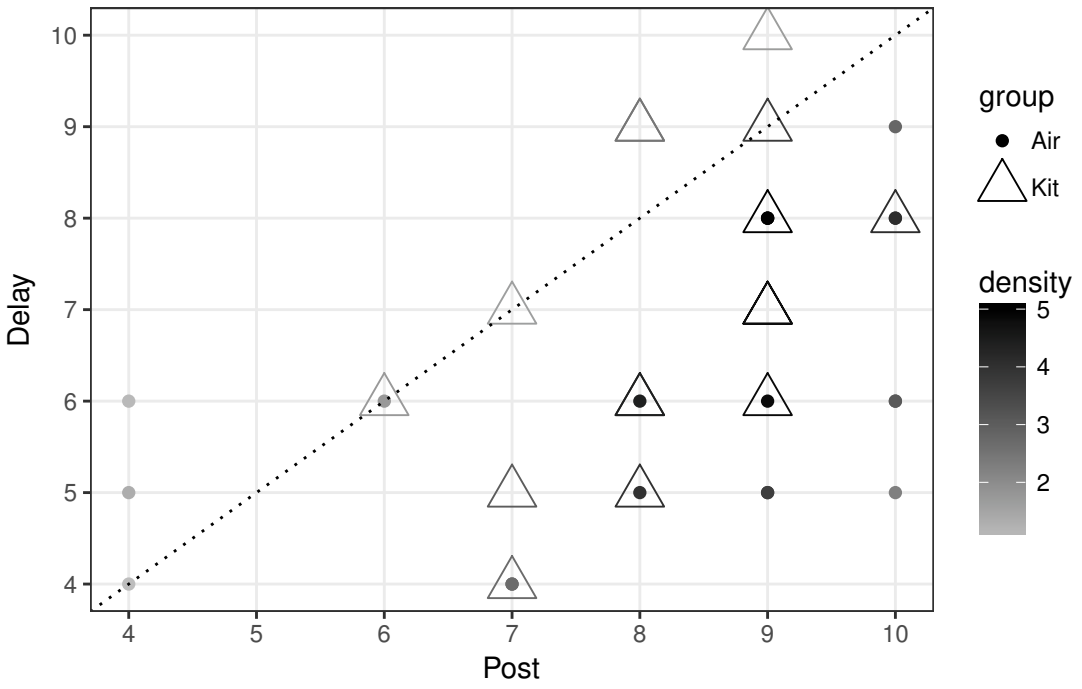


FIGURE 5.8: A scatter plot of post-test and delayed test scores

or if some participants ended up studying the subject by themselves between phases. Once again, users from the Air condition have bigger losses, going up to as much as a five-question difference. Users from the Kit condition, however, have at most a drop of three questions.

Although not a main concern of this experiment, time-on-task, and the number of actions differences between Air and Kit conditions show the same trends as in experiment 1. This indicates that cognitive load is also being reduced in this experiment.

Our main findings for experiment 2 is that a reduction in layout management burden and visual load did not affect immediate reading comprehension but did negatively affect retention of those gains. It is assumed that this occurs because users commit information deeper in memory while trying to organize the layout of the map. We can assume that the cognitive load associated with layout management and the display of information has a germane load component.

5.3.6 General Discussion

Experiment 1 showed that overall cognitive load related to building the map is reduced by using the Airmap interface. This can be associated with a reduction of cognitive load. This indicates that the proposed approach can reduce cognitive load. This suggests that adding layout automation and spatial separation to educational software can provide

benefits linked to reduced cognitive load, such as reducing stress and dissatisfaction[80, 81].

Experiment 2 showed that there was little difference in reading comprehension gains. As such, we can assume that the portion of load reduced is not associated with comprehending the information. Thus, as far as comprehension goes, Airmap shows superior performance when compared to Kit-build, since there is a reduction of cognitive load without reducing comprehension.

This, however, did not hold true for retention. The Kit-build interface outperforms the Air interface after a two week period. As such, we have a trade-off between cognitive load and retention. Thus, the cognitive load differences between the two applications are related to information retention. Past research has shown other cases of association between cognitive load and retention[82]. Literature has also shown other cases where cognitive load reduction diminishes learning effects in general[83]. While it is believed this is mostly caused by the layout management burden factor, there is also the visual load reduction factor. Isolating the factors to see how they affect retention is a matter for future research.

Results further inform teachers who use Kit-build in class, allowing them to choose between the Air interface and the traditional Kit-build interface, depending on the necessity of content retention. The possibility to reduce time-on-task also makes it easier to incorporate computer-based concept mapping on busy classes. Furthermore, developers of other closed concept map building tools can now make a more informed decision about the worth of adding automated layouts and spatial separation to their tools.

The results also suggest that learning applications, in general, could be made easier to use without sacrificing immediate comprehension by using the described approaches when they are applicable and when retention is not of concern. This reduces the mental effort used by learners, allowing them to focus their energies on other activities. It also suggests that a reversed approach can increase retention gains of learning activities. Mixing areas of the interface to eliminate spatial separation and forcing the user to manage the layout would be examples of reversed approaches. Activities in which retention play a strong role, such as vocabulary learning[84] and science classes[85], could benefit from these reversed approaches.

Results also have implications in the comparison between CMA and TMC. The trends shown by the results are similar to those seen in comparisons between CMA and TMC. Kit-build, without automated feedback, performs similarly to traditional creation in reading comprehension but outperforms it in retention[33]. The experiments performed

shed some light on why this happens. Differences in retention are not only because of having to search for the appropriate links and nodes but also because users have to manage the layout of those provided pieces. Having to keep information in the working memory while organizing the layout does little for improving comprehension but does help in committing the comprehended information into long-term memory. That being said, verifying how Airmap fares against traditional map building would give further information into the factors that affect retention in concept map building.

5.4 Kit-build Experiment 2 New and Previous Knowledge Analysis

5.4.1 Method

5.4.1.1 Data Analysis Methods

Each research question needs a quantifiable metric. The metrics can be modeled after answer transitions between a test done before the concept map is built (pre-test), after the concept map is built (post-test), and after a delayed period (delayed post-test). Table 5.6 shows how each question can be classified. Each classification is related to one of the research questions.

"Review" is related to reviewed knowledge. "New" is related to new knowledge. "On-Delay" metrics are related to the two week retention period after building the map. Metrics which don't have the word "OnDelay" on them are related to the immediate measurements. Those immediate measurements are the pre-test and post-test performed minutes before and after building the map.

The questions are classified and then counted for each metric. This gives raw metrics for each user. Based on the raw metrics, normalized metrics are then calculated. The normalized metrics take into account individual ceilings into the calculation of each metric and are more representative of each research question. The normalized metrics and their formulas can be seen in Table 5.7.

5.4.1.2 Results

Normalized values for Review, New, Retained Review, and Retained New were calculated for each participant using their answers for the pre-test, post-test and delayed post-test from the data set. Table 5.9 shows the number of participants of each condition,

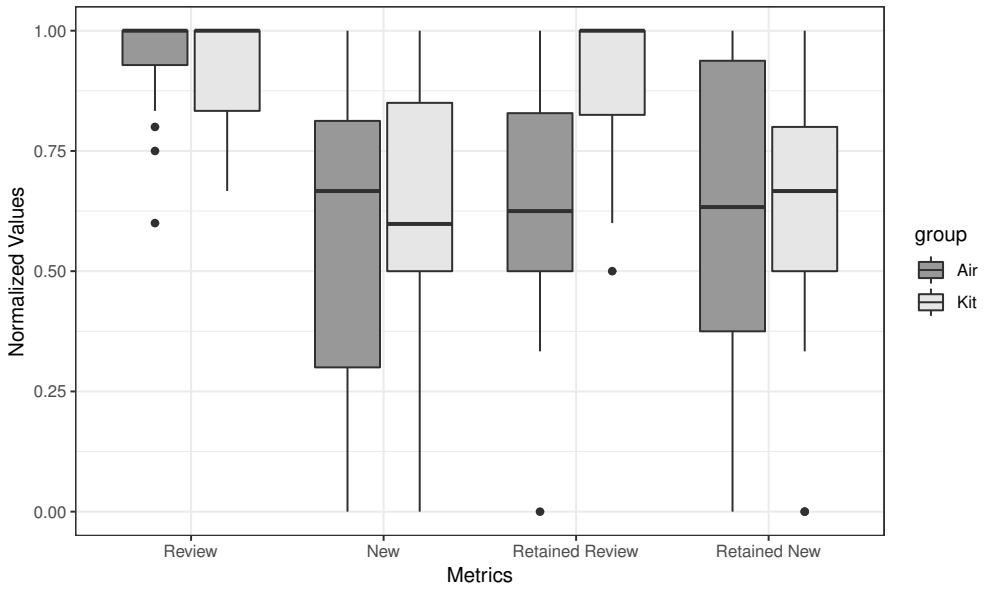


FIGURE 5.9: Boxplots of the normalized values for Air and Kit conditions. Retained Review, which is related to retained reviewed knowledge, represents the biggest difference between the two conditions.

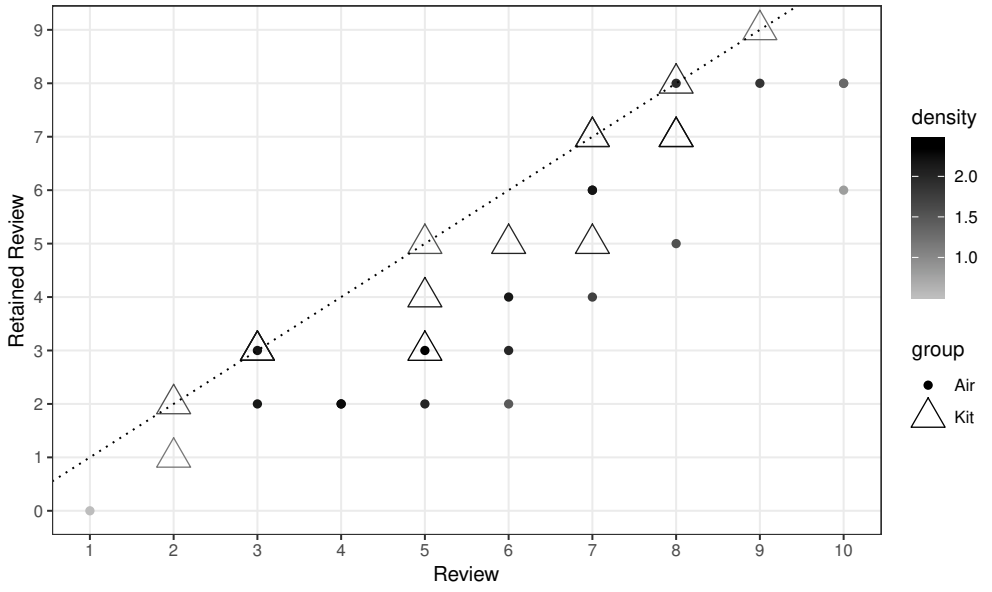


FIGURE 5.10: A scatter plot of Review and Retained Review. Both metrics are related to reviewed knowledge. The farther away from the diagonal line, the more the user forgets. The Kit condition, represented by triangles, is able to retain more after the two-week period when compared to the Air condition

TABLE 5.6: Question classification table

	Answer Correctness		
Classification	*Pre-test*	*Post-test*	*Delayed Post-test*
Review	Correct	Correct	NA
New	Incorrect	Correct	NA
ReviewOnDelay	Correct	Correct	Correct
NewOnDelay	Incorrect	Correct	Correct

TABLE 5.7: Calculated user metrics and their formula. Pre refers to pre-test scores. Review, New, ReviewOnDelay, and NewOnDelay refer to the number of questions belonging to each classification for that particular user.

Metric	*Formula*
Normalized Review	$\frac{Review}{Pre}$
Normalized New	$\frac{New}{1-Pre}$
Normalized Review Retention	$\frac{ReviewOnDelay}{Review}$
Normalized New Retention	$\frac{NewOnDelay}{New}$

TABLE 5.8: The format of the log data for test answers in the experiment

User ID	Test	Question Number	Answer	Correctness
12345	Pre-test	2	4	True
12345	Post-test	3	2	False

TABLE 5.9: Average and standard deviation for the four relevant normalized metrics. Review and Retained Review are related to reviewed knowledge. New and Retained New are related to new knowledge. Kit is the condition which takes influence from the positioning task.

Group	N	Review	New	Retained Review	Retained New
Air	19	0.92 (0.13)	0.66 (0.29)	0.60 (0.27)	0.57 (0.38)
Kit	16	0.89 (0.12)	0.72 (0.21)	0.87 (0.17)	0.60 (0.34)

alongside the average and standard deviation of the relevant normalized metrics. Figure 5.9 shows box plot comparisons of the two conditions.

To address how the positioning task affects immediate retained knowledge, we compare the values of Normalized Review shown in the boxplots of Figure 5.9 and the average values seen in Table 5.9. There is very little difference in Normalized Review between the two interfaces, with users remembering around 90% of their pre-test answers in the post-test.

To address how the positioning task affects immediate new knowledge, we compare the values of Normalized New shown in the boxplots of Figure 5.9 and the average

values seen in Table 5.9. There is very little difference in Normalized New between the two interfaces, with users from both interfaces correctly answering around 70% of the questions in the post-test that they could not answer correctly in the pre-test.

To address how the positioning task affects delayed reviewed knowledge, we performed a Mann-Whitney test with Retained Review as the dependent variable and condition as the predictor. The test revealed that Retained Review for the Kit condition (Mdn = 1) was significantly higher than Retained Review for the Air condition (Mdn = 0.62), U = 58, p = 0.002. Looking at Table 5.9, Airmap users remember around 60% of revised information. In contrast, Kit-build users remember 87% of the revised information. Not only that, but the standard deviation is lower for Kit-build, suggesting results are more stable. Both Air and Kit maintain similar transitions during Map building, but Air drops steeply after the two-week delay, as far as reviewing is concerned. In contrast, Kit shows little loss in reviewed knowledge after the two-week delay.

Looking at the scatter plot in Figure 5.10, multiple Kit-build users forgot none of the test answers related to reviewed information after two weeks. In the worst case scenario, Kit-build users would forget two answers, while Airmap users could forget up to four answers.

To address how the positioning task affects delayed new knowledge, we compare the values of Retained New in the boxplots of Figure 5.9 and the average values seen in Table 5.9. There is very little difference in Normalized Retained New between the two interfaces. This suggests that the two interfaces do not differ in retention as an acquisition tool. Users of both interfaces remember around 60% of the acquired information. This value is similar to the Normalized Retained Review users have during Airmap use. This suggests that users process reviewed information and new information at around the same level while using Airmap.

5.4.2 Discussion

Results show that there was little difference in immediate new knowledge, in the retention of new knowledge, and in immediate reviewed knowledge. As such, we can assume that the differences between the interfaces are not associated with processing new knowledge. Thus, Airmap outperforms Kit-build when new knowledge is of concern since users can make maps using less effort without decreasing immediate and delayed understanding of new knowledge. The reduction in effort, believed to also cause a reduction in cognitive load, is desirable because it has been associated with various benefits, such as reduced stress and higher satisfaction[80, 81]. Results are in line with past research that stated

the positioning task does not affect immediate learning gains[86], but it goes further to also state that it does not affect the retention of new information.

This, however, did not hold true for delayed reviewed knowledge. Kit-build outperforms Airmap in retention of reviewed content after a two week period. As such, we have a trade-off between effort and reviewing retention. Cognitive load reduction leading to a reduction in general retention has been shown in other research as well[82, 83], which is in line with the theory that Airmap has lower cognitive load than Kit-build[86]. Unlike past results, results suggest that this influence on retention is limited to reviewing activities. The reduction is believed to be mostly caused by the layout management burden, but there is also the visual load reduction factor. Isolating these two factors to see how they affect reviewing retention is a matter for future studies.

Results also inform further educators who use closed concept map building tools. Previously it was stated that Kit-build should be used whenever retention is of concern[86]. However, current results suggest that Kit-build should be used as a tool for reviewing. If the user does not have a good grasp of the content, Airmap is better suited since a good portion of information will be new. Furthermore, developers of other closed concept map building tools now have more information when deciding whether or not to add automatic layout management and spatial separation to their tools.

Past research has also pointed out that learning applications, in general, could be made easier to use by applying automatic layout management and spatial separation when retention is not of concern[86]. The same work also pointed out that using a reversed approach could benefit retention gains in learning activities. Adding the positioning task to the activity would be the reversed approach. Current results go further, stating that retention is only prejudiced during reviewing activities, so the amount of activities that could benefit from removing the positioning task is higher than previously thought. However, the reversed approach that was thought to benefit overall retention, is suggested to only influence retention during reviewing. As such, only learning activities which focus on reviewing knowledge should consider this reversed approach. Fields in which retention play a strong role, such as vocabulary learning[84] and science classes[85], could benefit from review activities focused on these reversed approaches.

5.5 Kit-build Experiment 2 Node Oriented Analysis

5.5.1 Coupling Nodes and Questions

In order to carry out the analyses in this study, it is necessary to couple portions of the map with external questions. While it would be simpler to couple concept map propositions with propositions, since propositions represent meaningful information in the concept maps, nodes were chosen instead. This is because, when discussing propositions, it is only possible to discuss if the proposition is present or not. When using nodes, it is possible to take into account if the nodes are used in correct propositions, incorrect propositions or in no propositions. This allows us to argue by using proposition information while also being able to take into account whether the nodes are not used at all. Fig. 5.11 shows examples of the three possible proposition states.

One thing to note is that central nodes which are used in a wide variety of propositions are not included in this analysis. This is because that would add a lot of noise into the analysis. In the material used in this study, a concept map between Komodo dragons is used. In this case, the node ”Komodo dragon” is related to every question, so it is omitted.

Table 5.10 has some examples of questions used in this study alongside the nodes they are coupled with.

5.5.2 Method

Table 5.10 shows two examples of questions included in the test used. Table 5.11 shows some proposition examples from the map. Each proposition is composed of two concepts connected by a link.

5.5.3 Results

The dataset contains entries from 51 users, but only the data related to the 34 users who completed the delayed post-test was used. Out of the 34 users, 18 are Airmap users and 16 are Kit-build users.

612 entries generated by the pairing of nodes and questions were identified in the collected data. The amount of entries divided by proposition classification, interface and delayed test answer can be seen in Table 5.12. The entries distribution can be more easily visualized in Fig. 5.13. Proposition non-existence shows the worse performance

in delayed post-test answers. Surprisingly, proposition correctness and proposition existence does not show much difference in delayed post-test performance. These matters will be further investigated below as the research questions are addressed.

To address our first research question, whether learning performance related to the content represented by closed concept map nodes can be predicted by whether the propositions using the nodes are correct, we performed a Mann-Whitney test with question correctness as the dependent variable and node proposition correctness as the predictor. The test revealed that question correctness for nodes that have correct propositions (Mdn = 1) was significantly higher than question correctness for nodes that did not have correct propositions (Mdn = 1), U = 17.81, p <0.001.

The delayed post-test correctness rate is 0.72 for nodes with correct propositions. In contrast, the delayed post-test correctness rate for nodes with incorrect propositions is 0.56. These ratios have been calculated by adding and dividing the appropriate values in Table 5.12. Given that the users are receiving automated feedback and that the data used is from before the feedback, it makes sense for the difference to not be so high.

To address our second research question, whether learning performance related to the content represented by closed concept map nodes can be predicted by whether propositions using the node exist, we performed a Mann-Whitney test with question correctness as the dependent variable and node proposition existence as the predictor. The test revealed that question correctness for when propositions exist (Mdn = 1) was significantly higher than question correctness for when propositions do not exist (Mdn = 0), U = 15.28, p <0.001.

Delayed post test correctness is 0.67 for nodes that are involved in any proposition. 0.47 is the post test correctness for nodes that were not used in any proposition. Interest to note is comparing proposition existence with proposition correctness, there is not much gap between propositions existing (0.67 correctness) and propositions being correct (0.72 correctness). However, the gap is bigger between propositions not existing (0.47) and not having correct propositions (0.56). This means that not using the node on any propositions has a bigger impact than not making correct propositions. While proposition existence is a stronger predictor, it is rare for users to leave nodes unused in Airmap. So correctness and existence can be used together.

While both research questions have been answered, the interface used and differences between users was not taken into account in the analyses. For that reason, additional exploratory analyses using the interface and the user as covariates were performed. Since the Mann-whitney test cannot take into account covariates, two logistic regressions were performed. The first one took into account proposition correctness using the node, the

TABLE 5.10: Examples of questions in the tests and the nodes they are related to in the concept map.

Question	Choices	Related Nodes
3) Komodo's large size has been attributed to:	Island gigantism and being an excelent predator	Large Size
	Being an excelent predator and being a representative of extinct large varanid lizards	Island Gigantism (...)varanid lizards
	Island gigantism and of extinct large varanid lizards being a representative	
	Being an excelent predator and their venomous bite	
	Island gigantism and being and their venomous bite	
7) About their (Komodo Dragons) reproduction	They mate in September	Between May and August
	They mate in October	
	They mate between September and December	
	They mate between May and August	
	They mate in December	

interface used and the users. The second one took into account proposition existence using the node, the interface used and the users.

Information on the first regression can be seen in Table 5.13. Correct propositions using the node is a significant predictor. Interface is not a significant predictor. Some of the users, mainly those who performed well, have been identified as significant predictors. This means that some users perform well regardless of proposition correctness.

Information on the second regression can be seen in Table 5.14. Proposition existence using the node is a significant predictor. Like the previous analysis, interface is not a significant predictor but some of the users are identified as significant predictors.

The two analyses suggest that the effect of proposition correctness and proposition existence remains even after controlling for the interface and the users.

Results show that analytics using closed concept maps can be used to understand how much students learn specific parts of the content. This goes beyond previous results which focused only on the correlation between test scores and map scores[45].

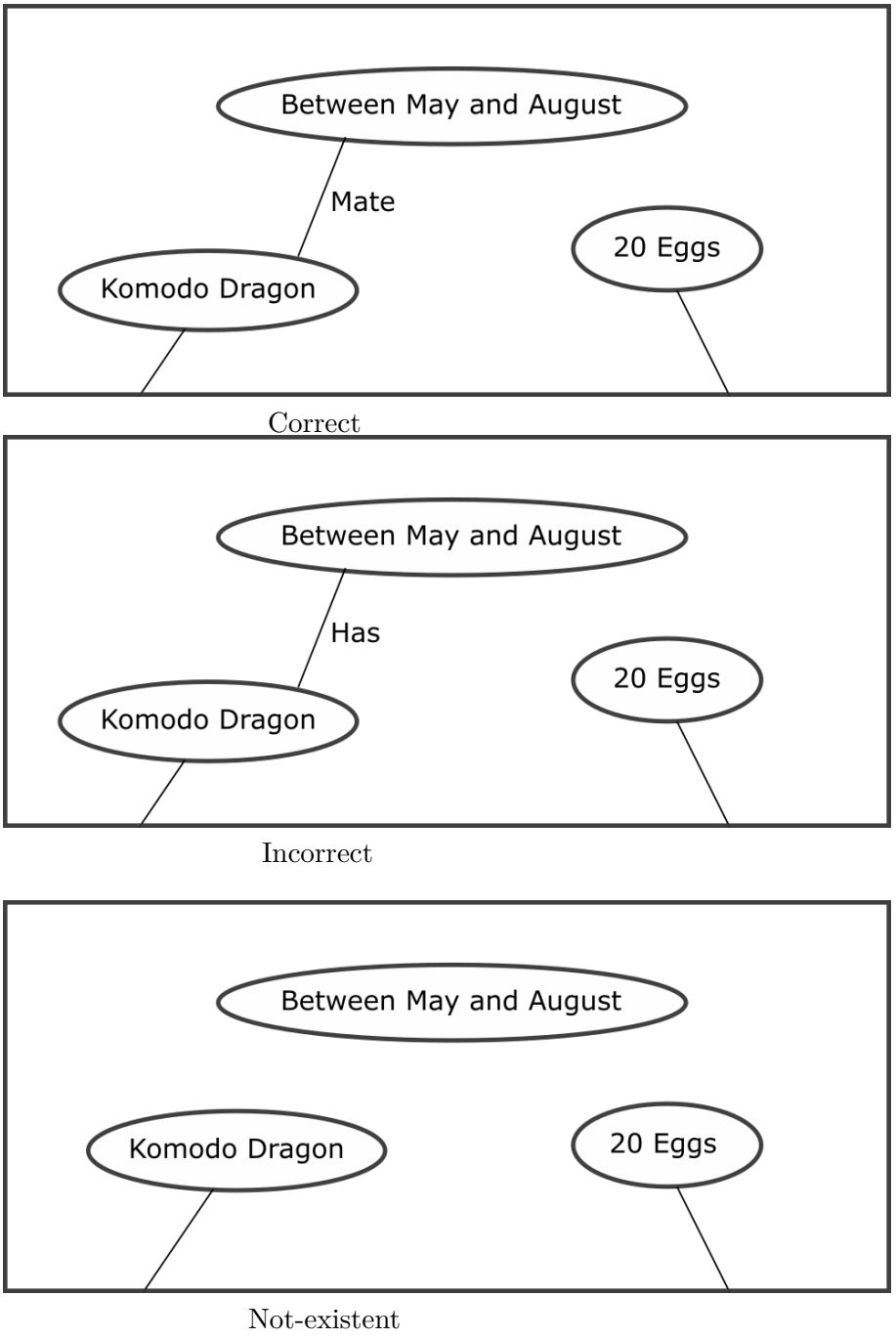


FIGURE 5.11: Examples of the three possible proposition states for node "Between May and August"

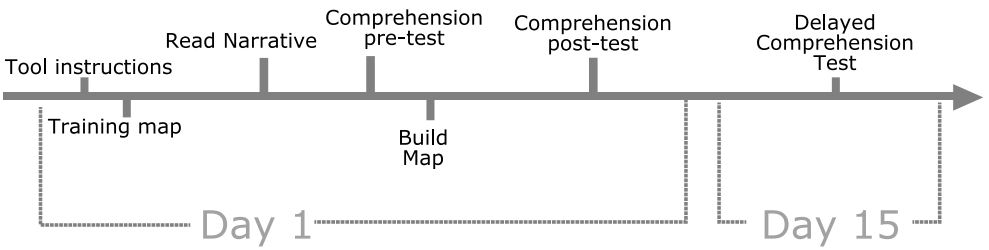


FIGURE 5.12: Timeline for the Experiment.

Table 5.11: E

Concept 1	Link	Concept 2
komodo dragon	has	large size
large size	attributed to	Island gigantism
large size	attributed to	representative of extinct large varanid lizards
komodo dragon	mates	between May and August

Table 5.12: Amount of data entries for each node classification based on the propositions that use the node. Entries are further divided based on the interface used and based on whether or not they answered the corresponding question correctly on the delayed post test.

Node	Interface	Test	Amount
Incorrect	Airmap	Incorrect	58
		Correct	63
	Kit-build	Incorrect	61
		Correct	85
Correct	Airmap	Incorrect	64
		Correct	139
	Kit-build	Incorrect	33
		Correct	109
Not exist	Airmap	Incorrect	17
		Correct	11
	Kit-build	Incorrect	32
		Correct	32
Exist	Airmap	Incorrect	105
		Correct	191
	Kit-build	Incorrect	62
		Correct	162

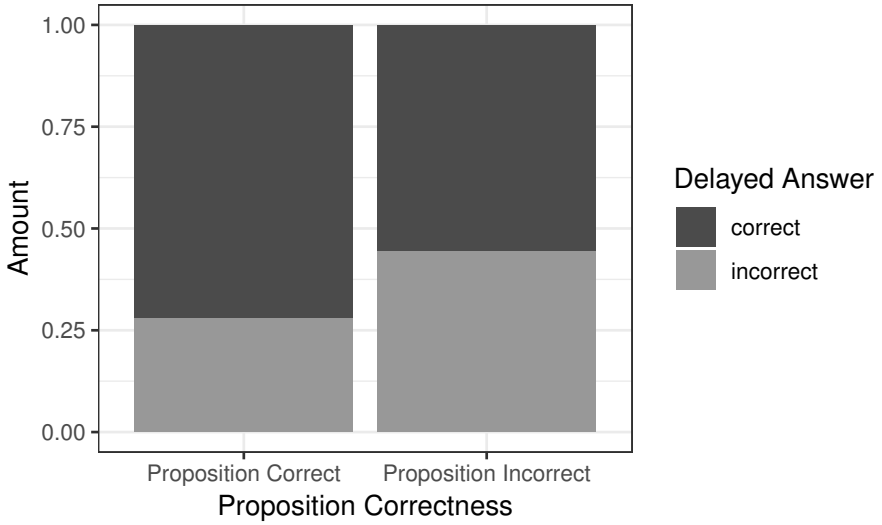


Figure 5.13: Bar graph showing the relationship between proposition correctness and the answer to the related questions on the delayed post test. Proposition correctness is associated with correct answers on the delayed test.

TABLE 5.13: P values and odds ratio for the logistic regression related to proposition correctness using the nodes. HV signifies high values that were omitted, for one user that had perfect scores.

	OR	2.5 %	97.5 %	p
Correctness	1.86	1.25	2.78	<0.001
Interface	1.62	0.42	6.38	0.49
user2	0.70	0.18	2.65	0.60
user3	1.02	0.26	3.87	0.98
user4	1.15	0.29	4.70	0.84
user5	1.15	0.29	4.70	0.84
user7	0.88	0.23	3.34	0.86
user9	3.60	0.82	19.77	0.11
user11	0.78	0.20	2.96	0.72
user18	0.90	0.23	3.44	0.88
user19	0.95	0.25	3.59	0.94
user20	5.74	1.12	44.09	0.05
user22	4.54	0.88	34.86	0.09
user23	2.52	0.55	14.02	0.25
user24	7.33	1.46	56.12	0.03
user25	3.60	0.82	19.77	0.11
user26	0.36	0.09	1.36	0.14
user27	1.67	0.41	7.22	0.48
user29	0.93	0.24	3.67	0.92
user30	HV	HV	HV	0.98
user31	0.85	0.22	3.26	0.81
user32	0.54	0.13	2.08	0.38
user34	2.03	0.50	8.78	0.33
user37	1.45	0.38	5.70	0.58
user39	0.64	0.16	2.43	0.52
user40	0.76	0.20	2.94	0.69
user41	1.40	0.34	6.06	0.64
user42	2.07	0.53	8.47	0.30
user43	4.39	0.99	24.25	0.06
user44	0.72	0.18	2.75	0.64
user45	5.25	1.17	29.21	0.04
user47	3.53	0.84	16.74	0.96
user48	1.04	0.27	3.94	0.49
user49	1.61	0.42	6.50	0.56
Constant term	0.75	0.28	2.07	0.56

TABLE 5.14: P values and odds ratio for the logistic regression related to proposition existence using the nodes. HV signifies high values that were omitted, for one user that had perfect scores.

	OR	2.5 %	97.5 %	p
Existence	2.11	1.11	4.09	0.02
Interface	1.26	0.33	4.85	0.74
user2	0.64	0.17	2.37	0.51
user3	0.80	0.21	2.98	0.74
user4	1.27	0.32	5.12	0.73
user5	1.27	0.32	5.12	0.73
user7	0.80	0.21	2.98	0.74
user9	4.00	0.91	21.86	0.08
user11	0.78	0.20	2.96	0.72
user18	0.78	0.20	2.96	0.72
user19	0.83	0.22	3.11	0.78
user20	6.56	1.26	51.05	0.04
user22	5.37	1.06	41.03	0.06
user23	3.18	0.71	17.47	0.15
user24	6.40	1.29	48.50	0.04
user25	4.00	0.91	21.86	0.08
user26	0.44	0.11	1.64	0.23
user27	1.65	0.41	7.05	0.48
user29	1.00	0.26	3.88	1.00
user30	HV	HV	HV	0.98
user31	1.32	0.32	5.48	0.70
user32	0.54	0.13	2.13	0.39
user34	2.59	0.60	11.83	0.21
user37	1.26	0.33	4.85	0.74
user39	0.99	0.23	4.20	0.99
user40	0.98	0.25	3.81	0.98
user41	1.65	0.41	7.05	0.48
user42	1.60	0.42	6.41	0.50
user43	4.00	0.91	21.86	0.08
user44	1.10	0.26	4.52	0.90
user45	4.65	1.05	25.78	0.05
user47	2.80	0.68	13.03	0.80
user48	1.18	0.31	4.54	0.42
user49	1.75	0.45	7.07	0.36
Constant Term	0.59	0.19	1.87	0.36

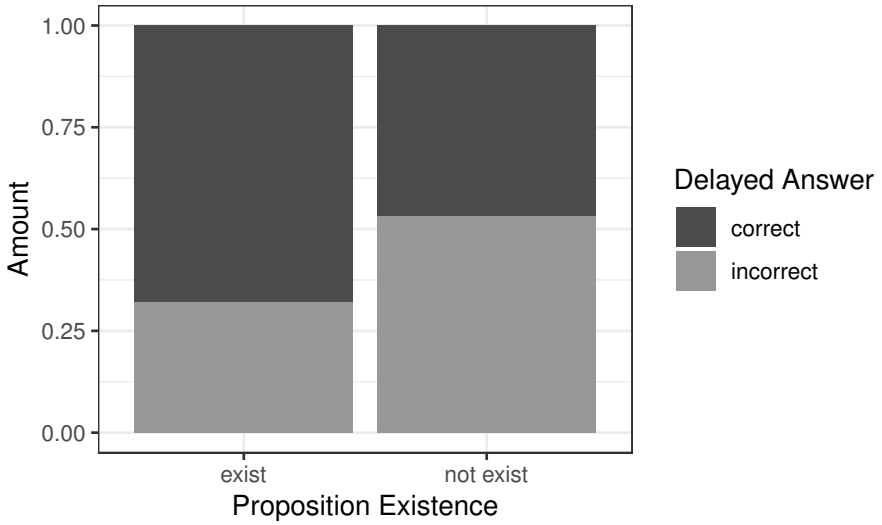


FIGURE 5.14: Bar graph showing the relationship between proposition existence and the answer to the related questions on the delayed post test. Proposition existence is associated with correct answers on the delayed test.

Chapter 6

Conclusion

6.1 Monsakun

In this study, the design of an application for teaching the conceptual understanding of contextual problems for Kindergarten students was presented. Furthermore, the viability of the application was verified through use in a Kindergarten classroom. The interaction of users with the application was successful in many respects. Students showed good affective responses and satisfactory performance, outperforming the GTS calculated measures. The results suggest the following:

1. The application is successful in allowing for meaningful interaction with the Triplet Structure Model without the use of text and arithmetic expressions;
2. Students interactions with the Triplet Structure Model improve while using the application, quickly clearing problems which had given them trouble in the past.

Their growth patterns and the distance to gaming the system patterns suggest that the interactions of the students with the software are meaningful. This indicates that the Triplet Structure Model is effective for younger children too. The results strengthen the software position as a tool to help kindergarten students to understand contextual problems conceptually.

The design and viability verification of the application, alongside the data analysis suggesting meaningful use and growth inside the application are the main contributions of this study.

The interactions of the students get better as they clear problems. As such, improvement inside the application has been verified. The lack of evaluations from outside the system,

such as using a control group, pre-test, and post-test makes it hard to verify how well this transfer to other contexts. This is one issue to be tackled in future studies.

The case-by-case results suggest that students who struggled during some critical levels are also engaging and improving their understanding, they just took longer than other students to overcome hard challenges for the first time.

Looking at the average values, students perform well in the early levels. However, some of the students may be leaving Level 1 and Level 2 without a clear interpretation of the pictures. Adding user modeling and procedural question generation to the application is one approach to make sure that students are leaving the earlier levels with the desired interpretation of the pictures. Furthermore, increasing the number of easier problems in Level 3 problems might make it easier for students to clear the harder levels.

The next step in this research is to allow students to create the story pieces themselves. The objective would be to make children think deeply about each quantity separately.

6.2 Kit-build

Software-based concept map assembling from provided pieces is an effective learning method. Yet, no cognitive load studies of this approach have been found in the literature, despite the benefits of cognitive load optimization. This study attempted to lower the cognitive load associated with the activity by developing an interface with automatic layout management and spatial separation of elements. Results show that the developed interface was successful in reducing cognitive load. Furthermore, no significant reduction of gains in immediate comprehension was found. However, there was a significant drop in performance after a two week retention period. Thus, having to search around for the pieces in a complex space and having to manage the layout during concept map creation help students commit the information deeper into memory.

One limitation of the present study is that there was no measure of gains in comprehension and retention without the automatic feedback feature. Another limitation is that the two factors related to cognitive load, layout management and spatial separation, were not isolated.

In future work, new interfaces that isolate layout management and spatial separation will be developed. The challenge of developing these interfaces lies in introducing these elements without changing other aspects of the interface. Another matter for future study is analyzing differences in learning gains without the use of feedback mechanisms.

Finally, whether or not the time saved by reducing the cognitive load could be better spent in other activities, such as having students make associations between the text and the map, will also be investigated. This could provide further gains in both comprehension and retention.

6.3 Overall Conclusion

The open information structure approach has shown promising results in the design of learning applications. However, the use of these applications in different contexts is a problem that is not only limited to the open information structure. However, there is the potential for applications built with the open information structure to be adapted to new contexts without changing the information structure. This simplifies the redesigning process since the information structure can be reused. To the best of our knowledge, this has not been studied before. This study examines two use cases where applications built with the open information structure approach have been redesigned while maintaining the information structure. It uses these two use cases and multiple experiments to address the three research questions.

The first research question, ”Can applications built using the open information structure approach be redesigned while keeping the information structure imtact?”, is answered in both use cases. Both applications were redesigned and are usable while keeping the information structure intact.

The second research question, ”Do the redesigned applications fit their new context appropriately?”, has also been answered in both case studies. In case of the first use case, the redesigned Monsakun was successfully used by the Kindergarten students, which showed growing patterns and were not gaming the system. In the second use case, Kit-build, the redesigned version of Airmap managed to reduce the overall effort related to building the map.

The third research question, ”How does the redesign process affect learning gains in the new context?”, could not be answered in the first study case but was answered in the second one. The reason why it could not be answered is because the Kindergarten students cannot use the original Monsakun, so it is not possible to compare the learning gains. In the case of Kit-build, the redesigned version was compared to the original version and immediate learning comprehension was not affected. However, since long term results were affected, this shows that both versions of Kit-build are useful in the classroom and should be used based on the context.

This study shows new advantages of the open information structure approach. It informs designers that they can keep the underlying information approach when redesigning the applications. It gave an example of adapting an application from text to images. It also gives information on which closed concept map tools are better used for each context.

Bibliography

[1] T. Hirashima and Y. Hayashi, "Design of meta-problem with open information structure approach," 12 2018.

[2] T. Hirashima and Y. Hayashi, "Educational externalization of thinking task by kit-build method," in *International Conference on Human Interface and the Management of Information*, pp. 126–137, Springer, 2016.

[3] D. Pitt, "Mental representation," 2000.

[4] S. Vosniadou, "Capturing and modeling the process of conceptual change," *Learning and instruction*, vol. 4, no. 1, pp. 45–69, 1994.

[5] W. J. Clancey, *Situated cognition: On human knowledge and computer representations*. Cambridge university press, 1997.

[6] T. Hirashima, S. Yamamoto, and Y. Hayashi, "Triplet structure model of arithmetical word problems for learning by problem-posing," in *International Conference on Human Interface and the Management of Information*, pp. 42–50, Springer, 2014.

[7] A. A. Supianto, Y. Hayashi, and T. Hirashima, "An investigation of learner's actions in problem-posing activity of arithmetic word problems," in *Intelligent Tutoring Systems: 13th International Conference, ITS 2016, Zagreb, Croatia, June 7-10, 2016. Proceedings*, vol. 9684, p. 478, Springer, 2016.

[8] S. Yamamoto, T. Kanbe, Y. Yoshida, K. Maeda, and T. Hirashima, "A case study of learning by problem-posing in introductory phase of arithmetic word problems," in *Proceedings of the 20th International Conference on Computers in Education*, pp. 25–32, 2012.

[9] S. Yamamoto, Y. Akao, M. Murotsu, T. Kanbe, Y. Yoshida, K. Maeda, Y. Hayashi, and T. Hirashima, "Interactive environment for learning by problem-posing of arithmetic word problems solved by one-step multiplication and division," *ICCE2014*, pp. 89–94, 2014.

[10] A. Leff and J. T. Rayfield, "Web-application development using the model/view/-controller design pattern," in *Proceedings fifth ieee international enterprise distributed object computing conference*, pp. 118–127, IEEE, 2001.

[11] N. C. Jordan, "Early predictors of mathematics achievement and mathematics learning difficulties," *Encyclopedia on Early Childhood Development*, pp. 1–6, 2010.

[12] S. Griffin, "Building number sense with number worlds: A mathematics program for young children," *Early childhood research quarterly*, vol. 19, no. 1, pp. 173–180, 2004.

[13] N. I. Dyson, N. C. Jordan, and J. Glutting, "A number sense intervention for low-income kindergartners at risk for mathematics difficulties," *Journal of Learning Disabilities*, vol. 46, no. 2, pp. 166–181, 2013.

[14] P. Starkey, A. Klein, and A. Wakeley, "Enhancing young children' s mathematical knowledge through a pre-kindergarten mathematics intervention," *Early Childhood Research Quarterly*, vol. 19, no. 1, pp. 99–120, 2004.

[15] A. J. Wilson, S. Dehaene, O. Dubois, and M. Fayol, "Effects of an adaptive game intervention on accessing number sense in low-socioeconomic-status kindergarten children," *Mind, Brain, and Education*, vol. 3, no. 4, pp. 224–234, 2009.

[16] D. H. Clements and J. Sarama, "Effects of a preschool mathematics curriculum: Summative research on the building blocks project," *Journal for Research in Mathematics Education*, pp. 136–163, 2007.

[17] S. Kim, "Computer-assisted mathematical interventions on word problems for elementary students with underachievement in mathematics," 2017.

[18] V. Freiman, E. Polotskaia, and A. Savard, "Using a computer-based learning task to promote work on mathematical relationships in the context of word problems in early grades," *ZDM*, vol. 49, no. 6, pp. 835–849, 2017.

[19] M. Yerushalmy, "Slower algebra students meet faster tools: Solving algebra word problems with graphing software," *Journal for Research in Mathematics Education*, pp. 356–387, 2006.

[20] K. R. Koedinger and J. R. Anderson, "Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization," *Interactive Learning Environments*, vol. 5, no. 1, pp. 161–179, 1998.

[21] B. M. Hamadneh, H. A. Hamad, and M. M. Al-azzam, "The impact of applying computer-based training strategy upon developing the skill of solving mathematical

word problem among students with learning disabilities," *International Research in Education*, vol. 4, no. 1, pp. 149–158, 2016.

[22] T.-H. Huang, Y.-C. Liu, and H.-C. Chang, "Learning achievement in solving word-based mathematical questions through a computer-assisted learning system.," *Educational Technology & Society*, vol. 15, no. 1, pp. 248–259, 2012.

[23] J. Sweller, "Cognitive load theory, learning difficulty, and instructional design," *Learning and instruction*, vol. 4, no. 4, pp. 295–312, 1994.

[24] J. Leppink, F. Paas, T. Van Gog, C. P. van Der Vleuten, and J. J. Van Merrienboer, "Effects of pairs of problems and examples on task performance and different types of cognitive load," *Learning and Instruction*, vol. 30, pp. 32–42, 2014.

[25] T. Grunwald and C. Corsbie-Massay, "Guidelines for cognitively efficient multimedia learning tools: educational strategies, cognitive load, and interface design," *Academic medicine*, vol. 81, no. 3, pp. 213–223, 2006.

[26] X. P. Kotval and J. H. Goldberg, "Eye movements and interface component grouping: An evaluation method," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 42, pp. 486–490, SAGE Publications Sage CA: Los Angeles, CA, 1998.

[27] W. Schnotz, S. Fries, and H. Horz, "Motivational aspects of cognitive load theory," *Contemporary motivation research: From global to local perspectives*, pp. 69–96, 2009.

[28] G.-J. Hwang, L.-H. Yang, and S.-Y. Wang, "A concept map-embedded educational computer game for improving students' learning performance in natural science courses," *Computers & Education*, vol. 69, pp. 121–130, 2013.

[29] P. Kim and C. Olaciregui, "The effects of a concept map-based information display in an electronic portfolio system on information processing and retention in a fifth-grade science class covering the earth's atmosphere," *British Journal of Educational Technology*, vol. 39, no. 4, pp. 700–714, 2008.

[30] M. Willerman and R. A. Mac Harg, "The concept map as an advance organizer," *Journal of research in science teaching*, vol. 28, no. 8, pp. 705–711, 1991.

[31] E. Morfidi, A. Mikropoulos, and A. Rogdaki, "Using concept mapping to improve poor readers' understanding of expository text," *Education and Information Technologies*, vol. 23, no. 1, pp. 271–286, 2018.

[32] M. Omar, "Improving reading comprehension by using computer-based concept maps: A case study of esp students at umm-alqura university," *British Journal of Education*, vol. 3, no. 4, pp. 1–20, 2015.

[33] M. Alkhateeb, Y. Hayashi, T. Rajab, and T. Hirashima, "Comparison between kit-build and scratch-build concept mapping methods in supporting efl reading comprehension," *The Journal of Information and Systems in Education*, vol. 14, no. 1, pp. 13–27, 2015.

[34] P.-L. Liu, C.-J. Chen, and Y.-J. Chang, "Effects of a computer-assisted concept mapping learning strategy on efl college students' english reading comprehension," *Computers & Education*, vol. 54, no. 2, pp. 436–445, 2010.

[35] W. Reader and N. Hammond, "Computer-based tools to support learning from hypertext: concept mapping tools and beyond," in *Computer Assisted Learning: Selected Contributions from the CAL'93 Symposium*, pp. 99–106, Elsevier, 1994.

[36] L. Anderson-Inman and L. Zeitz, "Computer-based concept mapping: Active studying for active learners," *The computing teacher*, vol. 21, no. 1, 1993.

[37] U. Park and R. A. Calvo, "Automatic concept map scoring framework using the semantic web technologies," in *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pp. 238–240, IEEE, 2008.

[38] E. M. Taricani and R. B. Clariana, "A technique for automatically scoring open-ended concept maps," *Educational Technology Research and Development*, vol. 54, no. 1, pp. 65–82, 2006.

[39] A. J. Cañas, L. Bunch, J. D. Novak, and P. Reiska, "Cmapanalysis: An extensible concept map analysis tool," *Journal for Educators, Teachers and Trainers*, 2013.

[40] T. Hirashima, K. Yamasaki, H. Fukuda, and H. Funaoi, "Kit-build concept map for automatic diagnosis," in *International conference on artificial intelligence in education*, pp. 466–468, Springer, 2011.

[41] T. Hirashima, K. Yamasaki, H. Fukuda, and H. Funaoi, "Framework of kit-build concept map for automatic diagnosis and its preliminary use," *Research and Practice in Technology Enhanced Learning*, vol. 10, no. 1, p. 17, 2015.

[42] H. Herl, H. O'Neil Jr, G. Chung, and J. Schacter, "Reliability and validity of a computer-based knowledge mapping system to measure content understanding," *Computers in Human Behavior*, vol. 15, no. 3-4, pp. 315–333, 1999.

[43] C. Tao, *Development of a Knowledge Assessment System Based on Concept Maps and Differential Weighting Approaches*. PhD thesis, Virginia Tech, 2015.

[44] P.-H. Wu, G.-J. Hwang, M. Milrad, H.-R. Ke, and Y.-M. Huang, "An innovative concept map approach for improving students' learning performance with an instant feedback mechanism," *British Journal of Educational Technology*, vol. 43, no. 2, pp. 217–232, 2012.

[45] K. Yoshida, K. Sugihara, Y. Nino, M. Shida, and T. Hirashima, "Practical use of kit-build concept map system for formative assessment of learners' comprehension in a lecture," *Proc. of ICCE2013*, pp. 892–901, 2013.

[46] W. Wunnasri, J. Pailai, Y. Hayashi, and T. Hirashima, "Validity of kit-build method for assessment of learner-build map by comparing with manual methods," *IEICE Transactions on Information and Systems*, vol. 101, no. 4, pp. 1141–1150, 2018.

[47] W. Wunnasri, J. Pailai, Y. Hayashi, and T. Hirashima, "Reliability investigation of automatic assessment of learner-build concept map with kit-build method by comparing with manual methods," in *International Conference on Artificial Intelligence in Education*, pp. 418–429, Springer, 2017.

[48] P. Reiska, K. Soika, and A. J. Cañas, "Using concept mapping to measure changes in interdisciplinary learning during high school," *Knowledge Management & E-Learning: An International Journal (KM&EL)*, vol. 10, no. 1, pp. 1–24, 2018.

[49] K. Sugihara, T. Osada, S. Nakata, H. Funaoi, and T. Hirashima, "Experimental evaluation of kit-build concept map for science classes in an elementary school," *Proc. ICCE2012*, pp. 17–24, 2012.

[50] J. Sabourin, J. P. Rowe, B. W. Mott, and J. C. Lester, "When off-task is on-task: The affective role of off-task behavior in narrative-centered learning environments," in *International Conference on Artificial Intelligence in Education*, pp. 534–536, Springer, 2011.

[51] R. S. Baker, A. T. Corbett, and K. R. Koedinger, "Detecting student misuse of intelligent tutoring systems," in *International conference on intelligent tutoring systems*, pp. 531–540, Springer, 2004.

[52] J. E. Beck, K.-m. Chang, J. Mostow, and A. Corbett, "Does help help? introducing the bayesian evaluation and assessment methodology," in *International Conference on Intelligent Tutoring Systems*, pp. 383–394, Springer, 2008.

[53] A. J. Bowers, "Analyzing the longitudinal k-12 grading histories of entire cohorts of students: Grades, data driven decision making, dropping out and hierarchical cluster analysis," *Practical Assessment Research and Evaluation*, vol. 15, no. 7, pp. 1–18, 2010.

[54] N. Ming and V. Ming, "Predicting student outcomes from unstructured data.," in *UMAP Workshops*, 2012.

[55] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning analytics*, pp. 61–75, Springer, 2014.

[56] D. Ben-Naim, M. Bain, and N. Marcus, "A user-driven and data-driven approach for supporting teachers in reflection and adaptation of adaptive tutorials.," *International Working Group on Educational Data Mining*, 2009.

[57] R. S. Baker, A. de Carvalho, J. Raspat, V. Aleven, A. T. Corbett, and K. R. Koedinger, "Educational software features that encourage and discourage "gaming the system"," in *Proceedings of the 14th international conference on artificial intelligence in education*, pp. 475–482, 2009.

[58] R. Martinez-Maldonado, A. Clayphan, K. Yacef, and J. Kay, "Mtfeedback: providing notifications to enhance teacher awareness of small group work in the classroom," *IEEE Transactions on Learning Technologies*, vol. 8, no. 2, pp. 187–200, 2015.

[59] M. Schmidt and A. A. Tawfik, "Using analytics to transform a problem-based case library: An educational design research approach," *Interdisciplinary Journal of Problem-Based Learning*, vol. 12, no. 1, p. 5, 2017.

[60] A. Grubišić, S. Stankov, B. Žitko, I. Šarić, S. Tomaš, E. Brajković, T. Volarić, D. Vasić, and A. Dodaj, "Knowledge tracking variables in intelligent tutoring systems," in *Proceedings of the 9th International Conference on Computer Supported Education, CSEDU*, vol. 1, pp. 513–518, 2017.

[61] Y.-S. Lin, Y.-C. Chang, K.-H. Liew, and C.-P. Chu, "Effects of concept map extraction and a test-based diagnostic environment on learning achievement and learners' perceptions," *British Journal of Educational Technology*, vol. 47, no. 4, pp. 649–664, 2016.

[62] K. Leelawong and G. Biswas, "Designing learning by teaching agents: The betty's brain system," *International Journal of Artificial Intelligence in Education*, vol. 18, no. 3, pp. 181–208, 2008.

[63] M. Emara, M. Tscholl, Y. Dong, and G. Biswas, "Analyzing students' collaborative regulation behaviors in a classroom-integrated open ended learning environment," in *Making a Difference: Prioritizing Equity and Access in CSCL, 12th International Conference on Computer Supported Collaborative Learning (CSCL)*, Philadelphia, PA: International Society of the Learning Sciences., 2017.

[64] H. Jeong, A. Gupta, R. Roscoe, J. Wagster, G. Biswas, and D. Schwartz, "Using hidden markov models to characterize student behaviors in learning-by-teaching environments," in *International conference on intelligent tutoring systems*, pp. 614–625, Springer, 2008.

[65] Y. Dong and G. Biswas, "An extended learner modeling method to assess students' learning behaviors.," in *EDM*, 2017.

[66] G. P. Jain, V. P. Gurupur, J. L. Schroeder, and E. D. Faulkenberry, "Artificial intelligence-based student learning evaluation: a concept map-based approach for analyzing a student's understanding of a topic," *IEEE Transactions on Learning Technologies*, vol. 7, no. 3, pp. 267–279, 2014.

[67] V. P. Gurupur, G. P. Jain, and R. Rudraraju, "Evaluating student learning using concept maps and markov chains," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3306–3314, 2015.

[68] C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.

[69] R. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger, "Why students engage in" gaming the system" behavior in interactive learning environments," *Journal of Interactive Learning Research*, vol. 19, no. 2, p. 185, 2008.

[70] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[71] W. G. La, "juhgiyo/epforcedirectedgraph.cs," Mar 2017.

[72] W. Hann, "Julie's race," Jan 2001.

[73] F. G. Paas and J. J. Van Merriënboer, "Instructional control of cognitive load in the training of complex cognitive tasks," *Educational psychology review*, vol. 6, no. 4, pp. 351–371, 1994.

[74] R. Brunken, J. L. Plass, and D. Leutner, "Direct measurement of cognitive load in multimedia learning," *Educational psychologist*, vol. 38, no. 1, pp. 53–61, 2003.

[75] R. Brünken and D. Leutner, "Aufmerksamkeitsverteilung oder aufmerksamkeitsfokussierung? empirische ergebnisse zur „split-attention-hypothese "beim lernen mit multimedia," *Unterrichtswissenschaft*, vol. 29, no. 4, pp. 357–366, 2001.

[76] H. Astleitner and D. Leutner, "Applying standard network analysis to hypermedia systems: Implications for learning," *Journal of Educational Computing Research*, vol. 14, no. 3, pp. 285–303, 1996.

[77] S. Engeser and F. Rheinberg, "Flow, performance and moderators of challenge-skill balance," *Motivation and Emotion*, vol. 32, no. 3, pp. 158–172, 2008.

[78] A. R. Ellis, W. W. Burchett, S. W. Harrar, and A. C. Bathke, "Nonparametric inference for multivariate data: the r package npmv," *J. Stat. Softw*, vol. 76, no. 4, pp. 1–18, 2017.

[79] J. D. Marx and K. Cummings, "Normalized change," *American Journal of Physics*, vol. 75, no. 1, pp. 87–91, 2007.

[80] S. Zhang, L. Zhao, Y. Lu, and J. Yang, "Get tired of socializing as social animal? an empirical explanation on discontinuous usage behavior in social network services.," in *PACIS*, p. 125, 2015.

[81] A. Ward and T. Mann, "Don't mind if i do: Disinhibited eating under cognitive load.," *Journal of personality and social psychology*, vol. 78, no. 4, p. 753, 2000.

[82] F. Kirschner, F. Paas, and P. A. Kirschner, "Individual and group-based learning from complex cognitive tasks: Effects on retention and transfer efficiency," *Computers in Human Behavior*, vol. 25, no. 2, pp. 306–314, 2009.

[83] W. Schnotz and T. Rasch, "Enabling, facilitating, and inhibiting effects of animations in multimedia learning: Why reduction of cognitive load can have negative results on learning," *Educational Technology Research and Development*, vol. 53, no. 3, p. 47, 2005.

[84] K. S. Folse, "The effect of type of written exercise on l2 vocabulary retention," *TESOL quarterly*, vol. 40, no. 2, pp. 273–293, 2006.

[85] D. H. O'day, "The value of animations in biology teaching: a study of long-term memory retention," *CBE-Life Sciences Education*, vol. 6, no. 3, pp. 217–223, 2007.

[86] P. G. F. Furtado, T. Hirashima, and Y. Hayashi, "Reducing cognitive load during closed concept map construction and consequences on reading comprehension and retention," *IEEE Transactions on Learning Technologies*, 2018.