Scarecrow: If I only had AI*

David P. Miller,† Jacob Q. Milstein‡ & Cathryne Stein§

October 30, 2006

Abstract

At the 1992 AAAI robot contest one of the top finishers was Scarecrow – a robot that had no computer in the traditional sense, was built out of less than $200 of parts, and was explained and operated by a five year old. The designers sought to demonstrate the capabilities and competence that can be accomplished by using a strictly reactive architecture for well defined tasks such as that contest. This paper reexamines the Scarecrow robot and puts it into historical context. With fifteen years of perspective, we can also see what Scarecrow has to say about the perception of intelligence.

1 Finding the Tall Poles in a Problem Space

In 1992 AAAI held the first of its annual robotics contests. The contest that year was to find, and then visit in order, a set of tall poles in a physical problem space. A set of ten 10' tall 4" diameter pieces of PVC were placed vertically inside of a 40' diameter arena. Inside the arena were cardboard box obstacles, approximately $1 \times 2 \times 3$ foot set on their 1×3 side. Approximately fifty boxes were in the arena. The challenge to the robots in round one was to visit each of the poles and ID each pole (or create a unique ID for that pole). In round two robots were required to visit, in order, a set of three poles (selected by the judges) that the robot had previously identified.

Teams were allowed to place identifying markers (e.g., bar codes) on the poles in advance. A robot was judged to have visited a pole if it correctly identified a pole while simultaneously coming within two robot diameters of that pole. A robot successfully avoided an obstacle if it turned away from an obstacle rather than attempting to plow through it.

*with apologies to Mssrs. Harburg & Arlen: I could suck up amps of power; Computing for many hours; Making models verify; And my inferences would be seizing; While my memory was GC'ing; If I only had AI

†School of Aerospace & Mechanical Engineering, University of Oklahoma, Norman, OK. dpmiller@ou.edu

‡Wesleyan University, Middletown, CT

§KISS Institute for Practical Robotics, Norman OK

2 Scarecrow

In addition to Scarecrow, a dozen robots were entered into the contest. All but Scarecrow had many thousands of lines of custom code. All but Scarecrow had in excess of $10,000 in hardware. All but Scarecrow had institutional support from a university or research lab. All but Scarecrow were the product of several work years of effort.

The Scarecrow robot was built in the month before the contest at a cost of about $200. To the extent that it processed, it had all of its processing onboard. The primary operator's affiliation at the time of the contest was as a student at the Leslie-Ellis Pre-School of Arlington, MA.

2.1 The Scarecrow Challenge

The Scarecrow entry had several objectives which were not necessarily shared by the other entries or the contest organizers. In particular, some of the objectives of Scarecrow were to:

1. Create a robot that would actually do something that could be understood and would be exciting to elementary school students.
2. Demonstrate that, for many robot tasks, sophisticated hardware is not required.
3. Be a validation experiment for the theory that intelligence is not a state of mind, but a state of being (or perhaps more accurately, a state of being observed); intelligence is attributed to an agent by observers who perceive the agent 'intelligently' interacting within the environment where the agent is placed.
4. Accomplish the contest tasks without: sponsorship; graduate students; transistors; or much work.

Scarecrow was built during the month before the contest. It was completed the day before going to San Jose. While all the electronics and mechanics were tested and (mostly) debugged, the robot was never run in an arena-like environment until arrival in San Jose. Given this grievous lack of system level testing, the robot's overall level of performance during the contest was outstanding.

2.2 Related Work

> *Two hundred years ago people compared the brain to the workings of a clock. Today they say it is like a computer. In 1950 they probably said it was like a washing machine.* – Roger Schank, 1982.

Over the past three hundred years, engineers in Europe, Japan and elsewhere built automatons that ran on clockwork. In 1774 Henri-Louis Jaquet-Droz demonstrated a mechanical automaton capable of writing any phrase up to forty characters in length (Hiller (1976)); Around the same time in Japan, karakuri automatons such as the tea server (Takanashi (2002)) used whalebone springs and wooden gearboxes to create mobile systems that moved in programmable paths and could react to objects in their environment. These systems and many others showed that you could get complicated, and even reprogrammable, behavior out of purely mechanical systems.

Modern digital computing traces its heritage to mechanical systems such as the Jacquard loom, Kopplin (2002), and the analytical engine created by Babbage (see Hyman (1985)).

These are all incredibly innovative and complicated devices. However, despite their complexity, no a-priori knowledge of physics, electronics or even math is needed to understand how these devices work (though such knowledge was undoubtedly useful in their creation). One can sit down with such a device (or an inexpensive plastic replica (e.g., The Karakuri Corner (2006)) and spin a gear or throw a lever and see how it all works together. People interact with the world in a mechanical fashion and an intuitive understanding of many mechanical reactions just seems to be part of the human condition. To those not well schooled in the underlying physics and logic, an integrated circuit qualifies as a *...sufficiently advanced technology [which] is indistinguishable from magic,* Clarke (1962). So to keep Scarecrow accessible to those without a technical background, mechanical, rather than electronic, technology was emphasized in Scarecrow's construction.

Figure 1: *Scarecrow all dressed up for a night out in the arena*

2.3 The Design of Scarecrow

Scarecrow (see Figure 1) was built, in part, to demonstrate the capabilities and competence that can be accomplished by using a strictly reactive architecture for well defined tasks such as the 1992 contest.

When in operation, the robot heads off in a random direction until it senses an obstacle[1]. Poles and obstacles were distinguished by height and labeling. Poles were labeled with a conductive bar code (strips of steel wool wrapped around the pole in a pattern representing a 4-bit number). Scarecrow showed that it had "read" this bar code by displaying the number (in binary) on its front panel. This was its method of identifying to the judges that it had found and identified a pole.

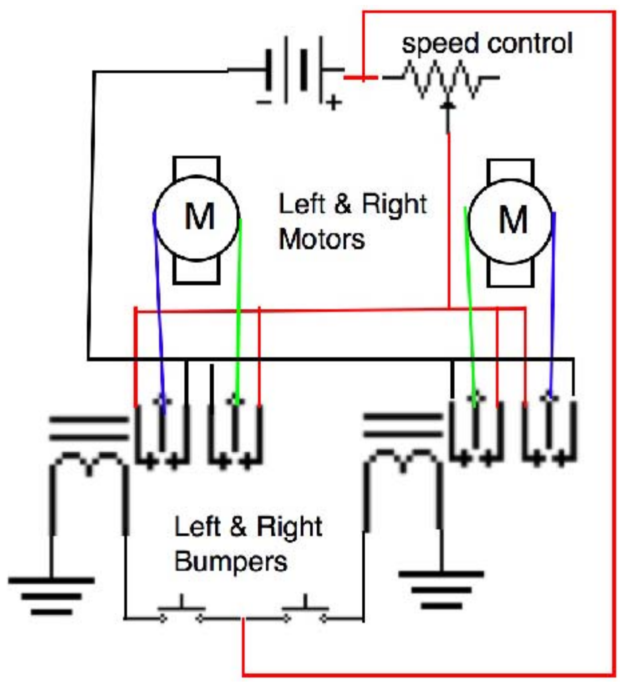

Figure 2: *The left and right bumpers simply fired a DPDT relay reversing their respective motors.*

The act of encountering an object (be it pole or obstacle) was done with sufficient enthusiasm to cause the robot's bumper to close a switch – energizing a DPDT (double pole, double throw) relay which reversed the direction of the opposing wheel – causing the robot to spin in place (see Figure 2). Scarecrow's hemispherical bumpers kept the switch closed until the robot had cleared the object, whereupon the switch was released and the robot continued its forward progress along a new heading. Mercury switches were used to soften acceleration and deceleration before and after impacts, and to act as an emergency shutoff during tipping.

The robot's size and speed were designed to be able to sweep out the entire arena area more than twice during the allotted twenty minute period. The result of this random walk was quite good and the robot actually explored approximately 80% of the arena during each of its runs.

Scarecrow maintained almost no state information. For it to visit the objects in the correct order, it was

[1]For Scarecrow, obstacle detection and collision detection were one and the same

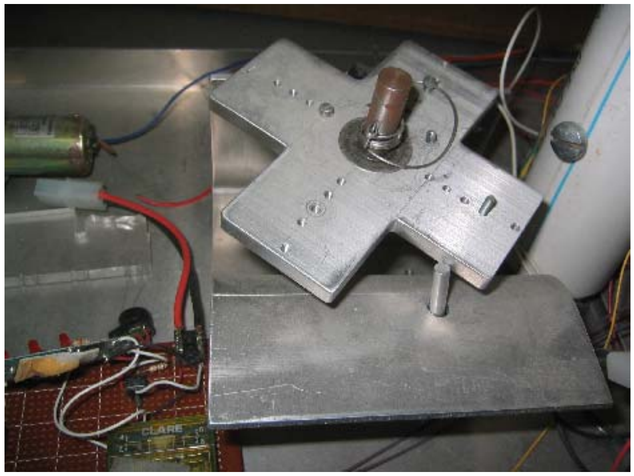

Figure 3: *The robot's four nibbles of ROM were set using screws; the 'program counter' advanced when the solenoid withdrew the restraining pin allowing the spring to turn to the next nibble.*

"programmed" (using a screwdriver) by the operator with object IDs in the desired order (see Figure 3). It used a simple (spring powered) finite state machine to keep track of which object it had last seen, and which it wanted to see next. If it came across the correct object, it recognized it (by ANDing the bits in ROM with those from the bar code reader) and announced (beeped) it had found the desired pole.

Figure 4 shows a schematic of the object recognition circuit. The four switches (bottom one is marked with a circle) along the right hand side represent the barcode reader made from conductive rings. In the figure the barcode reader sees 1010. If one of the bits from the barcode reader is a 1 then that powers a relay (bottom relay is marked with a square), which sends V_{cc} along one path. The column of SPDT (single pole, double throw) switches (bottom one marked with a triangle) represents the programmed bar code that the robot is searching for. The switches are depressed if a screw has been inserted in that bit location, or otherwise in the up position. If the memory bit and the reader bit are the same, then power is applied to that bit's SPST (single pole, single throw) relay (marked with a pentagon). In the figure the 1's, 2's & 8's bit all have agreement between the pattern in memory and that being read by the bar code reader. However the 4's bit in memory is a 1 while the bar code reads a 0 in that location; so the SPST relay is not powered and the circuit leading to the buzzer is not closed. When all four bits match, then the circuit to the buzzer is closed signaling that the pole has been found. A solenoid that is in parallel with the buzzer fires allowing the program counter to advance to the next pole ID code.

Figure 5 shows the circuitry dedicated to reading, interpreting and displaying the pole ID from the continuity bar code reader mounted on the robot's top (see Figure 6). This part of the system also worked quite well.

2.4 The Scarecrow Advantage

The major advantages of Scarecrow were:

1. All processing was onboard: there were no delays due to communications and the robot did not have

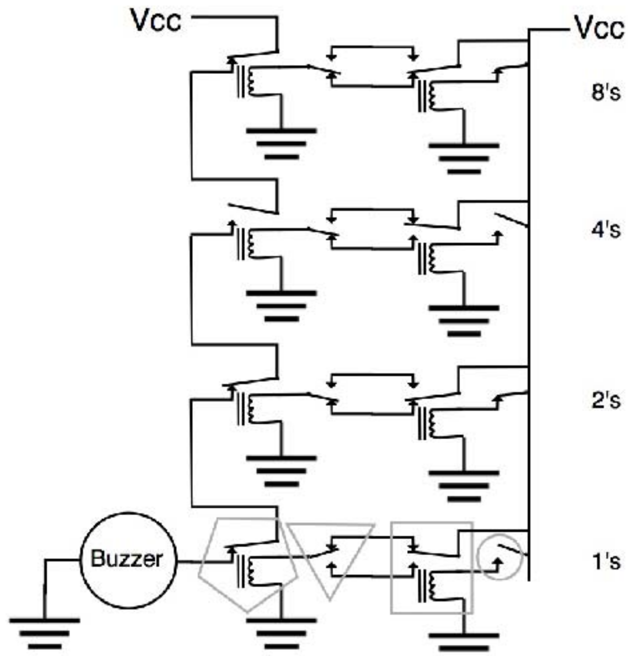

Figure 4: *The robot is looking for pole 1110; the bar code reader currently sees pole 1010.*

problems with external RF interference. The brains on-board technique was only used by two other contest entrants.

2. There was no computer: The robot could not suffer from software bugs, and a major source of hardware problems was eliminated.

3. All of the robot's sensors and effectors could be completely debugged using a multimeter. There were no encoded signals, waveforms, timing signals, etc to worry about.

4. The robot required less than two hours to unpack and set up. There was another one or two hours during the contest for mods and maintenance (battery charging), but the simple design was very robust and all members of Team Scarecrow had quite a bit of free time and at least 8 hrs sleep/night.

5. The design of the robot had a certain sense of uncertainty and danger which generated suspense and elicited great reactions from the audience (see Section 3.2).

For the contest, Scarecrow's major disadvantage, when compared to the other entrants, was the physical contact requirement for its pole ID sensor. There were some dramatic near misses, and the robot's score might have improved significantly if it could have read poles over a longer range.

3 Conclusions & Final Thoughts

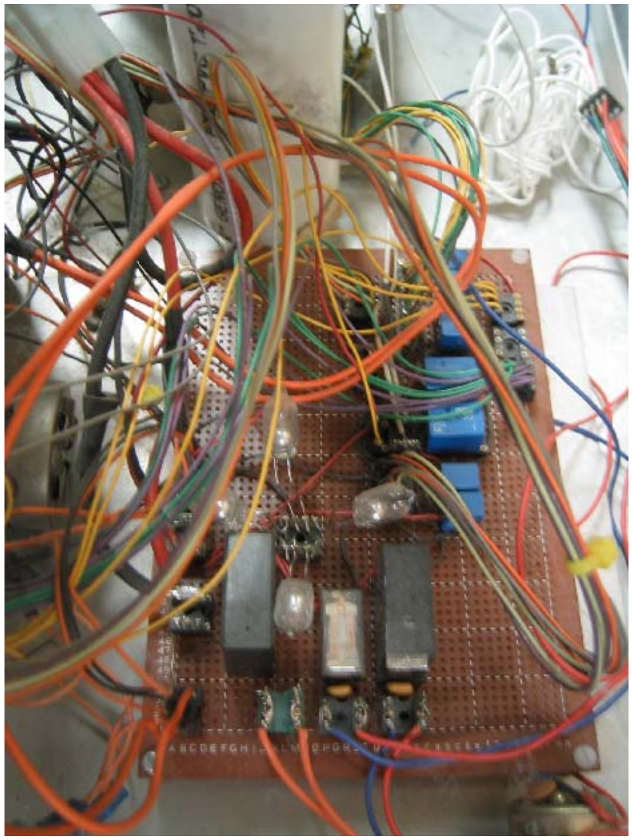

Figure 5: *Scarecrow's logic board consisted of relays to perform a 4-bit* **and** *between the bar code reader and the ROM. Mercury switches were used to cut power to the motors if the robot tilted too much.*

Figure 6: *Bar codes, made of steel wool, shorted out certain pairs of rings when the robot ran into them.*

3.1 AI's role in robotics

There have been many substantial advances in AI over the past forty years. Many of these advances have been incorporated into software and hardware that is widely used. However, in the quest for the general purpose robot, there has been less demonstrable success. While much research and some successful technologies have come out of general purpose robot research, the ultimate goal of a *Rosie the Robot* or *Commander Data* seems as far or farther away today than it did when the research first started. The reasons for this are obvious in retrospect: Making anything general purpose is hard – and goes against the natural evolutionary tendencies of specialization and optimization. The few example species in nature of animals that flourish in almost all environments (e.g., rats and humans) do not inspire confidence that such generalized capabilities are beneficial for the overall system in the long run[2].

Team Scarecrow prefers to think of robotic systems as more akin to expert systems. A robot should be designed for a specific task or set of tasks. When the scope of the robot is limited, reactive behavior control techniques have been quite successful. This is especially true when the hardware is developed in concert with the software. Scarecrow pushed this model of robotics to the limit. Scarecrow was specifically designed for this one particular contest, and is good for little else other than as an educational example. As an educational tool, Scarecrow is excellent. This is because the robot's reactions are incredibly simple. Braitenberg described how very simple reactions (e.g., a light sensor controlling a motor's speed) can be combined to exhibit behaviors that appear sophisticated (Braitenberg (1986)). Scarecrow is a Braitenberg vehicle with a mission.

While Scarecrow was a successful robot, it was not a good demonstration of AI software techniques – demonstrations of which were one of the major motivations for having the AAAI robot contest in the first place. Two other robots in the contest had all of their computation onboard. They were also the two robots who consistently outscored Scarecrow in the contest. They also had much more AI, in the traditional sense, then did Scarecrow. Table 1 illustrates some of the differences between these three robots.

Flakey and CARMEL outscored Scarecrow, but not by all that much. For 10% of the cost of either of those robots, a fleet of Scarecrows could have been created that in the first round would have found and identified all of the poles in a few seconds (since the arena would literally be packed with robots). The second round would have been trickier to do with a fleet of Scarecrows, but a single Scarecrow accomplished the task and did well enough that with an improved first round score it would have easily been the overall winner.

To ensure that there was more *traditional* AI in future contest robots, in 1993 the contest required the use of vision as one of the entrance rules. This resulted in more traditional (and slower) AI dominated robots. However, despite the continued inclusion of vision oriented tasks, within a couple years reactive and low (traditional) AI robots quickly dominated the contests by making use of specialized hardware such as Sargent *et al.* (1997).

When we were asked, at the 1992 contest, what we would do to encourage more AI in the contests, one of us (Miller) suggested making the task more realistic and relevant to a wider audience by picking a task such as vacuuming. While this suggestion did result in some interesting workshops (see Bonasso & Miller (1993)) the *Roomba*, (iRobot Corporation (2006)) is a strong existence proof that such tasks (despite the authors' thoughts at that time) do not require much AI, in the classical use of that term. The *Roomba* is also evidence that it is much easier to create reliable task-specific robots than general purpose robots. While there are many similarities in both the hardware and the software, the *Roomba* vacuuming robot is distinctly different from the *Scooba* floor washing robot. By creating a task specific robot the robot never has to decide whether it should be mopping or vacuuming a particular floor; it can assume that it is meant to do its 'thing' on whatever floor it is placed.

[2]Some readers have pointed out that this grouping may not be fair to rats.

Feature	CARMEL	Flakey	Scarecrow
Sensors	24 sonars around perimeter grayscale CCD camera	12 sonars at front, back, sides 8 touch sensitive bumpers structure-light sensor	2 bumpers 4-bit conductive bar-code reader
Software Structure	hierarchical modules	parallel behaviors	none
Moving: obstacle avoidance roaming	 EERUF and VFH 8-point start	 fuzzy rules and LPS composite behaviors: wander, avoid obstacles go-forward	 turn after contact go forward
Object recognition	tagged poles (bar codes) long range vision	type recognition of poles (no tags added) two-part identification: sonar identifies candidates structured light sensor recognizes poles	tagged poles (bar codes) physical contact
Mapping map design position correction	 global Cartesian map three-object triangulation	 patches and tolerant global map registration to walls	 no map no positioning
Planning exploration directed search	 6 predefined vision locations proceed to location indicated on global map	 traversal of perimeter and forays into center of arena follow walls till appropriate patch is reached, make foray into patch	 go really fast random walk
Parts & development cost	$\approx \$1,000,000$	$\approx \$500,000$	$\approx \$200$
Audience appeal	High	medium	Very High

Table 1: Carmel, Flakey and Scarecrow (adapted from Congdon *et al.* (1993))

A final interesting note about the *Roomba* family of robots is that their bumper design and motor control, when in contact with an obstacle, is identical to that of Scarecrow. Whether or not *Roomba* is a direct descendent of Scarecrow, or this is a case of parallel evolution, is uncertain.

3.2 Perception of Intelligence

Scarecrow generated some strong reactions from the participants of NCAI 1992 – both inside and outside of the robotics participant community. At the contest and for several years afterwards, the authors would receive queries asking about the spatial representation and path planning algorithms used by Scarecrow. Scarecrow was a very reactive robot. Its simple software-free architecture was able to react to stimuli much faster than architectures that put considerable computation into calculating their responses to sensor input, such as the TCA (Simmons (1992)), or even a little computation, such as Subsumption (Brooks (1986)). The time between stimulus and response in Scarecrow was constant and very short (nanoseconds). Scarecrow followed the fundamental tenant of reactive robots, that it maintains a full fidelity, mostly externally stored, model of the world[3]. Scarecrow has exactly eight bits of internal state information and two bytes of ROM.

What passes for intelligence may have more to do with not doing anything stupid than it does with solving calculus problems or planning detailed schedules. In other words, there just may not be any deep intelligence required for most tasks, and much of what appears to be intelligence is really just post-event rationalization

[3]i.e., the World itself.

for the set of actions that were taken (Dawes & Hastie (2001) & Neisser (1982)).

While this explanation may not cover everything (or even everyone), it certainly does explain the intelligence attributed by many in the audience to Scarecrow. The last row of Table 1 is subjective, but is based on the comments of several contest attendees. Even in the very technically savvy audience watching the contest, speed was a major draw, and an indication to many of the level of intelligence. Flakey was the slowest of these three robots (but by no means the slowest of the entrants). CARMEL was pretty fast and graceful. Scarecrow was very fast and teetered dangerously – an accident waiting to happen.

Scarecrow was also smartly decorated and made interesting noises. One of us (Milstein) insisted that a sound effects generator from a toy electronic gun be wired into the robot. While this at first seemed an unnecessary addition, as soon as the robot was in front of a crowd, its benefits became obvious. *What do the different noises mean? How does it know where to go? It looked like it was about to fall over – how did it recover?* These are a small sample of the multitude of questions asked of team Scarecrow by the audience.

The most commonly asked question of the other robots, but never of Scarecrow, was: *When is it going to move?*

While Scarecrow was (hopefully) the furthest from human on a cognition scale of the contest robots, it was certainly the easiest to anthropomorphize. The way it moved[4], the way it was dressed and the fact that it sounded a bit like R2D2 as it rolled by were all contributing factors to the audience attributing much more cognition to Scarecrow then was actually there. Studies have shown that it takes very little effort on the robot's part for a young child to attribute intelligence to that robot (Okita & Schwartz (2006)). Scarecrow showed that with the right level of animation, even well educated adult observers would sometimes do the same.

3.3 KIPR and Other Educational Fallout

After the competition, Jon Doyle asked one of us (Miller) how our employer felt about being affiliated with Scarecrow. Miller was on sabbatical at MIT from JPL at the time. Both organizations had expressed a mixture of admiration and horror at the project. Being 3000 miles from JPL and an independent visitor at MIT, both organizations were provided with plausible deniability – which they embraced at some level. Jon suggested that we should have made up an appropriate sponsoring organization such as *KISS*[5] *Institute*. We took his suggestion to heart, and we (Miller & Stein), along with Marc Slack, made KISS Institute a reality, with a mission to use robotics as an educational tool for students to develop skills and ambitions in technology related fields.

Since that time KISS Institute has held courses for tens of thousands of K-12 teachers and students, primarily through its Botball program (see Stein (2003)). Ironically, while Scarecrow used a heavily mechanical approach to reactive robotics, Botball, with its emphasis on C programming, is comparatively computationally intensive. It turns out that the liability insurance is much lower to teach kids how to program than it is to teach them to solder or cut aluminum. Processors used in Botball such as the XBC (LeGrand *et al.* (2005)), use an interface familiar to most kids – making software seem like something they already know,

[4]It swaggered due to the flexing of the aluminum plate to which the mast was mounted

[5]Keep It Simple, Stupid

and allow a flexibility of design not practical in mechanical or discrete component hardware. Botball embraces some traditional AI techniques such as machine vision. The heritage of KISS Institute and Botball from Scarecrow is not the technology, but the link between robotics and K-12 education. KISS Institute breaks technology into simple building blocks that young students can reassemble into amazingly complex and wonderful devices that do good things and inspire the students to do more.

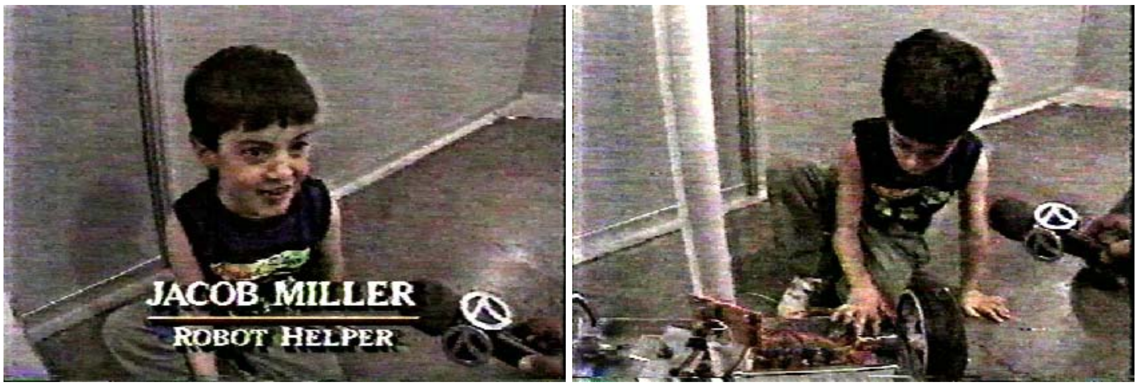

Figure 7: *My first press conference*

3.4 The Scarecrow Lesson

The authors have worked on a number of other robots that almost without exception have considerably more software and use more sensing than does Scarecrow. The technology lesson of Scarecrow is not to eschew software, vision etc, but rather to design to the requirements and goals of the task. Remember the old saying: *When all you have is a hammer, it is time to go to* Home Depot *and get some more tools*. And don't forget the equally important corollary: *Just because you have a nail-gun that doesn't mean you can't use a hammer to hang a picture*.

3.5 Final Thoughts

I (Milstein) was five years old when I gave my first press conference (Figure 7).

The fame, and the pressure at its deepest depths, came suddenly. I had always been interested in aesthetics and design; I doodled on occasion, and had messed with the *sculpey*[6] in a brief and spurious episode, but when I landed my first gig decorating Scarecrow, the pressure brought out the technique that would land me in front of the cameras. It took every bit of concentration to steady my young and anxious hands as I laid sticker after sticker upon the shiny surface of Scarecrow's hull. My father watched as I worked, with the keen compliments and sensitive silences that would drive me to an ethic of diligence that would have seemed more appropriate for an eight-year-old. Finally, I, with my parent's blessing, completed what would be known as Scarecrow's trademark look. Attaching a bowtie just below Scarecrow's sensitive head, I knew my work was finished.

[6]Sculpey is a polymer clay that can be hardened in a conventional oven, rather than a kiln.

When confronted by the press, I effortlessly offered a simple explanation of the mechanical strategy our robot employed, of how it was capable of effectively completing the AAAI course with an efficiency unmatched by the other, staggeringly expensive entries. Then the question came, and I was stunned:

> *I like the stickers on your robot. They are kind of crooked,*[7] *why did you put them like that?*

I was speechless. My five year old mind stumbled, fell, and cried out in a shriek of pain: "Looks good!"

Figure 8: *Crooked is all a matter of interpretation*

References

Bonasso, R. P., and Miller, D. P., eds. 1993. *Fall Symposium on Instantiating Intelligent Agents*. AAAI Press.

Braitenberg, V. 1986. *Vehicles: Experiments in Synthetic Psychology*. MIT Press.

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation* RA-2(1).

Clarke, A. C. 1962. *Profiles of the Future*. Warner Books, reprint edition (march 1985) edition.

Congdon, C.; Huber, M.; Kortenkamp, D.; Konolige, K.; Myers, K.; Saffiotti, A.; and Ruspin, E. 1993. Carmel versus flakey: A comparison of two winners. *AI Magazine* 14(1).

Dawes, R. M., and Hastie, R. 2001. *Rational Choice in an Uncertain World: The Psychology of Judgement and Decision Making*. Sage Publications.

Hiller, M. 1976. *Automata & Mechanical Toys: An Illustrated History*. London, England: Jupiter.

Hyman, A. 1985. *Charles Babbage: Pioneer of the Computer*. Princeton University Press.

[7] See Figure 8.

iRobot Corporation. 2006. Roomba technology. http://www.irobot.com/sp.cfm?pageid=124.

Kopplin, J. 2002. An illustrated history of computers, part 2. http://www.computersciencelab.com/ComputerHistory/HistoryPt2.htm.

LeGrand, R.; Machulis, K.; Miller, D. P.; Sargent, R.; and Wright, A. 2005. The XBC: a modern low-cost mobile robot controller. In *Proceedings of IROS 2005*. IEEE Press.

Neisser, U. 1982. John dean's memory: A case study. In *Memory Observed: Remembering in Natural Contexts*. W. H. Freeman. 139–159.

Okita, S., and Schwartz, D. L. 2006. Young children's understanding of animacy and entertainment robots. *International Journal of Humanoid Robotics* 3.

Sargent, R.; Bailey, B.; Witty, C.; and Wright, A. 1997. Dynamic object capture using fast vision tracking. *AI Magazine* 18(1).

Simmons, R. G. 1992. Concurrent planning and execution for autonomous robots. *Control Systems Magazine, IEEE* 12(1):46–50.

Stein, C. 2003. Botball: Autonomous students engineering autonomous robots. *Computers in Education Journal* 13(2).

Takanashi, S. 2002. Japanese mechanical dolls. http://www.cjn.or.jp/karakuri/takanashi.html.

The Karakuri Corner. 2006. Karakuri automata kits. http://www.karakuricorner.com/servlet/the-Karakuri-Automata-Kits/Categories.