Games for Computational Thinking

[Extended Abstract]

Panagiotis Apostolellis
Virginia Tech
2202 Kraft Dr
Blacksburg, VA, 24073
panaga@vt.edu

Chris Frisina
Virginia Tech
2202 Kraft Dr
Blacksburg, VA, 24073
special@vt.edu Michael
Stewart
Virginia Tech
2202 Kraft Dr
Blacksburg, VA, 24073
tgm@vt.edu

ABSTRACT

Computational Thinking (CT) is present in every domain just as the other core literacy skills of reading, writing, and arithmetic. Unfortunately, current curriculum both introduces and teaches CT concepts predominantly in programming courses. We designed two physical CT games and had the opportunity to informally test and observe one of them in action. Our vision is to have K-12 students engage in different games that introduce and broaden their understanding of CT throughout their education. The development cost for each age-appropriate game is minimal, increasing feasibility of implementation and target audience participation. Individually, the games challenge students to grow their understanding of CT in an acute, focused activity, while collectively they maintain CT as a core literacy skill throughout studentsâĂŹ education.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

Since WingâĂŹs influential work preceding and during her service to the National Science Foundation, the research around Computational Thinking (CT) has exploded into myriad lines of divergent pursuits. Following prior work such as the Great Principles of Computing (Denning) on the topic of defining either prerequisites for computer science or its transferable skills, and coinciding with other work attempting to broaden participation in computer science (CS Principles), Wing et al. have demonstrated the immensity of the research space around CT. Much of this work has offered working definitions of CT , but none has become the standard.

Despite the current diversity of views, various institutions have committed to agendas supportive of âĂIJcomputational thinking.âĂİ In their recently adopted Long Range Plan, Virginia Tech (VT) identifies CT as a fundamental competency of a broad general education. In support of this mission, Dr. Dennis Kafura developed a graduate course to âĂIJdevelop a better understanding of what CT means at the university level of teaching and learning, and how CT could be made accessible to students in all disciplines at VT.âĂİ

In pursuit of this understanding, our group developed activities to be performed with students at a younger age, prior to their entrance in college, to help them develop CT.

2. COMPUTATIONAL THINKING

The research around CT is divergent in its definition. This is in part due to the variety of research agendas that are served by work on CT. While certainly different research agendas might simultaneously be served by a single pursuit, this at least introduces tensions.

Our working definition is as follows:

Computational Thinking (CT) involves using cognitive skills to represent abstracted modeling, construct and evaluate processes, and apply appropriate strategies to enable us to interact in structured and creative problem-solving techniques to address problems and assumptions. Additionally, a computational thinker reflects on how we are affected by the computational artifacts with which we (must) interact.

2.1 Recruitment of Computer Scientists

Each year, for several years now, there have been insufficient computer scientists for market demand. The result is increased reliance on imported laborers, exported labor, and decreased productivity, innovation, and security. To this end, some believe that supporting an agenda of teaching CT to many if not all students could increase the number of computer scientists.

2.1.1 Broadening Participation

If indeed, âĂIJComputer science is no more about computers than astronomy is about telescopes,âĂİ why do so many âĂIJCS0âĂİ courses begin with computers, and usually with programming? Especially as they were historically taught (with some notable exceptions such as Papert et al.), programming and its allure through precise control over the machine does not appeal to all.

In order to broaden the participation in computer science to more diverse demographics of students, it may be necessary to change the entry routes and introductions to be similarly diverse. In this vein, projects like ALICE, Media Computation, and CS Unplugged are useful exemplars. Many of the definitions developed for CT have specifically excluded programming; however many of these same works then attempted to teach CT using some form of programming.

Activities that differ from programming may appeal to different demographics. Additionally, some demographics that may lack resources (human, electronic, and monetary) sufficient for learning programming may be able to utilize CS Unplugged or CT curricula.

2.1.2 Increasing Participation

Aside from efforts attempting to broaden participation, there are efforts that merely seek to increase participation. How should someone know they might be interested in computer science? From the earliest education, we expose students to reading, writing, and arithmetic. When are they exposed to computer science? Unfortunately, only after a student [or their parent(s)] elects to take a programming course do they encounter computer science.

CT, rather than programming, might be taught explicitly, as its own subject, or implicitly in the context of other subjects, at ages much younger than a high school programming course. Then, more students will have an opportunity to recognize an affinity for skills that would support a pursuit of education in computer science, or at minimum, understand the benefits of applying CT in fields of their own interest.

2.2 Preparing the 21st Century Citizen

While improving recruitment of computer scientists is a major focus of many CT research agendas, an emerging concern is the general publicâĂŹs preparedness for the increasingly technological modern age. Many researchersâĂŹ definitions of CT attempt to specify important skills related to technology and computation that the researchers themselves predict to be necessary as our society continues to adopt technological solutions in virtually every domain. Some call this computational or technological literacy rather than CT, but as Professor Dane Webster mentioned in a presentation to our class, âĂIJDoesnâĂŹt teaching math students to use a pencil help them to think mathematically?âĂİ

2.2.1 Caveats

While our group (like many of the researchers before us) consider abstraction to be a critical part of CT, we believe that it is necessary to simultaneously teach the dangers of abstraction. In line with Blackwell, et al., we agree that abstraction might be âĂIJantagonistic to reasonable human concerns.âĂİ Along the same lines, Turkle and Papert have argued long ago that âĂIJconcrete and personal approachesâĂİ to computing should not be downplayed in favor of formal and âĂIJabstract approachesâĂİ, nor should they be dismissed as more inferior. Their thesis, based on ethnographic and psychology investigations, aligns well with the theory of situated learning which postulates that people learn contextually, by being actively engaged in authentic (learning) activities, and not in the decontextualized and abstract manner of formal education.

2.3 Educational and Computational Thinking Games

The tradition of using games for education is a long one. Children grow up and get to know the world around them through play. Most of their interactions in an early age are through touch, and other senses as well, and this is why most of the games of early childhood are tangible. This realization was what led the German educator Friedrich Froebel to build a series of wooden blocks known as Froebelgrabe (i.e., Froebel gifts) in an attempt to foster childrenâĂŹs self-directed activity in exploring the world and using the environment as a learning aid. Based on this notion of learning by doing, researchers at MIT Media Laboratory have coined the term objects-to-think-with as a means to leverage the power of computation in the exploration process. LEGO/Logo is maybe the most famous offspring of this approach where students are learning to program by controlling enhanced LEGO structures (equipped with motors and sensors). Resnick et al. has proposed âĂIJdigital manipulativesâĂİ as a continuation of this work where students are called to control through Logo programming other tangible artifacts like blocks, balls, beads, and badges.

Other researchers have also argued about the educational benefit of using computationally-enhanced construction kits where kids build 3D models and âĂIJlearn through tactile experiences.âĂİ. Such smart artifacts are equipped with technology that allows them to communicate with each other, with computers, and eventually provide feedback about their state to their users. Various systems have been proposed since then, employing a variety of advanced technologies like programming robotic motion of motorized pieces through kinetic memory or creating structures with blocks embedding sensors, logic circuits, and actuators for science education. Although these are exceptional examples of high-tech application in the service of informal education, they are mainly research projects that in most cases do not become commercialized. On the contrary, we were interested to investigate how low-cost games with easy to manufacture components can be leveraged to provide CT exposure to very young children.

For such an audience, motivation is believed to play a sig-

nificant part in the learning process; this is even more the case if learning objectives are intrinsic to the goal of the game. To motivate them, learners are immersed in a fantasy world and are asked to do challenging tasks by instigating their curiosity. These practices have been used in many educational games, including ones that teach about computer science and programming. Examples include either standalone games that leverage elements of fun to teach simple programming concepts like pattern recognition and recursion (e.g., Light-Bot), but also tools that leverage the students innate creativity to learn programming through game building (e.g., Scratch, Snap, and AgentSheets).

It is natural to want to extend the enthusiasm and promise of educational games to every domain, CT should be no exception. To offer an environment in which play and hopefully fun also result in learning CT would be ideal. However unlike many other domains, the definition and scope of CT is still in flux, making it challenging to argue (or at least to have agreement) that a given activity would support CT, without being so attached to computers and CS concepts. For example, Berland and Lee explored the CT that might be taught in an existing, commercialized board game: Pandemic. As Pandemic is essentially a graph-theoretical representation of several major cities in the world, and the game is cooperative, with the result that the game mechanic is the opponent rather than the other human players, there is some promise to this idea. The aspects of CT they chose to focus on were conditional logic, distributed processing, debugging, simulation, and algorithm building.

During our research for other popular games that can potentially be used for teaching CT indirectly although they have not been designed for this purpose, we reviewed many current, commercialized games, but also came up with some hybrid ideas (see Appendix: Computational Thinking in Existing and Imagined Games). And, to our delight, exactly contemporary with this stage of our work, CT was getting attention from a much broader audience through a Kickstarter project, teaching programming to little kids using a board game. Having completed this extensive analysis, and been unsatisfied, and also questioning whether it would be possible to teach CT in tangible games, we decided to construct our own low-cost games.

2.3.1 RabBit EscApe

Besides the identified need to involve children with CT from an early age, few endeavors focus on primary school children and younger. RabBit EscApe is a board game with tangible wooden pieces intended for ages 8 and up. As shown from previous work, tangible wooden blocks like the Froebel blocks20 are very engaging and educational for this age group, contributing to childrenâĂŹs emotional, social, and physical development through play. The game was inspired by the combination of the magnetic tile game Picasso-Tiles and the work from Weller et al. (2008) on objects-to-think-computationally-with using the Posey construction kit and a pacman-style game to teach state machine representations to children. In our approach we decided to avoid using any electronic medium like a computer in an attempt to reach younger ages which are accustomed to fiddling with tangible objects but are not yet ready to make connections between physical objects and virtual representations. The outcome was a combination of a board game with wooden pieces with encased magnets, described in the following section.

2.3.2 Design

Our idea started by using manipulatives that have been a favorite kidsâĂŹ educational toy for a long time, but also wanted to provide some structure to the game to better support CT, hence we decided to construct a board, as well. We opted for wooden pieces both for their elegance and warm feeling, but also for health reasons preferred over plastic or other chemically manufactured materials.

The game is comprised of fourteen different shapes of wooden pieces half an inch thick which we call bits; each bit is equipped with small magnets (1/16âĂŹâĂŹ thick and a diameter of 1/4âĂŹâĂŹ) which are encased in different sides of the bit and can attract or repel each other depending on polarity. By putting two bits together, you can make a block which is usually a square, rectangle, or hexagon. Other weird shapes can be constructed as well for more advanced levels. Magnets are placed either at the middle of a bitâĂŹs side or near one of the sideâĂŹs corners to allow a variety of path formations. From the fourteen different shapes, there are twenty-nine unique bits due to varying magnet placement and polarity combinations. We also provide a selection of boards with a path comprised of all, or a subset, of these 29 bits. Boards are of different levels of difficulty depending on the combination of pieces and the ambiguity in block formations.

The goal of the game is to put the bits on the predefined path and help the rabbit (another piece) escape from the fierce apes; hence the name of the game RabBit EscApe! The apes are square pieces which are placed in different positions on the board and the player(s) need to make sure that the adjacent bit repels an ape piece (i.e., same polarity magnets), instead of attracting it which is the objective when constructing the path. Figure 1a below depicts the path of LevelA as it was designed on the computer, and Figure 1b is a photograph of the actual board with most pieces placed in the right order (two are left intentionally off the path for illustrative purposes).

What makes the game especially challenging is the correct identification of the bitsâĂŹ properties and subsequent utilization of these properties to construct the blocks and then the whole path. These properties are: the size of each bit the position of the magnet on the bit the polarity of each magnet (engraved on the bitâĂŹs long side) the orientation of the bit (or whole block)

Players must combine all these attributes in order to put the bits next to each other in such a manner that they stick together at the right place (center or top of the bitâĂŹs side), while also repelling the enemies positioned around the board. The one level we designed for this course utilized all 29 pieces and all 4 enemies; we have made two variations of it, one with the block divisions drawn on the map and a more difficult one without them. Both of them were test during an informal evaluation we ran in a school near the Virginia Tech, described in the section: Evaluation and Observations.

2.3.3 Play Strategies

Training - Scaffolding In order to help players understand the game mechanics and the importance of the different piece properties and how they affect completing the board effectively, we suggest a training session preceding the actual gameplay. The sessionâĂŹs purpose is to scaffold the playersâĂŹ understanding of how putting the pieces together allows them to construct different shapes and satisfy the activity objective. This need derived from our informal evaluation in a primary school and is described in the following section.

Scaffolding includes levels of increasing difficulty, starting from making a single block (comprised of two bits) and moving on to more complicated shapes. At some points they will have the option to construct the different shapes using more than one combination of bits, and will have to take into account other piece properties and examine the implications of their choices. This mental modeling of the different affordances of each piece and the possible blocks it can construct will enable them to be more efficient in the actual activities, without demanding extra cognitive load to process them during gameplay.

Another idea to aid the bit selection and usage process is to have a separate âĂIJcheat sheetâĂİ with all pieces and their amount, next to the actual game boards. Players can cross out any pieces that have been used with a marker in order to keep track of what has been used and what is still available. This cheat sheet can be used during the initial sessions where players are learning the available bits and practicing with the strategies of putting them together. Then, these mental aids can be progressively removed, letting the users depend on their ability to recognize the physical pieces. This is a common process of scaffolding in instructional design, also called fading, because the additional information (scaffold) is removed (faded) over time.

The possible suggested activities for teaching CT with the game are described in more detail in the section that follows, although we understand that there are many variations that can derive from the suggested ones.

2.3.4 Individual or Collaborative

Players are given a predefined number of bits and a board and need to complete the board with all the pieces (i.e. build the path printed on the board), in a predefined amount of time (optional). They are playing individually or as a team and can either place a block on the board in turns or negotiate about path building. This is the mode we have used during the school assessment, described below. To increase difficulty they can add one or more of the enemy blocks by rolling the dice to decide enemy position; polarity is predefined on each board but players can reverse it to make the game easier or more difficult (preferably before starting the placement of the blocks).

Since each player has to conduct a rule-based plan based on their understanding of the game and combine their shared knowledge we consider this to be a manifestation of distributed computation. Through the âĂIJconsiderations, contingencies, and strategy formationâĂİ of multiple parties a distributed cognition of different knowledge resources is created10 and has to be maintained during the game to reach completion. Distributed computation thinking was indicated as one of the distinguishing properties of CT compared to computer science, according to the National Research Council7.

2.3.5 Competitive

Players are playing against each other and have to complete the path but starting from different directions. They roll the dice and can place as many bits as the number rolled. This mode demands very good planning, since making the wrong choice will render the board impossible to complete. Players get 1 point for every correct bit placed. They can change bits or blocks by deducting double the points of bits (e.g., 4 points for taking out 2 bits = 1 block). Taking off bits of the opponentâĂŹs path is forbidden unless she consents, in which case they both lose the exact number of points as the bits (e.g., 2 points each for taking out 2 bits).

The competitive mode demands efficient modeling skills since opponents would need to simulate the construction a large portion of the board with the available pieces, to avoid the cost of losing points and negotiating retraction of a previous block or bit placement. In a sense the effective combination of bit properties can be thought as algorithms that players need to compile during simulation in order to avoid these adverse consequences. For example, deciding which two bits need to be placed together in a way to accommodate locking the blocks with adjacent ones (i.e., having the magnets in the right position and polarity), and do this for a series of blocks is a complex cognitive process similar to compiling an algorithm and testing hypothesis (simulation). Both model building and simulation (forming and contrasting hypotheses) have been defined as foundational aspects of CT and revealed increased benefits compared to traditional methods of instruction.

2.3.6 Board Construction

Players can use all the bits or a random subset and make a custom path on an empty board. They have to do this by drawing the blocks on the board but without placing the bits. They will need to use the bits and blocks by creating a mental model of the path, taking each âĂIJusedâĂİ block out before moving on, until they believe they have drawn the whole path. The only requirement would be to start the path from an entry point on one side of the board and finish on an exit point at another side. Adding enemies would be optional to increase the difficulty of the game. The second player, or another group of players, can then use this custom board to play the game in either of the first two ways.

Like the previous activity, this one demands very good planning and ability to combine bit properties in order to make a playable board. Considering bits and their properties to be the data, players need to construct sets of pieces as part of the path using conditional logic during the process similarly to writing an algorithm (e.g., if I place this bit here, then this block has a minus on the bottom and needs a big square piece with a plus on the top center). This activity in particular demands the complex skill of combining different requirements to build a set of steps that will lead to an efficient solution to a problem (i.e., build a path from point

A to point B, using X number of pieces, and avoid Y enemies) and has been defined as procedural thinking - teaching concept abstraction into algorithms, a core concept of CT7.

2.3.7 Development of CT

RabBit EscApe satisfies most of the characteristics of the Operational Definition of CT for KâĂŞ12 Education as defined by the International Society for Technology in Education (ISTE). More specifically, students playing the game should be able to: 1) logically organize and analyze data, 2) automate solutions through algorithmic thinking (a series of ordered steps), and 3) identify, analyze, and implement possible solutions with the goal of achieving the most efficient and effective combination of steps and resources. Considering the game mechanics and suggested activities discussed previously, we can make the following arguments considering support of these three characteristics:

1. Bits have to be organized in blocks and block combinations that are meaningful according to bit properties and the printed pathâĂŹs form; pattern recognition is important in this process for identifying which bits create blocks that can fit the path printed on the board while also attracting adjacent blocks or repelling enemy blocks (demands analyzing the board based on possible bits combinations and organizing them on the path)
2. Game setup C demands that players devise some kind of strategy for correctly utilizing bits and matching the polarity and magnet position; for this purpose they need to come up with some kind of âĂIJrecipeâĂİ for putting bits together while drawing the path on the board, also accounting for the remaining pieces (demands some kind of procedural or algorithmic thinking)
3. In all game setups, especially A and B, players need to correctly identify the combination of blocks and bits by analyzing the pathâĂŹs comprising shapes and then simulate possible solutions for effectively completing the board, with the least number of bits and in the minimum time (efficiency).

Additionally, the game supports all dispositions and attitudes that are essential dimensions of CT, as expressed by the aforementioned definition which is pertinent to K-12 students: Confidence in dealing with complexity Persistence in working with difficult problems Tolerance for ambiguity The ability to deal with open-ended problems The ability to communicate and work with others to achieve a common goal or solution

3. EVALUATION AND OBSERVATIONS

We did not have sufficient time to complete a rigorous evaluation of RabBit EscApe. However we did run an informal pilot. As much as possible without violating observation protocols of the childrenâĂŹs school and our own institution, we followed BrandtâĂŹs anecdote and interrogator method. In order to protect our subjectsâĂŹ privacy, we took no notes during the actual session, but immediately thereafter wrote up the observations, and later in the same day, we interrogated each otherâĂŹs accounts. Our notes, including revisions from interrogation, are included in the Appendix (Informal Observation Interrogation-Augmented Anecdotes).

From our pilot, we learned how much this activity must be pedagogically scaffolded. Prior to attempting a full puzzle, we should familiarize the participants with the pieces by providing simple puzzles involving very few (2-4) pieces. Then, we would gradually increase the difficulty by giving similar, but subtly different pieces. By scaffolding the activity in such a way, the students would approach the challenging problem shown in Figure 1 with an understanding of the properties of the pieces as described above (this process is described in more detail under the first section of the Play Strategies).

Among many other interesting observations, one that we thought supported this scaffolded introduction, as well as the potential for RabBit EscApe to promote CT is captured in MichaelâĂŹs notes. Regarding the studentsâĂŹ process for searching, Michael noticed the students describing to each other the shapes they wanted for a particular position on the board. At first the students described only the shape of the piece; however, they soon learned that the piece placement was too tightly constrained to merely try to swap out pieces of the same shape. After all, they could not be sure whether it was the piece in question that needed to be replaced, or its surroundings. They then realized that they should specify the shape and also the polarities of the magnets; however, even this was an underspecification. Because similarly shaped pieces may have the same number of magnets, but in different positions within a piece, they had to also specify the magnet locations. Through a sort of discovery process, the students used incremental formalism to indicate what pieces they were looking for, and simultaneously to enhance their spoken representation of the pieces (and likely their mental models of them as well).

Additionally, Chris and Panagiotis, supervising another group, were very effective in leading students to evaluate the different properties of a missing piece. Questions like "What does this big piece need to have?âĂİ and âĂIJWhat do we know about it?" by pointing on the larger block on the board (at the bottom right of Figure 1b) prompted students to use conditional logic in identifying the properties and eventually the piece. More specifically, one of the students started to respond to these prompts by saying âĂIJIt is a big pieceâĂİ and then âĂIJit needs to have a plus magnet hereâĂİ, while pointing at the bottom where the ape is, and âĂIJit should have two more magnets on these sidesâĂİ, pointing at the top and bottom left where the large piece connects with the remaining path. Such instances were indications that students could grasp the conditions that need to be in place in order for the pieces to be placed successfully.

3.1 Discussion

Fabricating our own game was an interesting challenge. For future work, we might propose just such an exercise for high school or later students in a âĂIJshopâĂİ-style class. The records to keep track of which pieces have been cut, tapped, sanded, and implanted with magnets is a challenge in itself, but to have consistent results at the end, the use of various

custom jigs is important. To determine the kind of physical and process scaffolding needed to produce consistent sizes of pieces in a timely fashion is important and difficult. From RabBit EscApeâĂŹs theoretical design, and our informal observations, we believe it shows promise for teaching CT. In the Development of CT section above, we outline the various elements of CT that RabBit EscApe was designed to influence. Then in our Observation section, we detail observations that support our implementationâĂŹs successful realization of these design intentions.

4. CONSCLUSION

As CT continues to be discussed in academic and professional communities and until a more concrete definition emerges, a significant leap must be made to educate our communitiesâĂŹ youth as society continues to develop systems with which they must interact, whether or not they can understand them.

VT has embarked on a bold course to develop the CT capabilities of all its graduates. Simultaneously the US National Science Foundation is funding research into how to develop these skills in the broader population. While we think that VT can be effective in this pursuit, we believe that to truly change the CT of the population as a whole, we must have myriad interventions at various stages of an individualâĂŹs formal and informal education.

In support of education students at a younger age (than undergraduate students), we have developed two proofs of concept for how it might be possible to teach CT in a relatively explicit way. We believe that addressing CT directly, in conjunction with more subtle interactions such as integrating CT with core content in existing formal education curricula (Tatar, et al.) will best facilitate the learning of CT and its successful transfer and application to each individualâĂŹs area of interest or expertise.

4.1 Citations

Citations to articles [1, 3, 2, 4], conference proceedings [3] or books [6, 5] listed in the Bibliography section of your article will occur throughout the text of your article.

You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [5]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide*[5].

This article shows only the plainest form of the citation command, using `\cite`. This is what is stipulated in the SIGS style specifications. No other citation format is endorsed.

4.2 Theorem-like Constructs

Other common constructs that may occur in your article are the forms for logical constructs like theorems, axioms, corollaries and proofs. There are two forms, one produced by the command `\newtheorem` and the other by the command `\newdef`; perhaps the clearest and easiest way to distinguish them is to compare the two in the output of this sample document:

A *Caveat* for the TeX Expert

Because you have just been given permission to use the `\newdef` command to create a new form, you might think you can use TeX's `\def` to create a new command: *Please refrain from doing this!* Remember that your LaTeX source code is primarily intended to create camera-ready copy, but may be converted to other forms – e.g. HTML. If you inadvertently omit some or all of the `\def`s recompilation will be, to say the least, problematic.

5. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LaTeX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

6. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

7. REFERENCES

[1] M. Bowman, S. K. Debray, and L. L. Peterson. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.

[2] J. Braams. Babel, a multilingual style-option system for use with latex's standard document styles. *TUGboat*, 12(2):291–301, June 1991.

[3] M. Clark. Post congress tristesse. In *TeX90 Conference Proceedings*, pages 84–89. TeX Users Group, March 1991.

[4] M. Herlihy. A methodology for implementing highly concurrent data objects. *ACM Trans. Program. Lang. Syst.*, 15(5):745–770, November 1993.

[5] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[6] S. Salas and E. Hille. *Calculus: One and Several Variable*. John Wiley and Sons, New York, 1978.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title

(if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations

Display Equations

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LATEX; you do not insert it. You just add the names and information in the `\additionalauthors` command at the start of the document.

A.6 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

B. MORE HELP FOR THE HARDY

The acm_proc_article-sp document class file itself is chockfull of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LATEX, you may find reading it useful but please remember not to change it.