

Humans for EDA and EDA for Humans

Valeria Bertacco

Department of Computer Science and Engineering, University of Michigan
Bob and Betty Beyster Building, 2260 Hayward Avenue, Ann Arbor, MI 48109

valeria@umich.edu

ABSTRACT

Two misconceptions have been plaguing the electronic design automation (EDA) industry for decades: i) EDA solutions scale to larger complexities at an insufficient rate to keep pace with improvements in silicon designs; and ii) since EDA applications target silicon chip developments, the growth of EDA as an industry is bounded by the growth of the semiconductor industry.

With this paper we address these misconceptions and we argue that they can both be overcome. To this end, we overview a number of initial studies highlighting possible directions that EDA can pursue to (i) break off from its traditional ways of scaling solutions and applications to larger complexity, that is, by developing better heuristics for its complex algorithms. (ii) We also discuss alternative domains where EDA technology can be applied, beyond that of silicon design, so that the semiconductor industry is no longer the limit of EDA growth.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; B.6.3 [Logic Design]: Design Aids—*automatic synthesis, optimization, verification*

General Terms

Algorithms, Design, Verification

Keywords

EDA, Human Computing, Social Networks, Satisfiability

1. INTRODUCTION

The Electronic Design Automation (EDA) industry has been developing solutions to support silicon design for over 40 years. During this time, the scale and complexity of the problems that the industry could solve have made great strides: from supporting the development of the few-thousands-transistors chips of the 70's to that of the billion-transistors chips of the present day. However, EDA has been plagued for a long time (at least two decades) by a few misconceptions:

1. **The design complexity that EDA solutions can tackle is scaling at an increasingly slow and incremental pace.** This perception is mainly motivated by the increasing costs of silicon design developments. However, modern designs do not just have a larger scale than their previous generation counterparts. They also present a vast number of challenges that were not concerning just a few years ago, from parasitic effects, to clocking complexity, to hard-to-validate highly concurrent execution over complex communication protocols. Even if EDA is lagging in addressing and finding systematic solutions for these issues, over the past few decades it has delivered one of the best scaling trends ever observed in computing. Because of the needs outlined, however, we still ask whether we can do better (and the semiconductor industry demands it).
2. **The growth of EDA as an industry is limited by the semiconductor industry growth.** Because EDA is dedicated to developing tools and algorithmic solutions for the semiconductor industry, its market size can only grow if and at the rate that the electronics industry demand allows it. A common metric to evaluate this demand is to track the number of new ASIC design starts each year, which has suffered a persistent downtrend during the past decade, due to the steep increase in development and manufacturing costs of nanoscale-technology silicon chips [15, 20]. What we investigate in this work, though, is whether there are opportunities for EDA to serve other industries beside semiconductors, and thus unlock the potential for new growth.

With this work we explore some answers to the challenges discussed above. To address the first question, we investigate initial solutions that have the potential to break off the traditional practice of improving the scalability of EDA algorithms by relying on the performance improvements of the underlying computing hosts or by tuning and improving the algorithmic search heuristics in performance and quality. What we propose is to leverage human-computing and crowdsourcing solutions to EDA problems to develop completely different approaches that break off from our traditional way of thinking. Moreover, we want to learn from the way humans attack the complexity of these problems, and derive from it new approaches to developing automated heuristics.

To address the second challenge we look at possible additional domains to which EDA solutions can be applied. In particular, we consider the fast growing market of human-related and human-serving applications, such as social networks. Social networks are quickly expanding to provide a large number of services to humans. Their members are quickly growing in number, with Facebook leading at an estimated 845 millions registered members in 2011 [19], and more than 12 distinct social networks exceeding the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2012, June 3-7, 2012, San Francisco, California, USA.
Copyright 2012 ACM 978-1-4503-1199-1/12/06 ...\$10.00.

100,000,000 members level [27]. If EDA could provide services to these communities, or to companies that provide services to these communities, its applications would serve a much larger market than they are today.

The Design Automation Conference has witnessed a handful of studies in the past few years that have proposed alternative applications for the EDA field, for instance, by applying EDA solutions to the page ranking problem in web searches [12], and to social networks [7] (as we will discuss below).

2. HUMANS FOR EDA

Human computing is an approach to computing that has gained increasing interest in the past decade. In human computing, a computational process is carried out by outsourcing certain or all steps to humans. This computational model is effective when the tasks assigned to humans are those that are notoriously challenging for computers but straightforward and efficient for humans; examples of such tasks include image recognition, translation, *etc.* The term “crowdsourcing” is used when the process involves networks of individuals to achieve its goals. Humans are motivated to take part to the process by a wide range of “rewards”, ranging from monetary compensation, to enjoyment in game-like processes, to gaining online reputation, *etc.*

The goal in using human computing to solve EDA problems is to break off from traditional approaches for algorithm scaling. The hope is to find completely new approaches to solving a problem by leveraging skills that are typical of humans, or by developing new solutions based on the way humans attack the problem. The main challenge in pursuing this venue is that the complexity of the problems instances in EDA is often beyond what can be managed by humans.

An example of a human computing solution for EDA is FunSAT [6]. FunSAT is a visual puzzle game recently developed in our research group. The game presents the player with visual puzzles derived from instances of the SATisifiability problem [8]. The goal is for human players to leverage their unique visual reasoning and pattern recognition abilities to solve the puzzle, that is, find a satisfiable assignment for the SAT instance.

2.1 FunSAT single-player

FunSat has some similarity with a handheld electronic game, called “Lights Out”, a logic puzzle where the player manipulates a grid of buttons, some lit up and some off: every move lights up some buttons and turns off some others, while the player strives to organize his moves so to switch off all the lights [1].

In FunSAT, the game’s board includes (i) rectangular control buttons representing the Boolean variables in the instance and (ii) circles of varying size representing clauses. Users can click on variables to cycle through their Boolean value assignment (true, false, not assigned) while the game shows which clauses are satisfied (green), falsified (red) or still unresolved (gray). In addition, the game represents clauses with circles of size proportional to the number of literals they include, so that players can visualize the size of each clause, and intuitively gain a sense of how easy or difficult is to satisfy each of them (larger clauses include more literals and thus present more opportunities to be satisfied). FunSAT also uses varying gray color intensity for unresolved clauses to indicate how many unassigned literals are left in the clause; that way players can prioritize their attention in addressing unresolved clauses: the darker an unresolved clause, the most critical the situation, since there may be just one unassigned literal remaining. As a result, small, dark gray circles require top priority attention by a user: they correspond to clauses comprising just a few literals, most of which

have already been assigned.

Finally, players can zoom in and out in the game board to approach the problem by region, and can hover over a clause to see all of the variables it depends on. Figure 1 shows a screenshot of the game. At the beginning, all clauses are unresolved (grey), they become green when a partial assignment satisfies them, and red if they are falsified. To leverage the human ability of spatial perception and area, we lay out clauses in a grid. Variables surround the “clause grid”. Players advance in the game by levels, after solving an instance they are offered a more complex one in the next level.

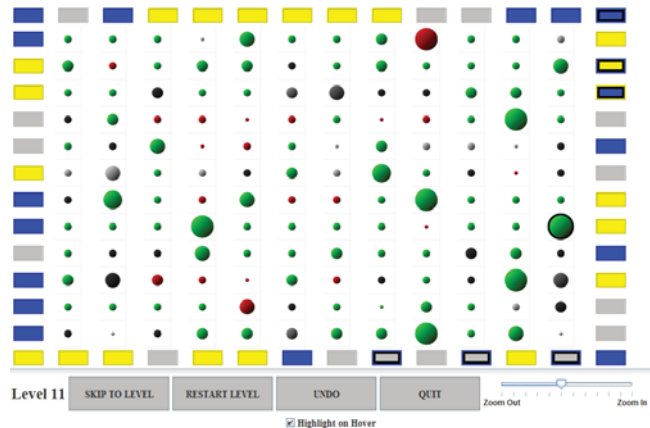


Figure 1: Screenshot of FunSAT - single-player. Circles represent clauses; rectangular buttons are Boolean variables controls. Assignments are applied by clicking a button; switching the corresponding variable from unassigned (grey), to blue (true), to yellow (false). Hovering over a clause highlights all the variables it depends on.

Through the game, human players can approach the challenge of solving SAT instances in a very unique way, compared to algorithmic solvers, using such skills as intuition and visual perception. As they click on different variables, they observe the visual impact on the grid of clauses and can progressively and intuitively tune their selection towards assignments that lead to a large fraction of satisfied clauses (visually perceivable by a higher fraction of green circles), until the entire grid is green, indicating that the instance has been solved. Classic SAT solving techniques, such as random restarts[16] and backtracking, are also naturally included in the game strategy, but with a “human twist”. For instance, backtracking is naturally used by players when they feel that they are at a dead-end corner of the search, and simply change the assignment of a few variables to move away from the situation. Learning also occurs when players identify color and shape patterns that are generated by their selection and use this visual learning in developing their game strategy. This game appeared online in 2009 [4] and it is implemented in Java.

2.2 Crowdsourcing FunSAT

Scaling FunSAT to large instances presents a challenge, as it is limited by screen real estate, human patience and humans’ ability to deal with complex problems. To address this aspect, we have recently developed a massively multi-player online version of this game [13]. To still leverage the visual intuition skills of humans we developed a new representation of the problem: clauses from a SAT instance are now presented in polar coordinates on a circle par-

tioned into one or more concentric sections. The controlling variables are placed along the outer perimeter of the circle (see Figure 2). This representation is much more compact for large instances, which typically have a large clauses-to-variables ratio. In addition, the game provides capabilities for zooming in and out conic sections of the circle, so that each player can visualize only a portion of the instance. Finally, a summary representation of the instance is provided in the fashion of a “world map” on the left side of the main game board, so that users can keep an eye on the overall situation at all times. This structure, along with the overall game controls is represented in Figure 2.

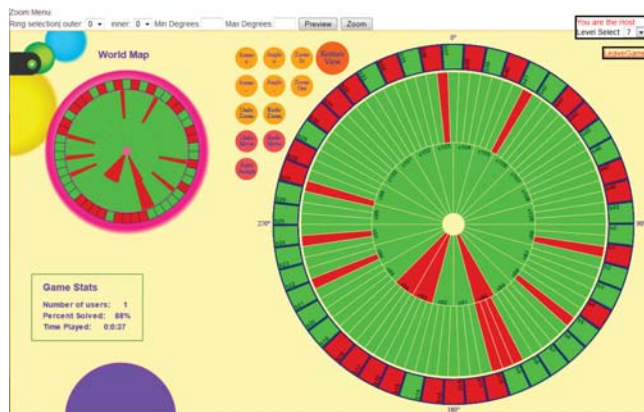


Figure 2: A FunSAT multi-player board. A game board is shown on the right with an assignment of variables (control buttons at the perimeter of the circle) that leaves 13 clauses falsified. Clauses are placed along two concentric inner circles. The smaller board on the left is a world map, useful when a player is zooming into a small portion of the game board. In between are several controls to zoom and rotate the board, create a random initial assignment and undo/redo moves.

The game setup lends itself to a wide range of multi-player interaction strategies. The one implemented in the first released version of the game is a *collaborative strategy*, where each user is assigned and controls a fraction of the instance’s variables and players are meant to interact with each other to coordinate their assignments and satisfy the instance. The game is organized in game rooms; within each room, a game leader controls when to start a new level and/or end the game; incoming players can choose to start a new game room or join an existing one. The game room management and the coordination of moves among players is coordinated by a central server, which can be setup by any entity that wants to provide the online game to its community. To communicate, users can use the chat window in the game’s GUI, or other chat and/or voice interaction mediums, such as Ventrilo [9].

Figure 3 is a snapshot of an advanced level of FunSAT multi-player, where clauses are laid out on several concentric circles (as shown in the world map) and one of the players is studying a conic section of the problem. Players can observe the dependency between clauses and variables by highlighting a clause or a variable. We tested this setup several times within our research group – in our case players could talk live to each other during the game – and found that collaboration enabled them to solve complex instances, of much larger size than those in the single-player version.

Other game strategies that we considered were *antagonistic strategies*, where the most effective players could over time take control

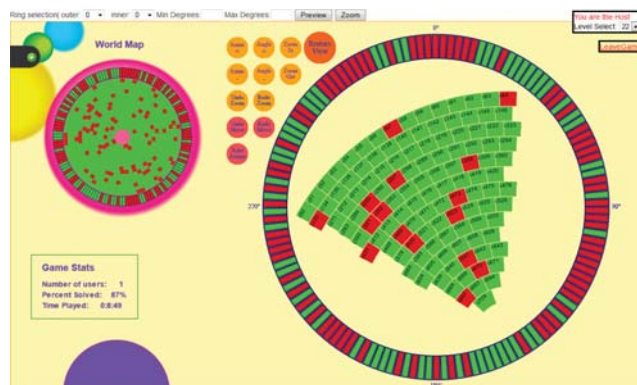


Figure 3: FunSAT multi-player, advanced game level. One the players in this game is studying a conic section of the board to evaluate if she should modify the assignment of the variables she controls or if she should negotiate an assignment modification with another player.

over other player’s variables. This approach would provide positive feedback and motivate players not only with the completion of a game level, but also by increasing a player’s variable control while he is solving a level.

The game is implemented in HTML5 [11] with a Python[17] back-end and requires a central server to synchronize and distribute moves across multiple players. One such central server is available at the University of Michigan and serves the players that connect to the game’s website at our institution.

3. EDA FOR HUMANS

The second misconception that we want to address in this work is that EDA as an industry is limited by the market’s demand of the semiconductor industry. In contrast, over the past decades, EDA has developed highly scalable and effective solutions to a number of problems that arise in many other domains. As discussed in the introduction, there has been a persistent downtrend of new ASIC design starts over the past decade: while this is a major source of concern in the industry, the shrinking size of the traditional EDA market could encourage companies to broaden their horizons and apply some of their solutions to alternative domains.

An area that has been growing at a very fast pace in the last decade is that of applications that provide services to humans: their social needs, their knowledge, their connection and communication needs. There are more than 100 social networks available online today, several of which have more than 100,000,000 registered members. Studying the characteristics and the connectivity of these networks would be a natural application for EDA’s formal verification solutions. Indeed, the typical membership size of a large social network is approximately equivalent to the number of states in a Finite State Machine (FSM) representing a digital system with tens to a few hundreds storage bits: a system with 100,000,000 states requires at least 27 bits. This is a size that is commonly tackled by formal verification tools.

It seems that formal verification tools would be apt to analyzing graphs representing relational aspects of a social network group and extract a wide range of collective information: how strongly connected the group is, or which are the closely knit subgroups, whether two members could indirectly communicate with each other, etc. Other tasks with a more direct commercial aspect to them in-

clude searching the graph for users with similar interests, activities or background (that could be represented by edges to specific types of nodes) for the purpose of job hunting, community organizing or even match making. A similar analysis could be carried out across multiple social networks to gather more information about a member than what can be extracted from a single network and/or validate his/her identity by comparison. Studying groups of members instead of individuals could benefit targeted advertising. For instance, a high connectivity to other network members is a good indicator that the member is an influential individual in the community and would be an ideal candidate to promote a new service or product. Min-cut algorithms [3], very common in place & route solutions, are valuable in this context to identify a subgroup with relevant characteristics for the desired type of advertisement.

We recently pursued a study in this space, proposing to leverage model checking tools to analyze social networks [7]. Model checking is an approach to formal verification of digital designs that has been gaining increasing momentum and scaling to complex designs, so that today it is possible to model check complex properties over fairly involved design units. The design under study is modeled as a FSM, where each state of the design is represented by one vertex of the graph, while edges represent valid transitions between nodes. To analyze social networks, we can construct similar graphs, with vertices representing members and edges representing relevant relational properties. We performed a first evaluation of this alternative EDA application by studying the relational properties of the customer base of a Twitter [22] application startup [21] (that is, the customers are a subset of Twitter's members). We described the aspects of the group that we wanted to check as formal properties described in Computational Tree Logic (CTL); we then verified them using NuSMV [5]. Our study considered a portion of the network including only approximately 50,000 members and was able to prove or disprove the properties in a matter of seconds.

After this initial study, we strove to push the scale of our analysis further. We considered a larger Twitter subgroup of approximately 40,000,000 members [14] as the target for model checking. The main issue we encountered was in dealing with the explicit representation of the graph, that is, file size (25GB) and memory access delays. To reduce the size of the representation we pre-analyzed the graph to extract cycles (subgraphs where each user can communicate to all other users, possibly through third parties) and abstracted them away. We also needed to explore some implementation optimizations to minimize the disk seek time. This allowed us to verify some of the simpler formal properties in this large subgraph. We are currently considering the application of min-cut to aggressively abstract the graph and then apply model checking iteratively at an increasingly fine granularity, so to only expand the relevant portions within the social network graph.

4. RELATED WORK

Amazon made available one of the first human computing solutions with its Mechanical Turk [2], inspired by the "Turk", a fake chess-playing machine from the 18th century. The Turk was built to appear as an automaton that could play chess, while in reality it hid a human chess master inside who would control the machine. Similarly, Amazon's Mechanical Turk leverages a pool of human users to solve problems of disparate nature. The Mechanical Turk consists of task listings, ranging from categorizing products to writing articles, and offering a wage to anyone who completes a task: It appears to be an effective way to motivate a large group of people to perform menial, yet essentially human tasks.

Recently, a number of online human computing processes have also gained popularity, including the ESP game [24], re-captchas

[23] and duolingo [26]. All these solutions leverage humans' desire to play fun games that use their skills in image recognition and language translation, tasks that are extremely challenging for computers. The ESP game [24] is an internet game asking players to tag images: by collecting several tags from distinct individuals for each picture, the game can provide high quality tags for a large number of pictures available online, and thus greatly boosts the quality of image searches. Re-captchas [23] leverage humans to type words from scan images of books where character recognition software has failed. Finally, duolingo [26] uses human computation to translate documents between languages. To provide an estimate of the amount of human computation cycles available, von Ahn [25] reports that in 2003 humans have collectively played 9 billions hours of computer solitaire.

Complementary to these efforts, recent research has attempted to infer the computational model of the human brain [10], and has found that this model can deliver notable advantages for certain tasks (such as image recognition) over traditional computation. In addition, a recent work [18] proposes to leverage functional magnetic resonance imaging (fMRI) techniques to observe the brain activity in digital designers while at work. By studying the observed neural patterns, the authors hope to boost designers' productivity by developing better learning techniques to support their training and by selecting the most promising talent.

On the front of pursuing alternative venues and applications for EDA's solutions and algorithms, a few works have appeared in recent years at the Design Automation Conference. As an example, [12] proposes to apply the algorithmic solutions developed for parasitic extraction in silicon designs to page ranking in web searches.

5. CONCLUSIONS

In this paper we outlined approaches to overcome two classic misconceptions in EDA: (i) that scaling in EDA has been improving at a slower pace in recent years, insufficient to tackle the demands of the semiconductor industry; and that (ii) the application of EDA's solutions is limited by the needs of silicon design developments. We have shown that bringing humans into the equation has the potential to overcome both these issues.

On the first front, human computing can provide and inspire new ways to solve difficult algorithmic challenges and break the scalability barrier. Along this direction we overviewed an example solution that employs humans to solve SAT problem instances by presenting them as a game. On the second one, EDA's solutions and tools appear to be well positioned to solve large scale challenges in domains beyond semiconductors, such as those that directly benefit humans as a group. Specifically, we discussed the application of formal verification techniques to the study of social networks.

Overall, both these research directions have just began to attract the attention of a handful of EDA researchers, and we believe that much more can be gained by pursuing them further.

Acknowledgements

The author would like to thank her student Andrew DeOrio who led much of the effort in devising the solutions outlined in this paper, namely FunSAT [6], both single-player and multi-player, and the application of EDA to social networks [7]. She would also like to thank Erica Christenssen and John Krzemien for developing high quality online implementations of the FunSAT game, while overcoming the numerous issues and hurdles that are always encountered between an idea and its publicly available implementation. Finally, Joshua Lim worked on analyzing a large Twitter membership database to validate a number of formal properties.

6. REFERENCES

- [1] Addicting Games. Lights out. www.addictinggames.com/puzzle-games/lightsout.jsp.
- [2] Amazon. Mechanical turk, 2008. www.mturk.com.
- [3] C. Chekuri, A. Goldberg, D. Karger, M. Levine, and C. Stein. Experimental study of minimum cut algorithms. In *Proc. of Symposium on Discrete Algorithms (SODA)*, pages 324–333, Jan. 1997.
- [4] E. Christenssen, A. DeOrio, and V. Bertacco. FunSAT - single-player, 2009. www.funsat-single.eecs.umich.edu.
- [5] A. Cimatti, E. M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV version 2: An opensource tool for symbolic model checking. In *Proc. International Conference on Computer Aided Verification (CAV)*, pages 359–364, July 2002.
- [6] A. DeOrio and V. Bertacco. Human computing for EDA. In *Proc. of the Design Automation Conference (DAC)*, pages 621–622, June 2009.
- [7] A. DeOrio and V. Bertacco. Electronic design automation for social networks. In *Proc. of the Design Automation Conference (DAC)*, pages 621–622, June 2010.
- [8] N. Een and N. Sörensson. An extensible SAT-solver. In *Proc. of the International Conference on the Theory and Applications of Satisfiability Testing (SAT)*, pages 502–518, May 2003.
- [9] Flagship Industries, Inc. Ventrilo. www.ventrilo.com.
- [10] J. Hawkins and S. Blakeslee. *On Intelligence*. Times Books, 2004.
- [11] HTML Working Group. *HTML 5 Reference*. World Wide Web Consortium (W3C), Mar. 2009.
- [12] V. Jandhyala. Physics-based, field-theoretic design automation tools for social networks and web search. In *Proc. of the Design Automation Conference (DAC)*, pages 280–281, June 2011.
- [13] J. Krzemien, A. DeOrio, and V. Bertacco. FunSAT - multi-player, 2011. www.funsat.eecs.umich.edu.
- [14] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proc. of the International Conference on World Wide Web (WWW)*, pages 591–600, Apr. 2010.
- [15] B. Lewis and G. Ramamoorthy. Market trends: Worldwide, ASIC and ASSP design starts continue declining trend, 2012. *Gartner Research*, Feb. 2012.
- [16] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proc. of the Design Automation Conference (DAC)*, pages 530–535, June 2001.
- [17] Python Software Foundation. Python programming language. www.python.org.
- [18] M. Potkonjak and F. Koushanfar. (Bio)-behavioral CAD. In *Proc. of the Design Automation Conference (DAC)*, pages 351–352, June 2008.
- [19] E. Protalinski. Facebook has over 845 million users. *ZDNet*, Feb. 2012.
- [20] K. Shuler. The three consequences of fewer design starts. *System-Level Design (SLD)*, Mar. 2012.
- [21] Twilk Inc., 2012. twilk.com.
- [22] Twitter Inc., 2012. twitter.com.
- [23] L. von Ahn. Games with a purpose. *IEEE Computer*, 39(6):92–94, June 2006.
- [24] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. Conference on Human Factors in Computing Systems (CHI)*, pages 319–326, Apr. 2004.
- [25] L. von Han. Human computation. *Google Tech Talks*, July 2006.
- [26] L. von Han. Massive-scale online collaboration. *TEDxCMU*, Apr. 2011.
- [27] Wikipedia. List of social networking websites, 2011.