# Human Computing for EDA

Andrew DeOrio and Valeria Bertacco

Department of Computer Science and Engineering, University of Michigan
{awdeorio, valeria}@umich.edu

## ABSTRACT

Electronic design automation is a field replete with challenging – and often intractable – problems to be solved over very large instances. As a result, the field of design automation has developed a staggering expertise in approximations, abstractions and heuristics as a means to side-step the NP-hard nature of these problems. Approximations and heuristics are at heart a natural application of human reasoning. In this work we propose to harness human potential to solve some of these problems. Specifically, we propose FunSAT, a massively multi-player puzzle game for SAT solving. FunSAT leverages visual pattern recognition skills, abstract perception and intuitive strategy skills of humans to solve complex SAT instances. Players are motivated by the puzzle-solving challenges of the game and by its social interaction aspects.

## Keywords

Satisfiability, Verification, Human Computing

## 1. INTRODUCTION

Despite great advances in computer hardware and algorithm design, many Electronic Design Automation (EDA) problems are still beyond the reach of available computing systems, because of their intractable complexity and large scale. In this work, we propose to use human-powered reasoning to tackle these complex problems.

Compared to computers, humans rely on very different approaches to problem solving. There are a number of activities in which humans are more skilled than computers, and they tend to attack problem solving with the skills in which they are most comfortable. As an example, humans can easily recognize visual images and patterns and, in general, they are skilled in visual reasoning. People can identify musical melodies and use creativity to generate new ones; they can cleverly apply adaptive abstraction techniques to problems, often without even being aware of the specific technique that they are using. All of these are tasks in which computers perform very poorly. In the game of Go for instance, human players, with their intuition and propensity for pattern recognition, perform consistently better than software agents.

Not only do humans excel in a number of unique areas of intelligence, but they frequently expend this cognitive potential playing computer games: a recent study estimates that 34% of internet-connected adults play online games regularly [1]. Massively Multiplayer Online (MMO) games offer the additional attraction of a social component to gaming: the number of players registered with “World of Warcraft”, a popular role-playing game, recently reached 10 million [3]. Some games naturally resemble problem solving tasks in EDA problems. For instance, “Pipe Dream”, a popular game from the 1980s, requires players to arrange pipe segments on a grid. Pipe Dream is clearly reminiscent of routing wires in physical synthesis. The vast gaming population provides a rich opportunity to harness human intelligence to perform useful tasks, provided that they are presented as games.

In this work, we propose to leverage human skills such as visualization, abstraction and strategy to solve complex EDA problems. In order to achieve this goal, we need to present the problem in a format that is intuitive and amenable to human skills. Moreover, we need to cast the problem as a game, where players are rewarded for solving a puzzle. Typically, EDA problem instances are complex, often containing millions of primitives or solution paths. To address this complexity, we devise two strategies: (i) problem partitioning and abstraction, and (ii) our games should be multi-player. In a multi-player game, players coordinate their activity as a team to achieve a goal that is beyond the reach of any individual. The reward for solving a puzzle (a sub-problem) could range from advancing levels in the game, to the opportunity for socialization by interaction with other players.

Several CAD problems are suitable for games. Beside “Pipe Dream”, discussed earlier, placement could be cast as a packing puzzle, possibly similar to the game of “Tetris”. Both of these tasks are naturally visual and have potential for providing more effective solutions than machines. Indeed, engineers today routinely intervene in the place and route processes to correct and solve sub-problems that machines cannot tackle. In this work, we focus on SATisfiability solving, a central problem in EDA known to be NP-complete. While available SAT solvers can solve many complex problem instances, other instances often arise that are beyond the reach of software solvers, due to either the sheer complexity of the instance or the narrow solution space. Typical solvers employ a wide range of heuristic approaches, strategy, randomization, *etc.* We propose presenting SAT instances in visual and abstract ways, so that human players have the potential of finding solutions where software solvers fail.

## 2. HUMAN COMPUTING TODAY

Few applications have been recently proposed for human computing. Von Ahn [7] leverages human computing to index web images: he has created a game in which a player’s goal is to tag pictures. Images are shown to users who suggest associated words, which are then tabulated for popularity and later used as search terms. A more nefarious example of human computing is in solving captchas, a popular security measure on the web that strives to ensure that a user is indeed human. Captchas leverage computers’ lack of skill in visual recognition to block software applications from accessing a site. To counter this security measure, spammers have developed infrastructure to have other humans solve captchas on other sites by enticing them with incremental rewards [2]. Other applications of human computing are set forth in [6], including language translation and text summarization, among others.

Complementary to these efforts, recent research has attempted to infer the computational model of the human brain [5], and has found that this model can deliver notable advantages for certain tasks (such as image recognition) over traditional computation [7]. Finally, Amazon’s Mechanical Turk [4] operates in the space of crowdsourcing by also leveraging a pool of human users to solve problems. However, in this case problems are not camouflaged as


*DAC 2009,* July 26 - 31, 2009, San Francisco, California, USA.

games and there is no explicit attempt to cast them so that humans have a direct advantage over a machine. 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. Absent from these applications are the intractable problems extremely common in EDA, some of which have great potential for human computing.

# 3. A HUMAN-POWERED SAT SOLVER

FunSAT is a visual puzzle game. It transforms SAT instances into puzzles and presents them to human players who solve them. In developing FunSAT, we strove to present the problem in a way that leverages the unique visual reasoning and pattern recognition abilities of humans. SAT instances are presented visually and hierarchically, similarly to a map, where a player can zoom in and out to approach the problem at different levels. We envision the possibility of structuring the game into levels so that a player at first is only allowed to zoom out a certain amount; once players have solved local portions of the puzzle, they are promoted to the next level, where they can access a larger region of the problem.

Figure 1 shows a screenshot of the game. Boolean variables are mapped to buttons, which can be assigned to true (purple), false (yellow) or unassigned (grey) with a click of the mouse. Clauses are represented by circles whose size is proportional to the number of literals in the clause, so that players can intuitively rank the difficulty of satisfying each clause. At the beginning of the game, all clauses are undetermined (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” and the relationship of variables to clauses is shown by hovering the mouse over a variable, highlighting the affected clauses.

We implemented a prototype of FunSAT in LabVIEW. As the game progresses, players click on different variables, changing their values, while observing the color impact in the grid of clauses. The final goal is to light the entire grid with green, thus satisfying all clauses. In attempting different assignments, humans learn color and shape patterns that are generated by different selections and develop gaming strategies based on this. FunSAT has some similarity with a handheld electronic game called “Lights Out”, a logic puzzle where players must switch off all the lights in the game by manipulating a pool of buttons.

## 3.1 Visual Aids

To help humans reach the goal of satisfying all the clauses, we implemented a number of visual aids, including an intuitive layout scheme, informative feature sizing and smart reminders of recent activity. A clever clause layout could suggest which clauses are affected by each variable, thus helping the player develop a winning strategy. To this end, during the construction of the visual problem interface for an instance, we arrange clauses sharing the most variables in close physical proximity.

Puzzle solving is bound to encounter dead ends: logging of past selection activity provides users with backtracking capabilities. This enables users to revisit the past few variable assignments, as well as providing an “undo” function.

## 3.2 Scaling

Scaling FunSAT to large instances presents a challenge, as the game limited by screen real estate and human patience. We propose an approach inspired by navigable maps: users can work on the instance at different levels of detail by simply zooming in and out. At small scale levels, users work with clusters of clauses and variables. At the same time, a small inset abstract view of the entire SAT “globe” provides information on the system-level impact of the selections. This hierarchical approach enables us to enhance the game in the multi-player direction: multiple people can operate in different areas of a same “map” and coordinate their efforts to reach a common goal. In order to minimize the impact of selections made in one part of the map to other regions, instances could be partitioned using a min-cut procedure. This minimizes the number of shared variables across different regions of the SAT "globe".

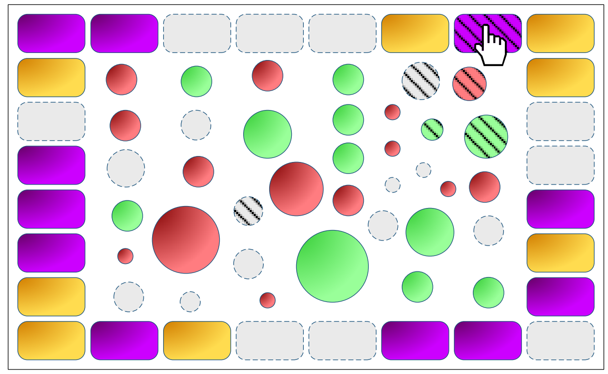

**Figure 1: Snapshot of FunSAT.** Circles represent clauses and the surrounding buttons correspond to variables. Assignments are applied by clicking on a button, which switches from unassigned (grey), to purple (true), to yellow (false).

# 4. CONCLUSIONS AND FUTURE WORK

We propose bringing the art of human computing to the science of SAT solving. Leveraging the human passion for games, along with its propensity for visual reasoning and pattern recognition, FunSAT presents SAT instances as a visual challenge amenable to these strengths. In addition, it has the scaling potential to realize a large multi-player online game, where players collaborate in solving the same instance. In the future, we hope to further strengthen player motivation by providing social networking benefits for successful players. Finally, casting EDA problems as games will allow unskilled players, to solve complex engineering problems with a low barrier to entry. Hopefully, a byproduct of this area of research will be an increased appreciation for engineering tasks in the general population.

# 5. REFERENCES


[1] Online game playing is still the most popular online activity. *Associated Content*, 17 August 2007. `www.associatedcontent.com`.

[2] PC stripper helps spam to spread. *BBC News*, 30 October 2007. `www.news.bbc.co.uk`.

[3] WoW surpasses 10 million subscribers, now half the size of australia. *Joystiq*, 22 January 2008. `www.joystiq.com`.

[4] Amazon. Mechanical turk, 2008. `www.mturk.com`.

[5] J. Hawkins and S. Blakeslee. *On Intelligence*. Owl Books, 2005.

[6] L. von Ahn. Games with a purpose. *IEEE Computer*, 39(6):92–94, June 2006.

[7] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *ACM SIGCHI*, pages 319–326, 2004.