ABSTRACT
Motivation – In this paper we discuss the beginning of a research project into the design of user interfaces for Computer Algebra Systems (CASs). Research approach – Findings/Design – We begin to examine a theoretical framework describing how people learn problem-solving skills. We also briefly discuss a field study we conducted. Research limitations/Implications – The field study, while small and preliminary, allowed us to see some areas where students ran into difficulty using Maple. The theoretical framework we begin to outline should help us understand the cause of some of those difficulties. Originality/Value – We begin to better understand how people use CASs to learn mathematics. We also begin to combine Cognitive Load Theory and the Skills-Rules-Knowledge framework. Take away message – This is the beginning of a research project that should help us better understand how CASs can be better designed to assist with learning and the problem-solving process.

Keywords
computer algebra system, user interface, cognitive load, skills-rules-knowledge

INTRODUCTION
Mathematical software has become commonplace in industry and academia. Such software helps its users solve a wide range of mathematical problems. In this paper we are going to develop a framework for studying the user interfaces for such software with the eventual goal of making significant improvements to those interfaces. The focus of this paper will be on the use of Computer Algebra Systems (CASs) in classroom settings but it is hoped that the results will be of use in industrial settings and for more advanced users.

We begin by looking at CASs and their current use in educational institutions, including a brief discussion of a preliminary field study, grounded in a Naturalistic Decision Making philosophy, conducted by the first author. We then develop a theoretical framework describing the cognitive processes involved in solving mathematical problems. This framework will help frame the issues discussed in this paper and guide our work going forward. We discuss user interface issues that affect the ease of use of CASs. Finally we look forward to where we expect the research to lead and at some of the techniques we expect to use such as Cognitive Work Analysis.

COMPUTER ALGEBRA SYSTEMS IN THE CLASSROOM
A CAS is a piece of computer software that performs symbolic manipulations on algebraic structures. These symbolic manipulations are done using exact and symbolic calculations as opposed to software such as Matlab, which operates primarily using floating-point numbers. It should be noted that the line between symbolic systems such as Maple and Mathematica and numerical systems such as Matlab is blurred as symbolic systems are becoming quite efficient at handling floating-point calculations and numerical systems, such as Matlab, have symbolic toolkits available.

CASs, such as Maple and Mathematica, are being used in classroom settings as a complement to traditional teaching methods. There are several possible reasons that an instructor or institution might have in introducing such a system into a course curriculum. First, a CAS allows a student to focus on mathematics at a higher level and avoid detailed computations. Second, the instructor is able to assign students larger, real-world problems. Finally, students are introduced to software they may encounter in the future either in industry or academia. Such CASs are being used at many post-secondary institutions. There are a number of examples of the use of such systems in secondary education in France as noted in Lagrange (2005) and Artigue (2003). An example of the use of CASs in the United States can be found in Palmieri (1991). The University of Waterloo uses one such CAS, Maple, in a number of first and second year courses in mathematics and engineering.

In any classroom use, a goal in the use of such a CAS is going to be the improved comprehension of mathematical concepts by the students. In order for that goal to be reached, we must ensure that the user interface does not impose an undue cognitive load on the user. We will discuss why this is the case and situations where undue cognitive load may occur later in this paper when we outline a cognitive model for learning and problem solving. As a start to developing
a theoretical framework and examining possible improvements to user interfaces for CASs, a small field study was conducted, which we will now examine.

A PRELIMINARY FIELD STUDY

As a starting point for this research we conducted a small, informal field study involving students in a first year Calculus course at the University of Waterloo. A field study allowed us to observe how the students would approach problem solving using Maple in a natural environment. By using a field study we would not add artificial limits or requirements that may affect how the students solve problems. Hutchins (1995) and Woods (1993) have both noted that if we constrain the environment in which a person solves a problem, as we might be tempted to do in a laboratory setting in order to simplify a study, we may miss critical observations of events that do not occur due to the constraints but would occur in a more natural environment. Mumaw, Roth, Vicente, and Burns (2003) have noted that field studies also give us an opportunity to discover new phenomena and develop new ideas. Also mentioned in Hutchins (1995), Woods (1993), and Mumaw et al. (2003) is that field studies are the only way to observe all the complexity involved in operating in a natural environment. For these reasons, we chose a field study as an appropriate method for the exploratory stage of this research.

In this study, the students used Maple to complete a lab assignment as part of their course work. Four students were observed for a total observation time of six hours. During the study the four students who participated completed an initial background information survey, were observed completing some of their assignment, and completed a survey to record their impressions regarding the use of Maple to complete their assignments. During the observations notes were taken regarding the general approaches students used when completing the assignments and any problems they ran into that might have be caused by the user interface. There were a number of problems that we observed students dealing with that occurred as they tried to use Maple to complete their assignments. In particular students had difficulty dealing with some subtleties that exist in the Maple language syntax and semantics. The students also had difficulty remembering the commands they needed to use and the parameters for those commands. They also had difficulties remembering what variables they had defined and what packages they had loaded. Packages are sets of commands that do not form the core of the Maple language but can be loaded as needed. The students often had to refer to earlier parts of their worksheet to remind themselves of what work they had previously done.

One final, particularly interesting problem that was observed involved difficulties students had when trying to understand the mathematics behind a concept and the Maple syntax necessary to explore the concept in Maple at the same time. When it was suggested to the students that they try figuring out the mathematics of the problem first and then generate the appropriate Maple syntax they had little trouble solving the problem; however, when trying to understand both at the same time they had difficulty. Since we provided assistance to the students, this is not a purely observational study; however, by interacting with the students in this manner we gained a deeper insight into the source of this particular problem, which all the students encountered. We will return to this problem later when discussing extraneous cognitive load. In summary, the field study revealed difficulties with syntax and semantics as well as the remembering of commands. More interestingly, there was some indication that interface issues might, in some situations, interfere with the understanding of mathematical concepts. Further study into how students use CASs, how the CASs help students, and when the CASs might cause difficulty will require an understanding of the cognitive issues involved in the use of such systems. We will now examine a cognitive model that has been developed to help explain how people solve problems and how they improve their problem solving skills in a given domain. Such a model should be helpful when surmising and examining how student solve problems and learn mathematics using a CAS.

A COGNITIVE MODEL

The model proposed here is based on the concepts of Cognitive Load Theory (CLT) found, for example, in Sweller (1988) and Cognitive Work Analysis found, for example, in Vicente (1999). There are many similarities between these approaches. Sweller’s work focuses on the ideas of cognitive load and has particular application to situations of learning and problem solving. Sweller (1988) observed that novice and expert problem solvers, with respect to a domain of expertise, solve problems differently. Novices use a means-ends analysis to work backwards from a goal setting sub-goals to be solved. (Note this differs somewhat from the means-end structure we see in a work domain analysis, where means-ends relationships are defined in terms of function or “how-why” relationships. Means-end analysis in this case refers to starting from the goal, or problem solution, and back-tracking through problem steps to the start of the problem, suggesting that this analysis is more focused on problem-solving sequence.) As such a process often results in backtracking during the solving process, a high cognitive load is imposed on the problem solver. Experts, however, are able to quickly classify a problem according to solution type and move from problem to solution without backtracking.

Sweller’s work is similar to and compatible with the three level model of cognitive control found in Rasmussen (1986) known as the Skills-Rules-Knowledge (SRK) model. In this model, novices or experts facing unanticipated problems take a knowledge-based approach to solving the problem. However, with experience and recognition of the problem
type, operators may use rule-based knowledge, which is less robust but faster and more efficient. Rule-based behavior is not inconsistent with some of the ideas behind recognition-primed decision making (Rasmussen, 1986, Klein, 1998).

Sweller’s work compliments this by proposing mechanisms that explain how learners to transition between levels of cognitive control. Sweller argues that learners develop schema, or mental structures, that allow the solver to classify problem states and identify how to move from the given state towards the solution. The process of moving from a novice to an expert within a given domain is the process of developing schema and storing them in long-term memory such that they will be easily accessed when solving a given problem. Cognitive load is the amount of activity in working memory at a given time. Cognitive load increases with the number of elements in working memory and with the complexity of the interactions between those elements. This is the basic tenet of Sweller’s CLT, which has been used to explain such well known phenomena as Miller’s 1956 limit on working memory of seven, plus or minus two and chunking (Miller, 1956).

Sweller, van Merrienboer, and Pass (1998) defined three types of cognitive load imposed when trying to learn a given domain of knowledge: intrinsic, germane, and extraneous. Ayers (2006) and Jeroen, van Merriënboer, Kester, and Paas (2006) nicely summarized the three types of cognitive load, which we will review here. Intrinsic cognitive load is the cognitive load placed on the learner by the nature of the material to be learned. Intrinsic cognitive load is caused primarily by element interactivity, the degree to which the information must be understood as a whole for it to maintain its meaning. Ayers notes that, since element interactivity is the inherent interconnectedness of the task elements, we can not change the element interactivity of a task without changing the nature of the task. However, it may be possible to build up knowledge in such a way that the intrinsic cognitive load on the user at a given point in the learning process is reduced; later we will examine how this might be accomplished. Germame cognitive load is the cognitive load imposed by schema formation. Since schema, once formed and stored in long-term memory, help move the problem solver from the novice to expert stage, this type of cognitive load is beneficial. Extraneous cognitive load is the cognitive load placed on the learner by the particular instruction material or method through which the material is being taught. There are many possible ways to reduce extraneous cognitive load by redesigning the presentation of learning material; we will examine some of those approaches in a general way later in this paper.

From the above we can see that germane cognitive load is the type of cognitive load we would like to impose on a learner since this will increase schema formation, which will improve the learner’s problem solving skills in the given domain. Since working memory is highly limited, we must reduce intrinsic and extraneous cognitive load to allow working memory to focus on germane types of cognitive load. We will look at ways to accomplish the goal of decreasing intrinsic and extraneous cognitive load during the learning and problem solving processes. Before that we will examine the role CASs as an aid to human cognition when solving mathematical problems.

**COMPUTER ALGEBRA SYSTEMS AS A COGNITIVE TOOL**

Hutchins (1995) in “Cognition in the Wild” discussed the implementation of pilotage on large ships. He discussed many tools used on such ships that reduce the cognitive loads on various crewmembers as they map the location of the ship as it moves. These tools include, for example, maps and charts, which we might think of as being fairly simple tools but in fact significantly simplify computation. He described the amount of effort that has gone into these tools as a type of pre-computation that unloads cognitive work from a user of the tools. The cognitive work that has gone into the creation of such tools is greater than any one person could afford to complete in one lifetime. As such, Hutchins also described the cultural processes that create these tools as a type of distributed cognition spread across people and time. Thousands of years of navigational knowledge has been accumulated in the methods and tools of modern pilotage. He refers to this process of knowledge accumulation as cultural cognition.

Similarly, pre-computation and cultural cognition is common in mathematics. The tools that have been developed to reduce the cognitive load on problem solvers are diverse and include mathematical tables such as statistics or integration tables and algorithms such as differentiation rules. Historically many of these tools were collected in textbooks and referred to by students and mathematicians as needed to solve problems when the limits of cognition and memory would not allow them to completely solve a given problem in their head. Many other tools have been developed over the past centuries that reduce the cognitive load on problem solvers such as slide rules and calculators.

CASs are a next step in the evolution of mathematical tools. They incorporate the knowledge of many branches of mathematics into one system and they make the knowledge relatively easy to access by anyone who knows how to use the system. These systems also automate computations and in a sense take on some of the computation load from a problem solver. We will look at how CASs take on some cognitive load from the user, reducing intrinsic cognitive load during learning and problem solving. We will then look at how these systems can reduce extraneous cognitive load but also how, in some cases, they imposed unnecessary extraneous cognitive load. We will also outline ways to alleviate this problem.
MANAGING INTRINSIC COGNITIVE LOAD

As we have noted, intrinsic cognitive load is the load placed on the learner by the nature of the materials being learned. It is desirable to minimize the intrinsic cognitive load; however, eliminating it completely will be difficult since it depends largely on the complexity and element interactivity of the task at hand. Ayers (2006) notes that “as expertise develops in a domain, the intrinsic load caused by a specific task decreases as the interactions become learned and incorporated into schemas”. We can take advantage of this fact when involving CASs in learning mathematics.

With any mathematical problem there are sub-problems to be solved that can be solved easily by a CAS, alleviating some of the intrinsic cognitive load from the learner. For a simple example, in a problem where a student is attempting to find the maximum of a function in an interval, the student must find the derivative of a function. While the mechanics of finding the derivative are important, in this context such mechanics impose an additional cognitive load that interferes with learning and understanding how to maximize a function.

One thing we must be careful of, however, is that unless the design of the CAS is done carefully it could impose additional extraneous cognitive load while reducing intrinsic cognitive load. We will now examine extraneous cognitive load as it relates to CASs.

SOURCES OF EXTRANEOUS COGNITIVE LOAD

An important consideration in the design of user interfaces is the extraneous cognitive load the interface places on the user, particularly if the CAS is being used as a learning tool. Any extraneous cognitive load will interfere with germane cognitive load and therefore will interfere with schema formation. We will look at a number of possible sources of extraneous cognitive load.
Chandler (1992) and Sweller (1994) discussed a split-attention affect that occurs if a learner needs to integrate information from several sources while studying a particular document. For example, a textbook problem and solution example as shown in Figure 1 imposes a split-focus on the learner as he or she must integrate the diagram with the text below the diagram to understand the solution. This split-focus creates extraneous cognitive load. A better design is shown in Figure 2 where the text and diagram is integrated.

There are a number of possible sources of split focus in a CAS. CASs generally have formatted their input and output in a linear fashion in a sequence of input and output statement. It has traditionally been difficult to create information rich documents such as in Figure 2. There are more features being added to some systems to create such documents, but more work remains to be done.

Another possible area of extraneous cognitive load appears when the syntax of the CAS and the mathematical syntax that a user expects to use do not match. In such a case the user must constantly translate between the syntax of the CAS and normal math notation and vice-versa, imposing additional cognitive load. Some CASs now allow a more natural method of entering mathematics which has to some extent alleviated this problem; however, there is still room for improvement. The problem that students involved in the above mentioned field study encountered when trying to understand both the mathematics and the Maple syntax at the same time was partially related to this problem.

A final source of extraneous cognitive load also noticed in the field study was the need to recall what work had already been done in the worksheet. When variables are defined and packages loaded there is often no visible reminder of this. As a result, the user has to use working memory to remember what variable are available and what packages have been loaded. This places an extraneous cognitive load on the user. Providing visual reminders to the user of what variables are defined and what package are loaded should also provide cues to help users retrieve from long-term memory information about what work they have already done in the worksheet. How cueing can help retrieve information from long-term memory is discussed in Card, Moran, and Newell (1983).

These and other sources of extraneous cognitive load reduce the usefulness of a CAS as a teaching tool since the extraneous cognitive load will interfere with schema formation. It is therefore necessary to identify ways to reduce extraneous cognitive load as well as use CASs to reduce the intrinsic cognitive load involved in learning mathematics. We will now look at where we will go from here to reach these goals.

**FUTURE WORK**

In the long run the goal of this research is to improve user interfaces for CASs. The first step will be to continue to develop a theoretical framework to reason about how people learn mathematics, solve problems, and use CASs. We will also look to tie in the SRK framework better with CLT. In the SRK framework the decision maker or problem solver can transition from using knowledge to rules to skills as he or she becomes more familiar with the decision or problem in question (Rasmussen, 1986). Schema formation and automation, key concepts of Cognitive Load Theory, as discussed in Sweller et al. (1998), may help explain how and when this transition occurs. Moreover, if we can develop methods to encourage germane cognitive load, and reduce intrinsic and extraneous cognitive load, we may be able to encourage and speed up the transition from knowledge-based behaviour to rule- and skill-based behaviour, effectively speeding up the learning process.

With respect to CLT we will continue look at ways CASs can reduce intrinsic cognitive load. We will also continue to look for instances where extraneous cognitive load is imposed by the CAS interfaces. Once these areas are identified we can develop ways to reduce these types of cognitive load.

A third type of cognitive load, germane cognitive load, was mentioned but not discuss in detail in this paper. As was mentioned, this type of cognitive load is desirable as it is the cognitive load imposed by schema formations and the schemas formed will improve problem solving ability. In this paper we have considered the goal of reducing intrinsic and extraneous cognitive load to allow more germane cognitive load. Once we accomplish this goal, we can consider ways to increase schema formation.

We will also consider when different instructional methods will be beneficial for different type of tasks and skill levels of the learner. Van Merriënboer, J., Kester, L., & Paas, F. (2006) discussed various methods that are appropriate in different cases. Their research should prove a useful starting point for this line of inquiry.

After we have developed a more thorough theoretical framework, we can apply a number of methods, such as cognitive work analysis, to develop some practical solutions. We will then be in a position to return to field studies to determine if the solutions that we derive improve how CASs help with learning and problem solving. We will also investigate how our research applies to the use of CASs outside the classroom.

**CONCLUSION**

CASs have proved useful as teaching tools for teaching mathematics as well as problem solving tool for professionals. In this paper we have outlined the beginnings of a theoretical framework for how people learn mathematics and solve mathematical problems. We have looked in a general sense at some areas where cognitive loads can be reduced by the
use of a CAS and where extraneous cognitive load may be imposed by the design of the user interface for a CAS. Finally, we have looked forward at were the research can go from here. It is hoped that this research will allow CASs to continue to improve as teaching and problem solving tools.

ACKNOWLEDGEMENTS
We are thankful to Maplesoft and the Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this work. We would like to thank Jonathan Histon and Bev Marshman of the University of Waterloo for their advice and guidance during the field study discussed in this paper. We would also like to thank the students of MATH 137 in Fall 2008 who participated in the field study.

REFERENCES


