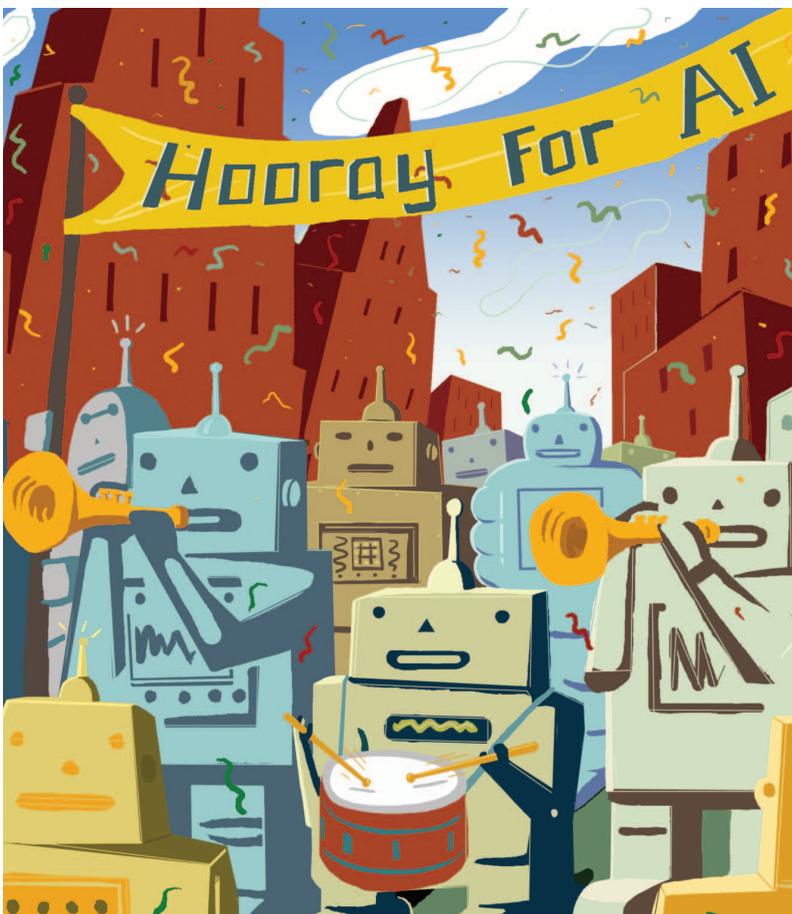


21st-Century AI: Proud, Not Smug

Tim Menzies, *West Virginia University*

"Take pride in how far you have come; have faith in how far you can go." —Anonymous

In the 21st century, AI has many reasons to be proud, but it wasn't always this way. New technologies such as AI typically follow the hype curve (see Figure 1¹). By the mid-1980s, early successes with expert systems²⁻⁵ caused skyrocketing attendance at AI conferences (see Figure 2) and a huge boom in North American AI startups. Just like



the dot-coms in the late 1990s, this AI boom was characterized by unrealistic expectations. When the boom went bust, the field fell into a trough of disillusionment that Americans call the AI Winter. A similar disillusionment had already struck earlier, elsewhere (see the “Comments on the Lighthill Report” sidebar).

If a technology has something to offer, it won't stay in the trough of disillusionment, just as AI has risen to a new sustainable level of activity. For example, Figure 2 shows that although AI conference attendance numbers have been stable since 1995, they are nowhere near the unsustainable peak of the mid-1980s.

With this special issue, I wanted to celebrate and record modern AI's achievements and activity. Hence, the call for papers asked for AI's current trends and historical successes. But the best-laid plans can go awry. It turns out that my “coming of age” special issue was about five to 10 years too late. AI is no longer a bleeding-edge technology—hyped by its proponents and mistrusted by the mainstream. In the 21st century, AI is not necessarily amazing. Rather, it's often *routine*.

A maturing technology

Evidence for AI technology's routine and dependable nature abounds. For example, in this issue (see the related sidebar for a full list), authors describe various tools to augment standard software engineering:

Comments on the Lighthill Report

In the 1970s, the Lighthill Report convinced the British government to end support for AI research in nearly all British universities. With hindsight, Lighthill's pessimism was unfounded. Even at its release, some strongly criticized the report, including John McCarthy (now professor emeritus of computer science at Stanford University):

The Lighthill Report argued that if the AI activities ... were any good they would have had more applied success by then. In the 1974 Royal Institution debate on AI, I attempted to counter by pointing out that hydrodynamic turbulence had been studied for 100 years

without full understanding. I was completely floored when Lighthill replied that it was time to give up on turbulence. Lighthill's fellow hydrodynamicists didn't give up and have made considerable advances since then. I was disappointed when BBC left that exchange out of the telecast, since it might have calibrated Lighthill's criteria for giving up on a science.¹

Reference

1. J. McCarthy, "Lessons from the Lighthill Flap," 2000, www-formal.stanford.edu/jmc/reviews/lighthill-20/lighthill-20.html.

- Yunwen Ye describes agents that assist software engineers using large libraries of components.
- Bernhard Peischl and Franz Wotawa show how to use AI diagnosis tools on software source code.
- Gary Boetticher demonstrates how well AI can learn effort estimations for software projects.

In other work, the AI field has generated many mature tools that are easily used, well documented, and well understood. For example, late last year, one of my undergraduate research assistants mentioned nonchalantly that he'd just run some data through four different data miners! That student was hardly a machine learning expert—and in the 21st century, he didn't need to be. The Waikato Environment for Knowledge Analysis toolkit (see Figure 3) contains dozens of state-of-the-art data miners, all tightly integrated around the same underlying database and object model. Weka is free, open source, well documented,⁶ compatible on many platforms, and easy to install (it took my student less than three minutes to download, install, and start running the learners). You can access it at www.cs.waikato.ac.nz/~ml/weka/index.html.

Natural language processing is another example of AI's success. In times past, NL processing was very difficult with a low chance of success. These days, researchers can rely on numerous tools to build successful NL applications. For example, NL processing often requires extensive background knowledge of the words being processed. Many general ontologies are now freely available. These public-domain ontologies range from WordNet (a lexical database for English) to OpenCyc (a formalization of many commonsense concepts). More spe-

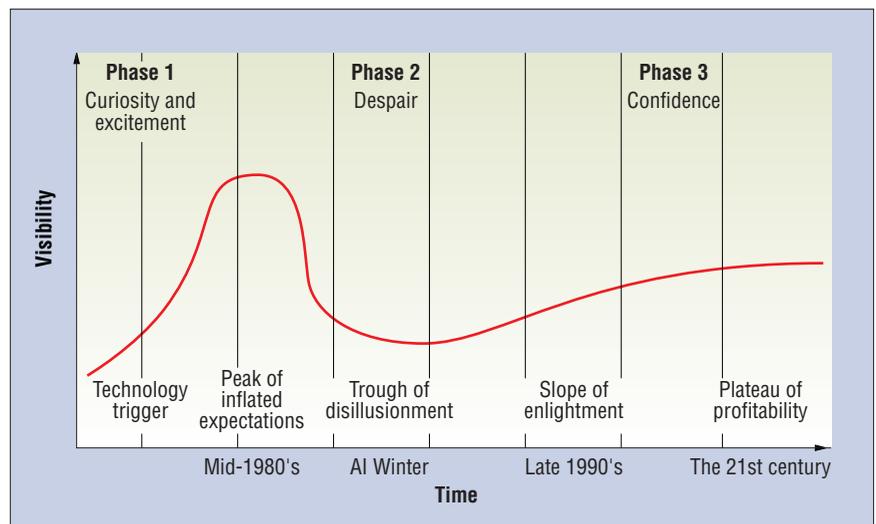


Figure 1. The hype cycle for new technology.

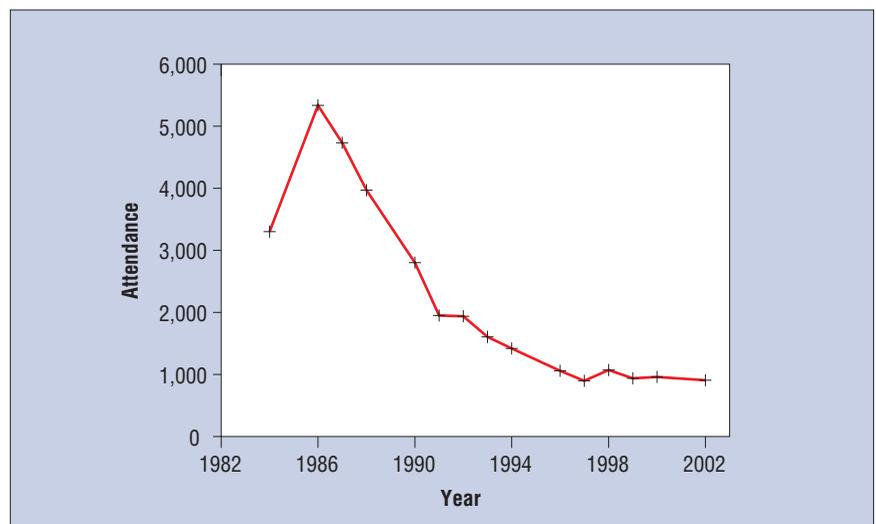


Figure 2. Attendance at the National Conference on Artificial Intelligence (AAAI), the senior American AI conference. Figures prior to 1984 are not available. No figures are shown for 1985, 1989, and 2001 because these were IJCAI (International Joint Conference on Artificial Intelligence) years.

In this Issue

What's AI Done for Me Lately? Genetic Programming's Human-Competitive Results

by John R. Koza, Martin A. Keane, and Matthew J. Streeter pp. 25–31. As computer time becomes cheaper, genetic programming will be routinely used as an invention machine to produce useful new designs, generate patentable new inventions, and engineer around existing patents.

Model-Based Diagnosis or Reasoning From First Principles

by Bernhard Peischl and Franz Wotawa, pp. 32–37. Modern model-based reasoning technology is fast enough to be applied to small to medium-sized programs.

Visual Object Recognition with Supervised Learning

by Bernd Heisele, pp. 38–42. Vision systems that learn and adapt represent one of the most important trends in computer vision research and might provide the only solution to the development of robust and reusable vision systems.

Programming with an Intelligent Agent

by Yunwen Ye, pp. 43–47. Programmers can miss the components they need in large component libraries. CodeBroker is an intelligent software agent that can automatically find the components that a programmer misses.

When Will It Be Done? Machine Learner Answers to the 300-Billion-Dollar Question

by Gary D. Boetticher, pp. 48–50. The international \$300-billion software development industry needs better predictors for software development costs. Data miners can learn such predictors to an impressive level of accuracy.

From Open IS Semantics to the Semantic Web: The Road Ahead

by David G. Schwartz, pp. 52–58. Millions of people around the world are writing Web pages. The semantic content of those pages is usually inaccessible. The Semantic Web is a global initiative to dramatically improve how we structure and share content on the Web.

cific ontologies are also freely available, such as the Unified Medical Language System (see Figure 4).

Overall, the ontologies are extensive. For

example, WordNet covers 111,223 English words, and UMLS's January 2003AA edition includes 875,255 concepts and 2.14 million concept names in over 100 biomedical

source vocabularies, some in multiple languages. Building such ontologies is a huge task, and David Schwartz (in this issue) discusses a global initiative to build semantic dictionaries via the World Wide Web.

Apart from ontologies, executable NL tools are also readily available. For example, Debbie Richards recently led a small university team that implemented a system to detect contradictions between different NL sentences in an object-oriented design.⁷ In the 1980s, such software would have only been found in science fiction. But in the 21st century, the Richards team had very little NL processing to implement. They just added (for example) negotiation tools to standard NL components. Those components included an answer extraction system from NL, a formal concept analysis component that generates a visualization of the text, and Prolog and Java tools that implemented the other components. This menagerie of tools seems complex. However, AI components are now mature enough to enable simple combinations.

Challenges

AI still can't be smug, despite the successes listed in the "AI Applications" sidebar. Although some AI areas are mature, there's still much to learn and some traps to avoid.

In his invited talk at AAAI 1999, Nils Nilsson argued that the easy days of AI are over:

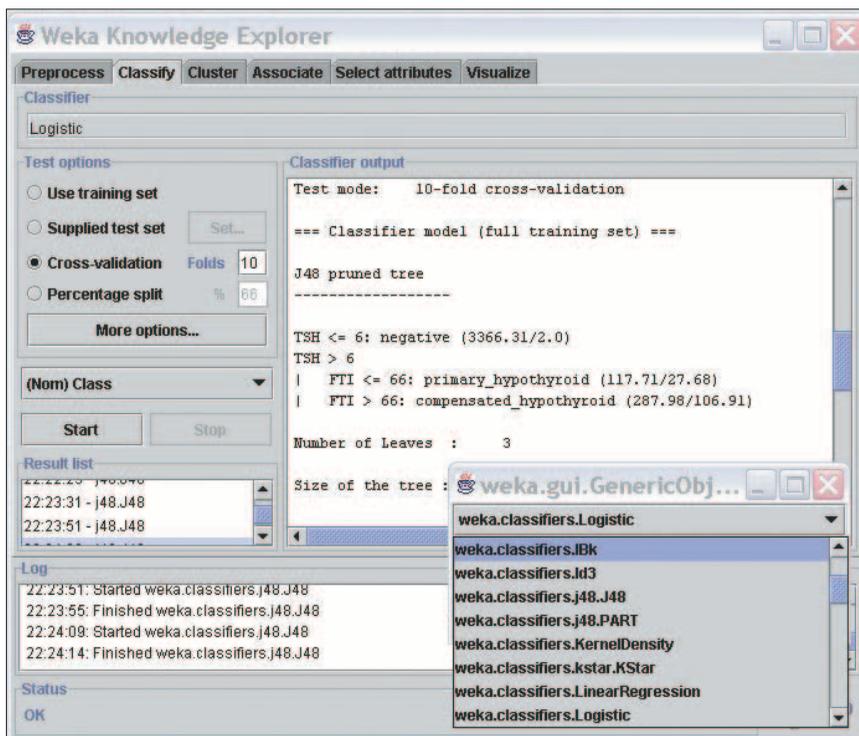


Figure 3. The Waikato Environment for Knowledge Analysis tool.

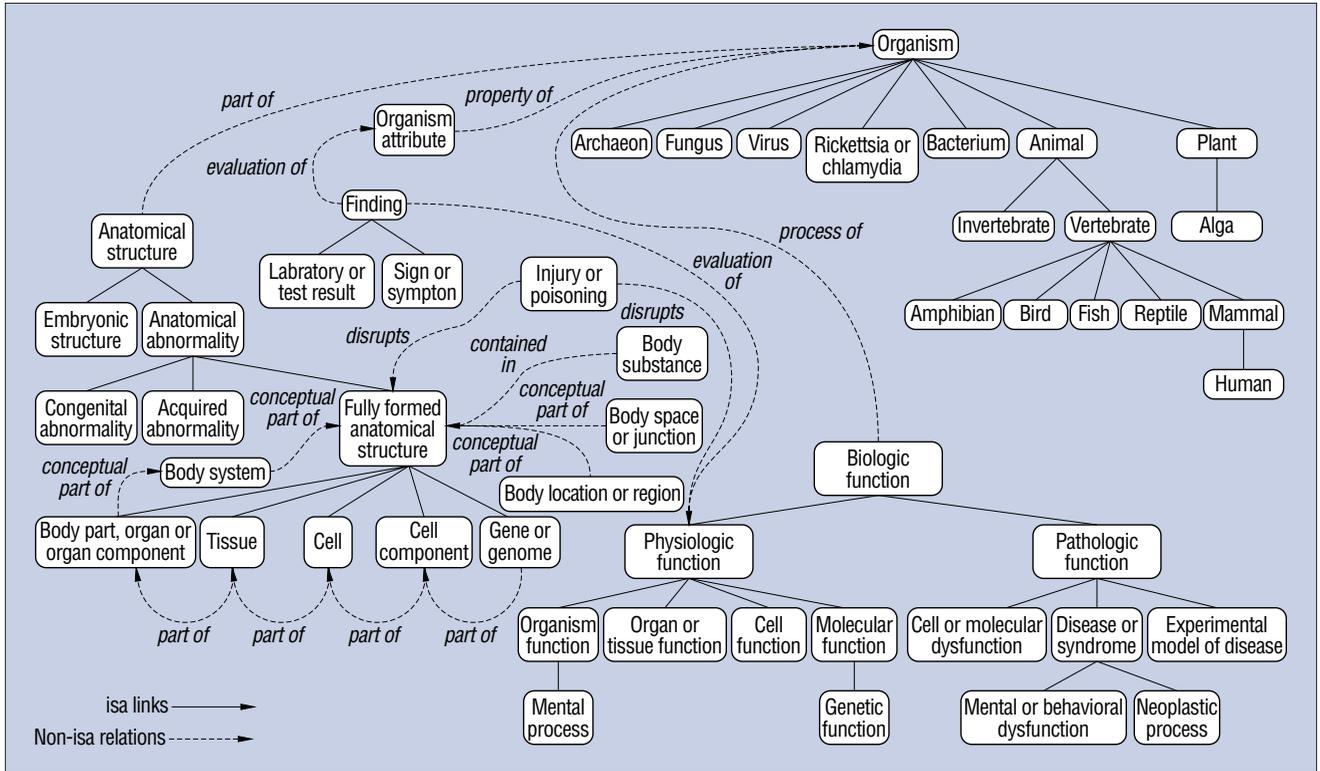


Figure 4. Part of the Unified Medical Language System semantic network (www.nlm.nih.gov/research/umls/META3.HTML). Each child in the hierarchy is linked to its parent by the *isa* link.

Useful URLs

Dynamic Analysis and Replanning Tool: www.bbn.com/lpds/dart.html

Navlab 5 USA tour: http://www-2.cs.cmu.edu/afs/cs/user/tjochem/www/nhaa/nhaa_home_page.html

OpenCyc: www.opencyc.org

The Remote Agent: <http://ic.arc.nasa.gov/projects/remote-agent/tlm2html/raxang.html>

The Unified Medical Language System: www.nlm.nih.gov/research/umls

Weka data mining toolkit: www.cs.waikato.ac.nz/ml/weka

WordNet: www.cogsci.princeton.edu/wn/wn1.7.1.shtml

The easy work (such as inventing A* and the idea of STRIPS operators), is over. AI is getting harder. In addition, AI researchers will have to know a lot about many related disciplines.

Nilsson offered Figure 5 as a partial list of related disciplines. He warned against a *fission effect*, which could tear apart the field. Paradoxically, this effect results from AI's success:

Fission is promoted by the tendency of AI to be pulled apart by the many adjacent disciplines that join with AI to field practical, large-scale applications in specialized niches.

In fact, some computer scientists and others might go so far as to say, "Why do we need AI as a separate field? One could carve it up, add the parts to adjacent fields and get along perfectly well without it."

Nilsson then proposed several large challenge problems to maintain a coherent field of study in AI. For the record, his hot list of near-term research includes case-based reasoning (again) for planning; using logic (again) for planning; SAT encodings of planning problems; large, reusable knowledge bases; agent (robot and softbot) architectures; agent-to-

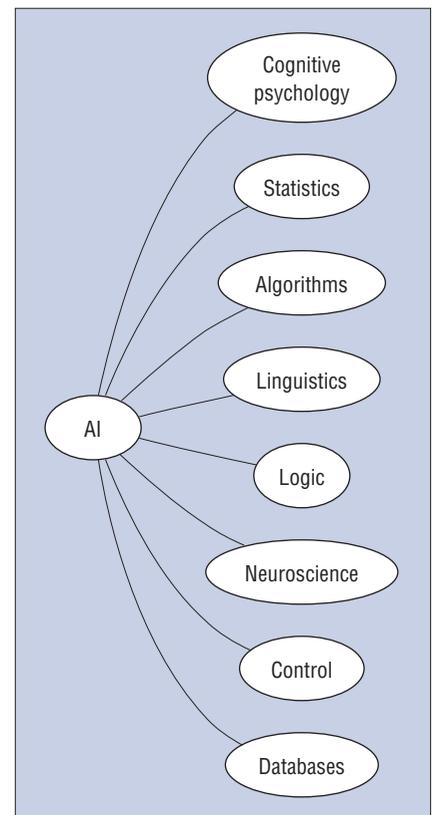


Figure 5. Near neighbors to AI.

AI Applications

AI has made much progress in specific application areas—for example:

Genetic programming

John Koza, Martin Keane, and Matthew Streeter (in this issue) discuss how genetic programming can duplicate human invention. They find they can reengineer new solutions to solve the same problems addressed by state-of-the-art patents.

Image recognition

Bernd Heisele (in this issue) describes the state-of-the-art in recognizing images from video. Soon, AI police agents will be able to monitor large crowds.

Expert systems

Since the 1970s, medical expert systems have been achieving human levels of medical expertise.¹ Such systems are now used daily and trusted around the world. For example, automatic tools for assessing electrocardiograms are now so good that it's routine for humans to pay for their services.²

During the 1980s and 1990s, DEC used the XCON (expert configurer) expert system to automatically configure computer hardware components.³⁻⁵ This system saved DEC millions of dollars a year and freed up designers to work on next-generation DEC computers.

Scalability

Since the 1990s, numerous researchers have reported that previous pessimism regarding AI's scalability might be unfounded. Stochastic inference procedures enable the processing of declarative theories that are orders of magnitude larger than anything previously processable.⁶⁻⁸ These results remove one of the fundamental objections to AI made in the Lighthill Report (see the related sidebar).

Logistic planning

In 1991, Iraq invaded Kuwait, so America invaded Iraq in



Figure A. Close-up of the Remote Agent, ion engine firing. (courtesy of NASA Ames/JPL)

Operation Desert Storm. DARPA claimed they saved more money using AI logistics planners during Desert Storm than was ever invested over decades into AI research.⁹

Vision systems

In 1995, an automatic vision system steered a vehicle across America. The ALVINN system from Carnegie Mellon University autonomously drove a van, Navlab5, from Pittsburgh to San Diego while human operators worked the brake and accelerator. The system controlled the steering for all but 52 miles of 2,849-mile journey, averaging 63 mph day and night in all weather.

Evaluators

In 1997, Deep Blue—an IBM supercomputer—defeated world chess champion Garry Kasparov in six games. It was the first time a computer had won a chess match against a current world champion under tournament conditions. Kasparov won the first game, lost the second, and played to a draw in the

agent communication languages; more expressive Bayes nets; Bayes net learning; and genetic algorithm and programming techniques.

Although Nilsson's comments are timely, I'm more confident than he about AI's future as a coherent discipline. The long-term goal of emulating general human intelligence remains, and that goal will bind future generations of AI researchers. The successes listed in the "AI Applications" sidebar show that you can achieve much without human-level sophistication. Nevertheless, I still dream of the day when my word processor writes articles like this one while I go to the beach.

The road ahead

The goal of creating general human-level intelligence has inspired, and still inspires, decades of talented graduate students who flock to the hardest problem they know. These students strive to distance themselves from those working on other well-defined, mostly solved problems. Hence, these students are always proud to boast that they are working on AI.

It's bad manners to form an army if you can't feed them. But our AI graduate students won't starve. As they work toward the long-term goal of human-level intelligence, they'll still be able to pay the rent using AI, for

example, by working in the emerging gaming industry. This industry is already huge (approximately 17 billion dollars revenue in 2002⁸) and is still growing. Our AI workers will stay busy building the next generation of gaming softbots.⁹ As the World Wide Web grows, these softbots will have access to "eyes" that can see more information than any human intelligence could see in a lifetime. As we continue to use software to control our world, these softbots will be given increasingly sophisticated "arms." With these eyes and arms, systems have a rising opportunity to learn and influence the world.

next three. In game six, "Kasparov got wiped off the board," according to grand master Ilya Gurevich (www.cnn.com/SPECIALS/1997/year.ender/scitech/06.deepblue.html).

Autonomous agents

In 1999, a NASA AI agent (see Figure A) ran a satellite beyond Mars for over a day, without ground control. The agent continually reviewed and updated the mission goals according to the satellite's functioning hardware.¹⁰ Earthlings could subscribe to a mailing list and get frequent bulletins from the satellite about what it was currently thinking about (see Figure B).

AI spin-offs

The list of spin-offs from AI labs is impressive and includes the mouse; time-sharing operator systems; high-level symbolic programming languages (Lisp, Prolog, Scheme); computer graphics; the graphical user interface; computer games; the laser printer; object-oriented programming; the personal computer; email; hypertext; software agents crawling the Web; and symbolic mathematics systems such as Macsyma, Mathematica, Maple, and Derive.

References

1. V. Yu et al., "Antimicrobial Selection by a Computer: A Blinded Evaluation by Infectious Disease Experts," *J. American Medical Assoc.*, vol. 242, no. 12, 21 Sept. 1979, pp. 1279–1282.
2. J. Willems et al., "The Diagnostic Performance of Computer Programs for the Interpretation of Electrocardiograms," *The New England J. Medicine*, vol. 325, no. 25, 19 Dec. 1991, pp. 1767–1773; abstract: <http://content.nejm.org/cgi/content/abstract/325/25/1767>.
3. J. McDermott, "R1's Formative Years," *AI Magazine*, vol. 2, no. 2, Summer 1981, pp. 21–29.
4. J. Bachant and J. McDermott, "R1 Revisited: Four Years in the Trenches," *AI Magazine*, vol. 5, no. 3, Fall 1984, pp. 21–32.
5. J. McDermott, "R1 ("XCON") at Age 12: Lessons from an Elementary School Achiever," *Artificial Intelligence*, vol. 59, nos. 1–2, Feb. 1993, pp. 241–247.

For another AI meal ticket, consider the growing field of model-based software engineering. Safety concerns are forcing the aviation industry to use MBSE. More planes are flying each day, but the odds of a software error are constant. Unless we can reduce the rate of software errors, by 2030 there will be a major air traffic accident reported daily. MBSE tools allow for early life-cycle software simulation, verification, and validation. Furthermore, they remove the need for laborious, possibly error-prone, manual code generation. So, the aviation industry is rapidly maturing MBSE. Soon, the broader software-

engineering community will be able to access and use MBSE tools. When that happens, accurate declarative descriptions of all software will exist. Bring on the AI! For example,

- Use case-based reasoning to find model-based components that are relevant to the current development
- Apply search methods or constraint satisfaction tools to optimize verification
- Work within knowledge acquisition and maintenance environments to enable faster model collection and modification

Date: Fri, 21 May 1999 20:53:51 GMT
From: Remote Agent
Subject: Notice from DS1 Remote Agent telemetry

1. At 21 May 1999 20:46:25 GMT, spacecraft time frame:

- ** Spontaneous spacecraft state changes just detected.
- ** Steering with thrusters is degraded but usable.

Ability to use X-facing thrusters for Y-axis torque has failed and appears unrecoverable.

2. At 21 May 1999 20:46:25 GMT, spacecraft time frame: RAX is looking for a way to recover and maintain the important states:

- ** Make it so MICAS camera is powered on.
- ** Make it so steering with thrusters is working okay.

Figure B. Text of an email converted from telemetry sent by the DS1 spacecraft's Remote Agent, an autonomy experiment developed at NASA Ames and JPL.

6. P. Cheeseman, B. Kanefsky, and W. Taylor, "Where the Really Hard Problems Are," *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI 91)*, AAAI Press, 1991, pp. 331–337.
7. H. Kautz and B. Selman, "Pushing the Envelope: Planning, Propositional Logic and Stochastic Search," *Proc. 13th Nat'l Conf. Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conf.*, AAAI Press/MIT Press, 1996, pp. 1194–1201; www.cc.gatech.edu/~jimmyd/summaries/kautz1996.ps.
8. D. Owen and T. Menzies, "Lurch: A Lightweight Alternative to Model Checking, to be published in *Proc. 15th Int'l Conf. Software Eng. and Knowledge Eng. (SEKE 03)*, World Scientific, 2003; <http://tim.menzies.com/pdf/03lurch.pdf>.
9. Maxwell Air Force Base Introduction to Artificial Intelligence, www.au.af.mil/au/aul/school/acsc/ai02.htm.
10. N. Muscettola et al., "Remote Agent: To Boldly Go Where No AI System Has Gone Before," *Artificial Intelligence*, vol. 103, nos. 1–2, Aug. 1998, pp. 5–48.

Will MBSE or softbot manufacturers use the term AI? Of course! A rose by any other name is still implemented using AI. Softbots will use search methods and data mining techniques. MBSE will still need to understand its knowledge representations' logic. These systems will integrate via the high-level languages we developed using ontologies we built and debugged. As we struggle to implement, understand, and optimize MBSE-built agents running around the World Wide Semantic Web, researchers will still rush to read the latest issues of *Artificial Intelligence Journal*, the proceedings from

Table 1. Remarkable events in the 20th and 21st centuries.

20th century (actual)	21st century (estimated)
Female suffrage	Cure for cancer
Practical, effective, and widespread contraception	Surviving the population explosion
Explosions: both atomic and population	Taming atomic terrorism
Antibiotics and the end to smallpox	Applying newfound knowledge of the human genome
The American civil rights movement	Human, animal, or cybernetic grafting
The push to democracy in Eastern Europe that toppled the Berlin wall	Replacing economic growth with economic sustainability
Computers everywhere	Practical fusion power
The international Olympic games	Creating self-sustaining ecologies in outer space
Mass communication: telephone, television, and World Wide Web	Asteroid mining
Humans flying higher and faster than any bird	Space colonies and trips to other stars
Walking on the moon	Contact with alien races

AAAI and IJCAI, and (of course) *IEEE Intelligent Systems*.

Modern AI workers can be very proud. Much has been accomplished. We have survived the birth trauma of this new technology. We have developed tools that enabled numerous landmark applications. We have matured those tools into dependable and reusable components. And we still inspire the smartest minds to work on the hardest problems.

As proud as we are, we mustn't be smug. Consider the list of landmark events shown in Table 1. Compared to any of those, is AI remarkable enough to be memorable in, say, 200 years time? I think not—but that can change. AI's mark in history could be prominent and permanent if the 21st century becomes the birthday of this planet's second intelligent race. Your own work—past, present, and future—will decide. ■

Acknowledgments

Nils Nilsson, Enrico Coiera, and numerous contributors to the comp.ai newsgroup kindly shared their lists of landmark AI applications. Also, Nigel Shadbolt and Chris Welty offered useful and timely advice during this issue's planning.

I conducted this research at West Virginia University under NASA contract NCC2-0979 and NCC5-685. The NASA Office of Safety and Mission Assurance under the Software Assurance Research Program led by the NASA Independent Verification and Validation Facility sponsored this work.

References

1. P. Wilson-Smith, "Fund Managers Take Fright over Myners," *Financial News*, 26 Mar. 2001.
2. J. Bachant and J. McDermott, "R1 Revisited: Four Years in the Trenches," *AI Magazine*, vol. 5, no. 3, Fall 1984, pp. 21–32.
3. D. Barstow et al., "Languages and Tools for Knowledge Engineering," *Building Expert Systems*, F. Hayes-Roth, D. Waterman, and D. Lenat, eds., Addison-Wesley, 1983, pp. 283–345.
4. B. Buchanan and E. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, 1984.
5. A. Campbell et al., "Recognition of a Hidden Material Deposit by an Artificially Intelligent Program," *Science*, vol. 217, 3 Sept. 1982, pp. 927–929.
6. I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999.
7. D. Richards, "Merging Individual Conceptual Models of Requirements," to be published in *Requirements Engineering*, 2003.
8. "The 6 Myths of Computer Research," *Game Research*, 2003, www.game-research.com/art_myths_of_gaming.asp.
9. J. Laird and M. van Lent, "Interactive Computer Games: Human-Level AI's Killer Application," *Proc. 17th Nat'l Conf. Artificial Intelligence (AAAI 02)*, AAAI Press, 2000; <http://ai.eecs.umich.edu/people/laird/papers/AAAI-00.pdf>.

For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/dlib>.

The Author



Tim Menzies is the software engineering research chair at NASA's Independent Verification and Validation Facility. His research interests include data mining, software engineering, knowledge engineering, and verification & validation. He received his PhD in artificial intelligence from the University of New South Wales, Sydney, Australia. He is a member of the IEEE and ACM. Contact him at tim@menzies.us; <http://menzies.us>.