A Cognitive Model of Tax Problem Solving

Roland Hübscher, Bentley College, USA, [rhubscher@bentley.edu]
Brigitte Mühlmann, Bentley College, U.S.A, [bmuehlmann@bentley.edu]

Abstract: Tax professionals are reluctant to use the Internal Revenue Code (IRC) due to its tremendous complexity and size. With the goal of providing an intelligent and scaffolding interface to tax professionals, we have studied the problem-solving behavior. Our research has shown that even experienced tax professionals lack the necessary problem-solving skills to use the IRC properly. Based on this research, we propose a cognitive model for tax professionals solving tax problems. We have applied the model to the data we have collected in our original experiment and show that it indeed explains the behavior of novices and experts rather well.

Keywords: Internal Revenue Code, problem solving, cognitive model

Introduction

The goal of our long-term project is to provide tax professionals with an intelligent interface to the Internal Revenue Code (IRC). Tax professionals tend to rely on secondary literature and avoid using the IRC due to its enormous complexity and size, although conclusions reached in treatises, legal periodicals, or legal opinions rendered by tax professionals are not accepted as authoritative sources.

Since we want to provide intelligent support to the tax professionals, it is important that we understand why novice and expert problem solvers behave as they do. We studied a group of graduate students in taxation and a tax expert solving a relatively simple tax problem and found that the students had surprisingly significant problems (Hübscher, Muehlmann, and Ono 2005). Whereas the expert showed an efficient approach taking advantage of characteristics of the code, the students used an approach of matching the given data to all sections in the hope that there is a match. This approach was only possible because the students were given a six-page excerpt of the code, and this approach will not scale up with the size of the IRC.

We propose a cognitive model of tax problem solvers based on our earlier work with the goal to explain the problem-solving behavior of tax problem solvers of any expertise. We view the problem solver as a resource-limited agent where the resources include time, tax knowledge and access to data. Further limitations at the meta-level are knowledge about how resource-consuming certain subproblems are and how valid their results are.

Effective problem solvers in different domains use different approaches based on the concepts, facts, and procedures associated with expertise and various types of usually tacit knowledge that underlies an expert's ability to make use of them to solve problems and accomplish tasks which is also known as strategic knowledge (Collins, Brown, and Holum 1991). Effective problem-solving knowledge thus involves domain-specific problem-solving heuristics, strategies that control the problem-solving process and learning strategies that experts use to acquire new concepts, facts, and procedures. It is an individual's ability to carry out complex problem-solving tasks that qualifies the person as an expert.

Because tax professionals are knowledge workers, it is important to keep both the performance and the educational aspects in mind (Argyris 1991). Therefore, an expert system approach where the system would solve the problem for the user would be inadequate, because the learning impact would be minimal. To ensure that learning occurs during use, we will provide an interface that acts as an intelligent assistant by scaffolding the user (Puntambekar and Hübscher 2005). Scaffolding in the context of learning was originally defined as an "adult controlling those elements of the task that are essentially beyond the learner's capacity, thus permitting him to concentrate upon and complete only those elements that are within his range of competence" (Wood, Bruner, and Ross 1976). Thus, the interface should only provide enough support that the tax professional, not the system, can solve it. And the support should be such that the tax professional is still challenged to an appropriate degree.

Scaffolding has been linked to the work of soviet psychologist Lev Vygotsky, although he never used the term scaffolding. According to Vygotsky, a novice learns with an expert, and learning occurs within the novice's Zone of Proximal Development (ZPD). ZPD is defined as the "distance between the child's actual developmental level as determined by independent problem solving and the higher level of potential development as determined through problem solving under adult guidance and in collaboration with more capable peers" (Vygotsky 1978). Enabling the learner to bridge this gap between the actual and the potential depends on the resources or the kind of support that is provided. Instruction in the ZPD can therefore be viewed as taking the form of providing assistance or scaffolding, enabling a child or a novice to solve a problem, carry out a task or achieve a goal "which would be beyond his unassisted efforts" (Wood, Bruner, and Ross 1976). Scaffolding in this context then implies the following (Stone 1998; Palincsar 1998).

- The learner must be aware of and interested in the goal of the learning activity. We can assume this for the tax professional.
- Continuous assessment of the learner needs to be used to calibrate the support.
- Scaffolding fades away over time and the learner must take control of the task. The tax professional becomes an expert, at least in solving a certain class of tax problems.
- The learner needs to be actively involved in the learning process. The problem solver and decision maker is still the tax professional, not some tax expert system.

What is essential here is that that the user needs to be continuously assessed so that the support can be calibrated and fade away to keep the user in the ZPD. Thus, we need to have a clear understanding of the problem-solving process continuum induced by the performance of novices on the one hand, and experts on the other. We present a cognitive model as an explanation of the tax solvers' behavior that is not just descriptive but also explanatory. It will then be our next step to test it with additional empirical studies.

In the remainder of this paper, we will first summarize the main results from our earlier study focusing on the different behaviors of a tax expert and tax students. Then, we will propose a cognitive model that is based on the results of that study. Next, we will explain the behavior of the various problem solvers. In the conclusions we will also suggest how we plan to further test and improve our cognitive model.

Solving Tax Problems

General approaches to problem solving are called weak problem-solving methods because they don't do well when applied to a specific problem. Their advantage is that they can be applied to a large class of problems. Examples of such approaches are depth-first search, hill-climbing and means-ends search. Nevertheless, weak methods can be specialized, combined and fortified with domain specific knowledge resulting in strong methods. These strong methods are therefore domain and problem specific and more efficient than the general ones, often by several orders of magnitude. Human problem solvers have also weak methods that they can apply to unfamiliar problems. Over time, as they solve more problem of the same class, they develop more specific strategies, know when to apply which strategy and take advantage of domain knowledge that allows them to make assumptions that may not be valid in other domains (Fensel and Motta 2001).

Since we are interested in people solving tax problems, it will be interesting to see whether such strategies and domain knowledge assumptions can be used to explain the problem solvers' behavior. Furthermore, there is evidence that experts in different domains reason differently. For instance, solving problems in mechanics, experts often use spatial problem-solving methods (Hegarty 2001), whereas this is rather an uncommon approach in the tax domain. Amsel and colleagues found that psychologists are better causal reasoners than lawyers (Amsel, Langer, and Loutzenhiser 1991).

A Study

We reported earlier on a study with the goal to understand how tax experts solve tax problems and what difficulties they have using the IRC (Hübscher, Muehlmann, and Ono 2005). Here is a brief summary of that study. We asked three graduate tax law students and a tax expert to solve a rather simple tax problem. They had access to six pages of the IRC that were relevant to the problem to be solved. We carefully analyzed the transcription of the think-aloud protocols and found that there were big differences between the expert's and the students' approaches.

The tax expert used a strategy that was mainly based on the local structure of the code sections, whereas the students showed a more data-driven approach. We differentiate between a local structure and a global structure. The local structure refers to the various parts like "the general rule," "exceptions," "computations of actual amounts," etc. within a section; these sections are not always explicitly labeled as such, however, an experienced tax problem solver is familiar with them. By global structure we refer to the hierarchy in which the thousands of sections are organized.

The data-driven approach of the students was reminiscent of the well-known approach taken by fumbling students in math and physics classes: Find a formula to plug in the given numbers and hope that whatever is computed by that formula is going to be useful. Here, the rules in the code sections played the role of the formulas.

The expert questioned neither the decisions made during the problem-solving process nor the solutions, but made some comments reflecting on the progress, e.g., by explicitly marking the start and accomplishment of a subgoal. Two students made several comments implying that they did not really trust their own approach or the result, while the third did not make any meta-comments at all and seemed to just follow a simple procedure. In general, the three less experienced subjects double-checked their result by skimming through most of the tax code to see whether there might be a paragraph that applies to the problem, which they had not used. Of course, this would have been impossible if they had access to the whole 5,000-page IRC and furthermore, it does not really address validating how they have arrived at the result.

The three main differences between the expert and the less experienced users (graduate students of taxation, some with extensive tax experience) were:

- The expert used a code structure-driven approach, the students a data-driven approach
- The expert used an efficient expectation-driven approach based on a deep understanding of the code and problem domain. The students used highly inefficient "strategies" (random, linear and memorized).
- The expert was actively reflecting on the problem-solving progress, whereas the other subjects either did not reflect at all or were full of self-doubt.

Explanation vs. Description

Although this study provided good insights into the problems novices have with the IRC, it was somewhat unsatisfactory, because it did not explain why exactly the behaviors were so different. The expert in the study did not express any heuristics or control strategies when reflecting on the problem-solving progress. If such strategies were applied, they remained in the domain of the expert's tacit knowledge. Therefore, the goal is to go beyond a description of the behavior by developing a cognitive model of solving tax problems.

Scaffolding requires continuous assessing of the users to support the users appropriate to their knowledge and skills. For now, we do not make any assumptions about how we are going to implement scaffolding. First, we need to understand how we can explain the problem-solving behavior of tax professionals with different expertise and how proper feedback and guidance will result in a correct solution—performance still matters—and improved problem-solving skills.

Cognitive Model

We propose a cognitive model that will help explain the problem solvers' behavior. In its current form, its purpose is to explain the behavior we observe and to guide the design of further experiments. Once we have validated the model, we will use it to guide the design of the intelligent interface and especially, adaptive scaffolding. The problem-solving approach consists of two main activities: Searching for relevant code sections and applying the code to the problem. Search will be heavily supported by the user interface and is not part by the cognitive model. Thus, we focus on the problem of applying the code.

For those unfamiliar with the IRC, Figure 1 shows an example of the beginning of a section, specifically of section 117 located in the IRC hierarchy in TITLE 26, Subtitle A, CHAPTER 1, Subchapter B, PART III. We will regularly refer to this section as we describe the model and how the problem solver uses the code. The fragment shown in Figure 1 is less that 0.01% of the IRC.

Figure 1: A fragment of IRC section 117.

Sec. 117. - Qualified scholarships

(a) General rule

Gross income does not include any amount received as a qualified scholarship by an individual who is a candidate for a degree at an educational organization described in section 170(b)(1)(A)(ii).

(b) Qualified scholarship

For purposes of this section -

(1) In general

The term "qualified scholarship" means any amount received by an individual as a scholarship or fellowship grant to the extent the individual establishes that, in accordance with the conditions of the grant, such amount was used for qualified tuition and related expenses.

(2) Qualified tuition and related expenses

For purposes of paragraph (1), the term "qualified tuition and related expenses" means -

(A)

tuition and fees required for the enrollment or attendance of a student at an educational organization described in section 170(b)(1)(A)(ii), and

(B)

fees, books, supplies, and equipment required for courses of instruction at such an educational organization.

(c) Limitation

(1) In general

Except as provided in paragraph (2), subsections (a) and (d) shall not apply to that portion of any amount received which represents payment for teaching, research, or other services by the student required as a condition for receiving the qualified scholarship or qualified tuition reduction.

(2) Exceptions

Paragraph (1) shall not apply to any amount received by an individual under -

(A)

the National Health Service Corps Scholarship Program under section 338A(g)(1)(A) of the Public Health Service Act, or

(B)

the Armed Forces Health Professions Scholarship and Financial Assistance program under subchapter I of chapter 105 of title 10, United States Code.

(d) Qualified tuition reduction

(1) In general

…

AND/OR Trees

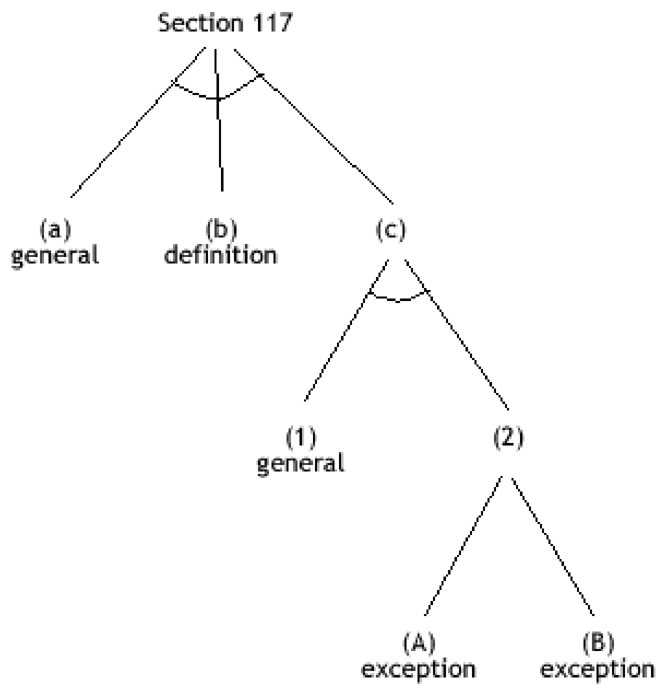
As mentioned earlier, code sections have a more or less consistent structure. For instance, often there is a general rule and some additional clarification (see Figure 1). Normally, sections consist of conditions to test the section applicability, definitions also to test whether the section applies to the entities mentioned in the problem, and computations. Of course, many of these parts refer to other sections and the parts themselves are organized, more or less, logically as a conjunction, disjunction or some combination of those. Applying a code section to the current problem can then be viewed as generating a, potentially large, AND/OR tree (Russel and Norvig 2002) and processing it. However, we don't make any claims that the problem solvers actually think about the problem space in terms of AND/OR trees, nor are we planning to formalize the whole IRC.

We can define AND/OR trees recursively as follows. An AND/OR tree is either a leaf, an AND node or an OR node. An AND node has subtrees that all need to be successfully processed, and an OR node has subtrees of which at least one needs to be successfully processed. Subtrees are of course also AND/OR trees. Each code section generates a tree whose structure reflects the structure of the section. When a section is deemed relevant by the problem solver, the tree that corresponds to the section is added to the overall AND/OR tree. For instance, in section 117 shown in Figure 1, in "(a) General rule" there is a reference to "section 170(b)(1)(A)(ii)" the corresponding tree would have to be added to the current tree.

A common local structure consists of a general rule G and a list of possible exceptions $E_1, E_2, \ldots, E_n$. More formally, this amounts to $G \wedge \neg(E_1 \vee E_2 \vee \ldots \vee E_n)$ or, more useful to generate the AND/OR tree, it can be rewritten as $G \wedge \neg E_1 \wedge \neg E_2 \wedge \ldots \wedge \neg E_n$. Thus, the general-case-plus-exceptions structure generates a simple AND node. Of course, the complete section 117 would generate a combination of AND and OR nodes.

The fragment of section 117 shown in Figure 1 can be represented as the AND/OR tree in Figure 2. An AND node is represented with an arc going through the link to its subtrees and an OR node has no arc. Thus, to process section 117, all of (a), (b) and (c) need to be processed. For (c) to be successfully processed, (1) and (2) have to be processed as well. However, for (2) to be processed, only one of (A) and (B) needs to be checked. For instance, if the definition in (b) does not apply to the current problem (then it is not successfully processed), then looking at (c) is a waste of time because section 117 is an AND node and all its subtrees (a), (b) and (c) must apply but since (b) doesn't, section 117 does not apply. Or, if exception (A) applies then (B) does not have to be checked. Note that although we call it an AND/OR tree since it does have this structure, this does not imply that we somehow compute a Boolean function based on the tree's structure. The tree structure simply allows us to clearly define what must be processed and where the problem solver will have a choice.

Figure 2: The AND/OR tree of the shown fragment of section 117.



A leaf is a node that is not further expanded. It is the problem solver's decision to expand a node or not. For instance, in section 117(a)(2) with the heading "Qualified tuition and related expenses" there are two conditions (A) and (B). The first makes a reference to section 170(b)(1)(A)(ii) but the tax solver might decide that, based on his or her previous experience, it is clear that this condition does not apply, so 117(a)(A) is a leaf. Another problem solver might expand it. However, most solvers would probably not expand condition (B) any further except if they are so inexperienced that they need to check for the definitions for some of the terms, that is, what the IRC really means when refereeing to fees, books, supplies, etc. Also note that the (grammatical) conjunction connecting conditions (A) and (B) is "and", however, the combination of (A) and (B) is a disjunction.

A tree is processed by either evaluating conditions, computing values (dollar amounts, time periods, etc.), assessing the validity of the result, aggregating the values from the subtrees, or expanding nodes, thus growing the tree. Since every problem solver, no matter how experienced, will implicitly or explicitly generate an AND/OR tree based on the code sections used, we use these AND/OR trees to define the problem space. We are interested what kinds of trees the various solvers create, and in what order they grow these trees. Therefore, we need to specify which nodes are processed, and when a node can be considered a leaf, that is, is not be further expanded.

Creating an Argument

The answer to a tax problem is not just the result, but also the argument justifying the result. This is different from some other, possibly very difficult problems where the correctness of the result can easily be tested. This is not so with tax problems. Convincing somebody of its correctness can only be done by providing an argument that uses proper reasoning, reference to the IRC and possibly some other sources like the code's intent, Internal Revenue Service interpretations, and some case law, and the data specific to the problem at hand.

An informal definition of an argument will do for the current discussion. Based on Toulmin's ideas (Toulmin 1969), there has been quite some work on which a formalization could be based, which we will take advantage of if required in the future (Prakken 2005). An argument for a claim requires the data and a warrant for each datum. This warrant explains how the datum supports the claim. As a result, the claim is supported to a certain degree, that is, its truth is not always known with absolute certainty. Furthermore, since the data may be known only to a certain degree, this may reduce the belief in the claim as well. And even a warrant, which is often based on the tax code, may add to the uncertainty to the extent it requires an interpretation by the problem solver.

The argument that backs the result needs to have a certain minimal quality, that is, the uncertainty needs to have at least some minimal value. The argument of course consists of many subarguments that form a large tree and is generated in parallel with processing the AND/OR tree. Let us assume that the problem solver has arrived at a node and processed that node arriving at a result with a large enough certainty. For instance in section 117(b)(2)(A) it is clear enough to the solver what an "educational organization" is in this context. Thus, the argument is strong enough and this node does not have to be further expanded. It is successfully processed. However, if the solver is not familiar enough with the definition of an educational organization, the node needs to be expanded and an argument is generated supporting the current problem's institution being or not an educational institution. A different kind of expansion happens when the definition is clear, but the data is not. Assume the institution in question is not widely recognized as an educational institution, say, the Alice Lloyd College. Then, further research into the exact status of Alice Lloyd College and possibly the IRC section 170 is required. (Alice Lloyd College is indeed a private college located in Pippa Passes, Kentucky.)

In other words, it is the quality of the argument that guides whether a node needs to be further expanded or whether it is going to be a leaf. But what is the proper quality of the argument? Interestingly, there are clear regulations for this in Section 6662 of the Internal Revenue Code. In general, a taxpayer must demonstrate that there is or was substantial authority for the treatment of any item, or, alternatively, that there is a reasonable basis for the tax treatment of an item, a lower level of confidence, if the taxpayer discloses the relevant facts affecting the item's tax treatment in the tax return or in a statement attached to the return. In the case of any item that is attributable to a tax shelter—a partnership, entity, plan, or arrangement with a significant purpose of tax avoidance or evasion—the taxpayer must reasonably believe that the tax treatment was more likely than not the proper treatment, which requires the highest level of confidence among the three standards.

Minimizing Costs

So far, we have described how an AND/OR is generated and whether a node is expanded or not. The final question is to specify in which order the nodes should be processed. There are always quite a few candidate nodes that are not processed yet and can be considered to be potential loose ends. Not all have to be processed, so the order does matter.

We postulate two simple guiding principles for this process. First, the tax professional must try to minimize the amount of taxes to be paid to the government, within the law and its intention, of course (American Institute of Certified Public Accountants (AICPA) 2000). The second principle states that the cost of solving the tax problem must be minimized. The cost is based on the amount of time required to solve the problem and to gather the data. Cost is also accrued by bothering the customer with frequent and possibly intrusive requests for additional data. What clients want is great service and value for money (Neely 2006). Thus, the goal of the problem solver is to create the cheapest AND/OR tree that results in a good enough argument to pay minimal taxes.

How to minimize the cost needs to be discussed for each of the three types of nodes. The cost of a leaf tends to be low as only a condition needs to be tested or a computation executed. Sometimes a tax expert also makes an assumption based on previous experiences. This assumption is then associated with an uncertainty and if this uncertainty is too high, the node may have to be expanded anyway. Thus, less experienced tax professionals may have to expand nodes more often. Experts also tend to validate the result, that is, explicitly check the result to determine whether it falls in the expected range. This will help to adjust their confidence into the result and allow them to either leave a leaf a leaf or expand it.

An AND node requires all its subtrees to be successfully processed. Thus, if one of the subtrees cannot be processed because it does not apply to the current problem situation, then the other subtrees need not be processed either. Thus, the goal is to find the cheapest refutation first. This will allow a cutoff in search-tree speak and allow avoiding unnecessary computations and potential data gathering. This also shows that it is just as important to understand when the law does not apply as it is to understand when it does apply. Experience from related problems allows experts to stop working on certain parts since they know that the tree will never be successfully processed because some refutation must be found (but they don't have to actually find it, they just "know" of its existence). This look-ahead can improve the problem-solving efficiency tremendously.

An OR node requires that at least one of the subtrees is processed. In this case, the problem solver should start with the cheapest node that has a good probability of being successfully processed.

The problem solver being familiar with the proper method of optimizing the processing of the tree is still faced with the problem of having to judge the certainty of results and the expected costs of processing trees. An inexperienced tax solver will do much worse and we expect that differences in expertise together with the optimization assumption will enable us to explain the different problem solvers' behavior.

To summarize, the main elements and assumptions of the proposed cognitive model of tax problem solving are the following.

- An AND/OR tree is generated based on the local structure of code sections.
- Parallel to the tree generation, an argument is created.
- A node is expanded if there is not enough confidence in the result and its argument.
- The cost of solving the problem and the amount of taxes paid are both minimized.
- The difference in problem-solving behavior between novices and experts can be explained in terms of confidence in results and arguments, and estimation of costs.

Note that the order in which the AND and OR nodes are processed is not an assumption but follows from the structure of the AND/OR tree and the assumption that cost is minimized. The exact order still depends on the cost and confidence estimates by the problem solver and often, there will be several nodes that are "best." However, a depth-first strategy seems to be reasonable since this strategy focuses on the current context whereas a breadth-first approach would result in jumping from one part of the problem to the next and this is generally a bad and confusing strategy for a human problem solver.

Explaining the Data

Does the cognitive model as described indeed explain the expert's and the students' behavior? It is rather easy to see that the model explains the expert's behavior well. The expert used a code-driven approach using the local structure to decide on which node to process next. The expectation-driven reasoning of the code suggests that the expert tried to estimate the cost and expected confidence of various nodes. This allowed the expert to cut off parts of the tree that could be ignored resulting in a smaller tree and lower cost. In addition, one of the nodes was not expanded because the expert explicitly evaluated the validity of the result and decided that the confidence in it was high enough based on experience.

Next, we will consider the observed behavior of the less experienced subjects and show how well our model explains it. Note that the goal is to explain the deviation from the optimal behavior by attributing certain resource limitations to the problem solvers like missing knowledge, incorrect estimates for confidence and cost, time pressure, etc.

Observation 1: The subjects used a data-driven approach. Based on the given data, they looked for a paragraph that might apply similarly to matching the input and output parameters of a function. They looked for those paragraphs randomly, linearly, and by direct lookup (student knew the six pages of code by heart).

Explanation 1: The students had no good concept of "relevant section" which refers to the search problem we do not deal with in this paper. A further reason for the inefficient and "unreasonable" search strategies is that the subjects did not seem to have a concept of expected cost and thus, each code subsection was as useful as any other and so selecting any unprocessed node was possible within our model.

Observation 2: Once the subjects had found a final solution, they double-checked it by skimming the whole code excerpt (six pages) to make sure they did not omit a relevant code subsection.

Explanation 2: The subjects did not know whether there were any loose ends, i.e., whether there are any leaves that needed further processing either by expanding them or by being aware that the result was known with high enough confidence.

Observation 3: There was no reflection supporting the problem-solving process. Two subjects did not reflect at all, and one subject was full of doubts during the whole problem-solving process.

Explanation 3: The non-reflecting subjects followed a procedure without being aware why they chose to work on a certain part of the problem rather than another (see Explanation 1). The doubting subject on the other hand seemed to be hampered by the insight that some of the leaves had too weak an argument and thus, the final result could not really be justified.

It is not surprising that the proposed model explains the expert well. After all, the model is based mainly on what we consider to be the "best" behavior and assume that, by considering the novices' resource limitations, their behavior can be explained. Thus, the cognitive model can be viewed as a combination of normative model plus resource limitations of the cognitive agents, the problem solvers. Since the cognitive model is not strongly formalized at this time, we are aware that it leaves quite a bit of wiggle room to explain the various observations. Nevertheless, we believe that even for the non-experts, the explanation of the problem-solving behavior is rather good.

Conclusions

We have proposed a cognitive model to explain the behavior of tax problem solvers of various expertise levels. The model is relatively simple, makes few assumptions and takes advantage of the relation between structures of the tax code, the problem space and the argument. The model explains our observations rather well.

In our next experiment, we need to address the issues that we discussed when explaining the tax law students' behavior. When thinking aloud, we will ask them to mention how they decided to work on a certain subproblem and why they did not expand certain non-trivial nodes. Of course, it will be important to modify the think-aloud protocol so that it does not disrupt the problem-solving performance too much. Furthermore, it has become clear that the subjects must have access to a non-trivial part of the IRC, because otherwise, they can employ primitive strategies that do not scale up with the size of the code. With these modifications and a more varied source of the subjects as originally suggested (Hübscher, Muehlmann, and Ono 2005), we hope to validate the cognitive model further or make the appropriate changes to it, if necessary.

Although this research is done with an intelligent, scaffolding interface as the main goal, our results so far suggest that we still have an incomplete understanding of what kind of support the tax professionals really need. Therefore, we refrain from making any suggestions as to which scaffolding mechanism we are going to implement. Nevertheless, it has become already quite clear that supporting the problem-solving process will be one of the most important elements to focus on.

Bibliography

American Institute of Certified Public Accountants (AICPA). 2000. Statement on Standards for Tax Services No. 1, Tax Return Positions.

Amsel, Eric, Rosanna Langer, and Lynn Loutzenhiser. 1991. Do Lawyers Reason Diffeently From Psychologists? A Comparative Design for Studying Expertise. In *Complex Problem Solving: Pinciples and Mechanisms*, edited by R. J. Sternberg and P. A. Frensch. Hillsdale: Lawrence Erlbaum Associates.

Argyris, Chris. 1991. Teaching Smart People How to Learn. *Harvard Business Review* (May-June):99-109.

Collins, A., John Seely Brown, and A. Holum. 1991. Cognitive Apprenticeship: Making Things Visible. *American Educator* 15 (3):6-11, 38-46.

Fensel, Dieter, and Enrico Motta. 2001. Structured Development of Problem Solving Methods. *IEEE Transactions on Knowledge and Data Engineering* 13 (6):913-932.

Hegarty, Mary. 2001. Knowledge and Processes in Mechanical Problem Solving. In *Complex Problem Solving: Pinciples and Mechanisms*, edited by R. J. Sternberg and P. A. Frensch. Hillsdale: Lawrence Erlbaum Associates.

Hübscher, Roland, Brigitte W. Muehlmann, and Mamiko Ono. 2005. Scaffolding Tax Professionals Using the Original Tax Code. In *The Fourteenth Annual Research Workshop on Artificial Intelligence and Emerging Technologies in Accounting, Auditing and Tax*. San Francisco.

Neely, A. 2006. The search for meaningful measures of performance. *Financial Times, Mastering Financial Management*, June 2, 10-11.

Palincsar, A. S. 1998. Keeping the Metaphor of Scaffolding Fresh--A Response to C. Addison Stone's "The Metaphor of Scaffolding: Its Utility for the Field of Learning Disabilities". *Journal of Learning Disabilities* 31 (4):370-73.

Prakken, Henry. 2005. AI & Law, logic and argument schemes. *Argumentation* 19:303-320.

Puntambekar, Sadhana, and Roland Hübscher. 2005. Tools fo Scaffolding Students in a Complex Learning Environment:

What Have We Gained and What Have We Missed? *Educational Psychologist* 40 (1):1-12.

Russel, Stuart J., and Peter Norvig. 2002. *Artificial Intelligence: A Modern Approach*. 2nd ed: Prentice Hall.

Stone, C. A. 1998. Should We Salvage the Scaffolding Metaphor? *Journal of Learning Disabilities* 31 (4):409-413.

Toulmin, Stephen. 1969. *The Uses of Argument*. Cambridge, England: Cambridge University Press.

Vygotsky, L. S. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press.

Wood, D., J. S. Bruner, and G. Ross. 1976. The Role of Tutoring in Problem Solving. *Journal of Child Psychology & Psychiatry & Allied Disciplines* 17 (2):89-100.