Intelligent IEEE Systems
www.computer.org/intelligent

Human Responsibility for Autonomous Agents

Ben Shneiderman

Vol. 22, No. 2
March/April 2007

Human Responsibility for Autonomous Agents

Ben Shneiderman, *University of Maryland, College Park*

Automobile airbag triggers and heart pacemakers require rapid automated reaction. Similarly, increasing numbers of computerized systems need only intermittent human control, such as planetary explorers, Web-based crawlers and spiders, stock market traders, and manufacturing assembly lines. Developing life-critical applications in transportation, medical, and battlefield systems will inevitably increase the stakes for system designers.

In all these cases, the rising levels of automation bring benefits but also can increase dangers. Automated or autonomous systems can sometimes fail harmlessly, but they can also destroy data, compromise privacy, and consume resources, such as bandwidth or server capacity. What's more troubling is that automated systems embedded in vital systems can cause financial losses, destruction of property, and loss of life.

Controlling these dangers will increase trust while enabling broader use of these systems with higher degrees of safety. Obvious threats stem from design errors and software bugs, but we can't overlook mistaken assumptions by designers, unanticipated actions by humans, and interference from other computerized systems.

The danger of mistaken assumptions

Automobile airbags provide a dramatic case study.[1] In their early years, airbags were estimated to save 1,500 to 2,500 lives per year in the US alone, but they might have inadvertently killed 25 to 50 children annually by deploying in low-speed crashes. Once these facts became clear, developers implemented improvements and extended user control by letting drivers reset deployment parameters if children were in the passenger seat.

Similarly with unmanned aerial vehicles, early assumptions were that autonomy would be high, but in reality, many aspects of successful operation require operator monitoring and control.[2,3] As designers identify failure patterns, accommodate critical decision points requiring human control, and limit conflicts among multiple UAVs, increasing levels of autonomy will be possible.

This column aims to promote greater awareness of human responsibility for computerized systems, especially those called autonomous. By using the term autonomous, some designers might assume high reliability and therefore reduce feedback to operators, inhibit human control, and fail to record performance for human review.

It seems important to remind all computerized-system designers, software implementers, and operators that they're legally liable and financially accountable for their systems. Contracts and license agreements might limit liability, but it's always wise to incorporate cautious planning, careful operation, and frequent reviews to reduce dangerous outcomes.

Thoughtful designers recognize human responsibility for system design, implementation, and operation, leading them to build advanced user interfaces that let operators effectively monitor and control autonomous systems.[4–6] These user interfaces will also provide logging tools that enable operators to understand system behavior across many operations and therefore improve it.

Monitoring, control, and logging

Monitoring tools inform users of the autonomous system's current state and activities. Some state information, such as battery power or current location, is easy to provide, and some activities, such as most recent actions, are simple to comprehend. Other state and activity reports can be much more complex, requiring the display of advanced information visualizations.[3] Network monitoring is a successful application that has seen widespread use of information visualization, especially to detect intrusions and improve performance.[7] New applications or system versions often require monitoring, but as trust increases, monitoring can decrease.

Control user interfaces have a long history, but the complexity of autonomous systems generates new opportunities for design innovation. Operator interventions to change goals or recover from problems begin with situation awareness and a rich model of the implications of any intervention.[8] Interventions could range from simple shutdown commands, to intricate schedule revisions, to high-

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

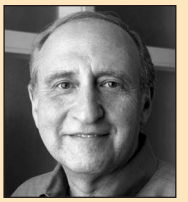
level goal changes that might require substantial alterations to plans generated by the autonomous systems. There might be several interactions during which the operator learns about the system state, completed activities, and implications of goal changes. Chemical plant, air traffic, or power systems control systems are mature, successful applications that have shown increased levels of automation while preserving human control. Unmanned aircraft, robotic undersea manipulators, and planetary explorers present evolving challenges because some operations require high levels of autonomy, but human control for goal setting and problem solving is also necessary.

Logging of autonomous systems lets operators critique an individual operation, much like a postgame review in sports, and retrospectively compare multiple operations. User interfaces should let system maintainers review individual operations to study performance, ensure proper completions, and detect anomalies. In particular, they can use logs to track down the cause of specific malfunctions or failed missions. User interfaces should also let developers of next-generation systems analyze logs for hundreds of sessions to spot opportunities for improvement. Flight data recorders are a good example of well-developed system monitoring that has high payoffs in increased safety and improved performance.

Preserving human control while increasing the level of automation is usually desirable and sometimes required.[9] Designers of autonomous systems who recognize human responsibility will include monitoring, control, and logging—features that are likely to lead to more reliable systems. ■

References

1. M.C. Meyer and T. Finney, "Who Wants Airbags?" *Chance*, Spring 2005, pp. 3–16; www.amstat.org/publications/chance/182.feature.pdf.
2. H.A. Ruff et al., "Exploring Automation Issues in Supervisory Control of Multiple UAVs," *Proc. Human Performance, Situation Awareness, and Automation Technology Conf.*, OmniPress, 2004, pp. 218–222.
3. H.A. Ruff, S. Narayanan, and M.H. Draper, "Human Interaction with Levels of Automation and Decision-Aid Fidelity in the Supervisory Control of Multiple Simulated Unmanned Air Vehicles," *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 4, 2002, pp. 335–351.
4. Q.H. Mahmoud and L. Yu, "Making Software Agents User-Friendly," *Computer*, vol. 39, no. 7, 2006, pp. 96, 94–95.
5. R. Parasuraman, T.B. Sheridan, and C.D. Wickens, "A Model for Types and Levels of Human Interaction with Automation," *IEEE Trans. Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 30, no. 3, 2000, pp. 286–297.
6. T.B. Sheridan, *Humans and Automation: System Design and Research Issues*, John Wiley & Sons, 2002.
7. J. Oberheide, M. Karir, and D. Blazakis, "VAST: Visualizing Autonomous System Topology," *VizSEC 06: Internet Proc. 3rd Int'l Workshop Visualization for Computer Security*, ACM Press, 2006; www.projects.ncassr.org/sift/vizsec/vizsec06/program/vizsec12.pdf.
8. M.R. Endsley and D.B. Kaber, "Level of Automation Effects on Performance, Situation Awareness and Workload in a Dynamic Control Task," *Ergonomics*, vol. 42, no. 3, 1999, pp. 462–492.
9. B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 4th ed., Addison-Wesley, 2005.



Ben Shneiderman is a professor in the Department of Computer Science, the founding director of the Human-Computer Interaction Laboratory, and a member of the Institute for Advanced Computer Studies at the University of Maryland, College Park. He's the author of *Leonardo's Laptop: Human Needs and the New Computing Technologies* (MIT Press, 2003), which won the 2004 IEEE Distinguished Literary Contribution Award. Contact him at ben@cs.umd.edu.