Machine Learning for Design, Week 7

Automating Design

Four chairs from a design agent



Round 1



Round 2



Round 3



Round 4

Brief: a new side chair for a museum café that draws on the 56 historical chairs. Gemini 2.5 Flash plans and checks, FLUX.1 [schnell] makes each image; run 1 of 4.

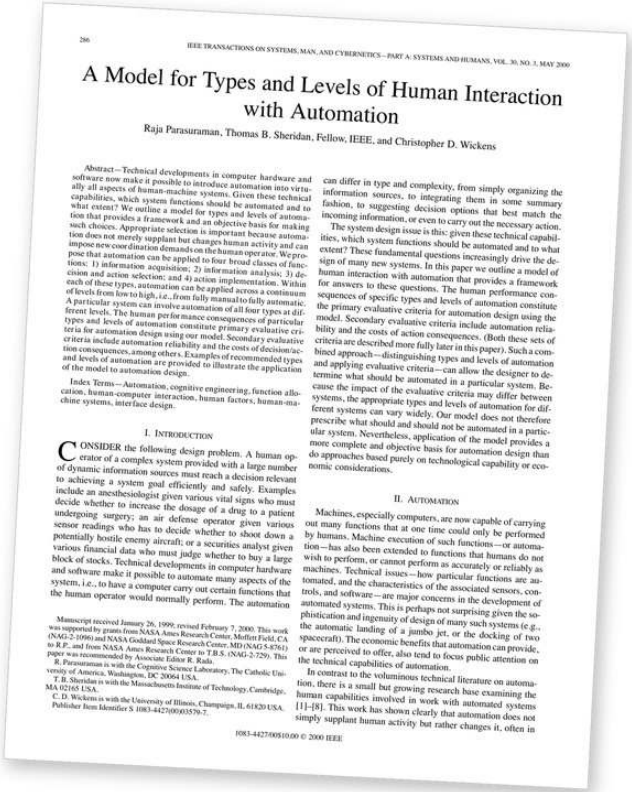
What should an agent decide on its own, and what should a designer decide?

Today

1. Levels of automation
2. Four runs of our agent
3. Ironies of automation
4. Discussion
5. Lab: read an agent's log
6. Working prototypes: test another team's tool
7. Crit

1. Levels of automation

Today's reading: Parasuraman, Sheridan & Wickens, “A model for types and levels of human interaction with automation,” IEEE SMC-A 2000



286 IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 30, NO. 3, MAY 2000

A Model for Types and Levels of Human Interaction with Automation

Raja Parasuraman, Thomas B. Sheridan, Fellow, IEEE, and Christopher D. Wickens

Abstract—Technical developments in computer hardware and software now make it possible to introduce automation into virtually all aspects of human-machine systems. Given these technical capabilities, which system functions should be automated and to what extent? We outline a model for types and levels of automation that provides a framework and an objective basis for making such choices. Appropriate selection is important because automation does not merely supplant but changes human activity and can impose new coordination demands on the human operator. We propose that automation can be applied to four broad classes of functions: 1) information acquisition; 2) information analysis; 3) decision and action selection; and 4) action implementation. Within each of these types, automation can be applied across a continuum of levels from low to high, i.e., from fully manual to fully automatic. A particular system can involve automation of all four types at different levels. The human performance consequences of particular types and levels of automation constitute primary evaluative criteria for automation design using our model. Secondary evaluative criteria include automation reliability and the costs of decision/action consequences, among others. Examples of recommended types and levels of automation are provided to illustrate the application of the model to automation design.

Index Terms—Automation, cognitive engineering, function allocation, human-computer interaction, human factors, human-machine systems, interface design.

I. Introduction

CONSIDER the following design problem. A human operator of a complex system provided with a large number of dynamic information sources must reach a decision relevant to achieving a system goal efficiently and safely. Examples include an anesthesiologist given various vital signs who must decide whether to increase the dosage of a drug to a patient undergoing surgery; an air defense operator given various sensor readings who has to decide whether to shoot down a potentially hostile enemy aircraft; or a securities analyst given various financial data who must judge whether to buy a large block of stocks. Technical developments in computer hardware and software make it possible to automate many aspects of the system, i.e., to have a computer carry out certain functions that the human operator would normally perform. The automation can differ in type and complexity, from simply organizing the information sources, to integrating them in some summary fashion, to suggesting decision options that best match the incoming information, or even to carry out the necessary action.

The system design issue is this: given these technical capabilities, which system functions should be automated and to what extent? These fundamental questions increasingly drive the design of many new systems. In this paper we outline a model of human interaction with automation that provides a framework for answers to these questions. The human performance consequences of specific types and levels of automation constitute the primary evaluative criteria for automation design using the model. Secondary evaluative criteria include automation reliability and the costs of action consequences. (Both these sets of criteria are described more fully later in this paper). Such a combined approach—distinguishing types and levels of automation and applying evaluative criteria—can allow the designer to determine what should be automated in a particular system. Because the impact of the evaluative criteria may differ between systems, the appropriate types and levels of automation for different systems can vary widely. Our model does not therefore prescribe what should and should not be automated in a particular system. Nevertheless, application of the model provides a more complete and objective basis for automation design than do approaches based purely on technological capability or economic considerations.

II. Automation

Machines, especially computers, are now capable of carrying out many functions that at one time could only be performed by humans. Machine execution of such functions—or automation—has also been extended to functions that humans do not wish to perform, or cannot perform as accurately or reliably as machines. Technical issues—how particular functions are automated, and the characteristics of the associated sensors, controls, and software—are major concerns in the development of automated systems. This is perhaps not surprising given the sophistication and ingenuity of design of many such systems (e.g., the automatic landing of a jumbo jet, or the docking of two spacecraft). The economic benefits that automation can provide, or are perceived to offer, also tend to focus public attention on the technical capabilities of automation.

In contrast to the voluminous technical literature on automation, there is a small but growing research base examining the human capabilities involved in work with automated systems [1]–[8]. This work has shown clearly that automation does not simply supplant human activity but rather changes it, often in

Manuscript received January 26, 1999; revised February 7, 2000. This work was supported by grants from NASA Ames Research Center, Moffett Field, CA (NAG-2-1096) and NASA Goddard Space Research Center, MD (NAG 5-8761) to R.P., and from NASA Ames Research Center to T.B.S. (NAG-2-729). This paper was recommended by Associate Editor R. Rada.

R. Parasuraman is with the Cognitive Science Laboratory, The Catholic University of America, Washington, DC 20064 USA.

T. B. Sheridan is with the Massachusetts Institute of Technology, Cambridge, MA 02165 USA.

C. D. Wickens is with the University of Illinois, Champaign, IL 61820 USA.

Publisher Item Identifier S 1083-4427(00)03570-7.

1083-4427/00$10.00 © 2000 IEEE

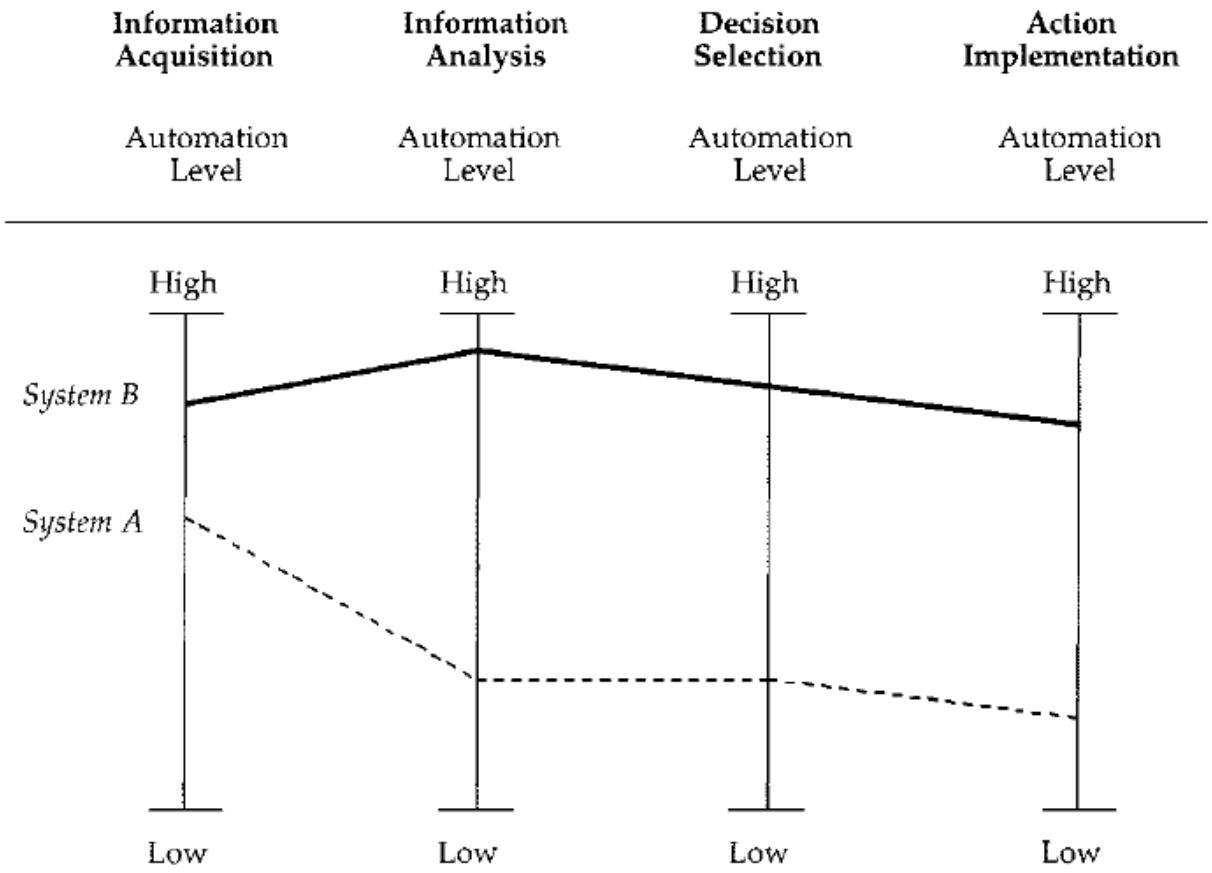
Levels of automation

TABLE I
LEVELS OF AUTOMATION OF DECISION AND ACTION SELECTION

HIGH	10. The computer decides everything, acts autonomously, ignoring the human.
	9. informs the human only if it, the computer, decides to
	8. informs the human only if asked, or
	7. executes automatically, then necessarily informs the human, and
	6. allows the human a restricted time to veto before automatic execution, or
	5. executes that suggestion if the human approves, or
	4. suggests one alternative
	3. narrows the selection down to a few, or
	2. The computer offers a complete set of decision/action alternatives, or
LOW	1. The computer offers no assistance: human must take all decisions and actions.

Parasuraman et al., IEEE SMC-A 2000, Table I

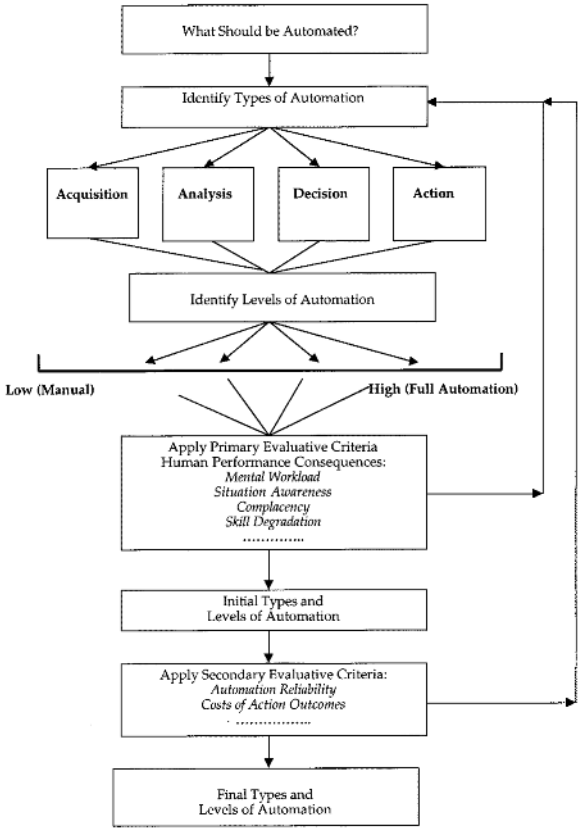
Four functions of automation



A system can automate each function at its own level.

Parasuraman et al., IEEE SMC-A 2000, Fig. 2

Choosing a level



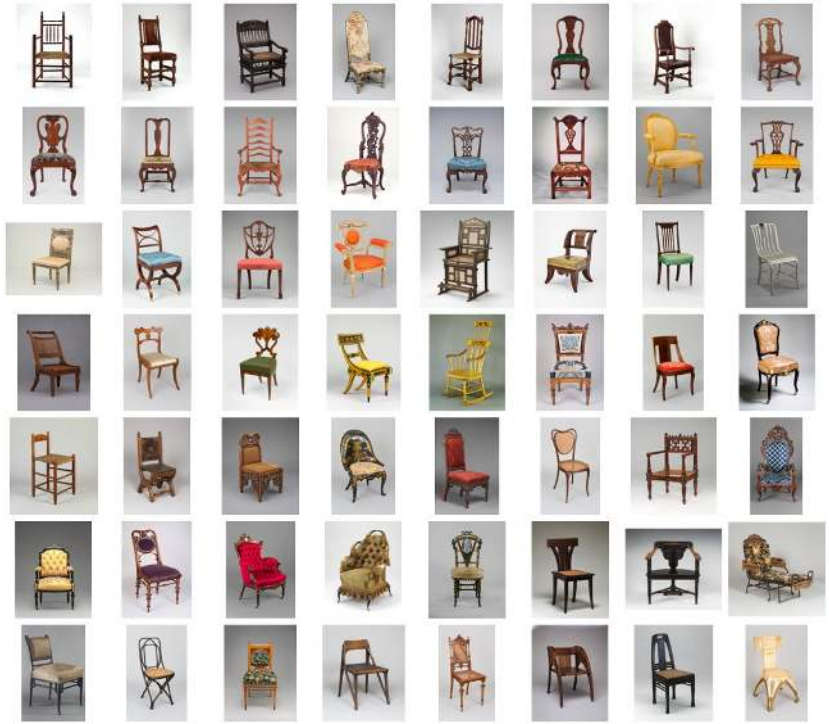
First: what each level does to the human operator

Then: how reliable the automation is, and the cost of a wrong decision or action

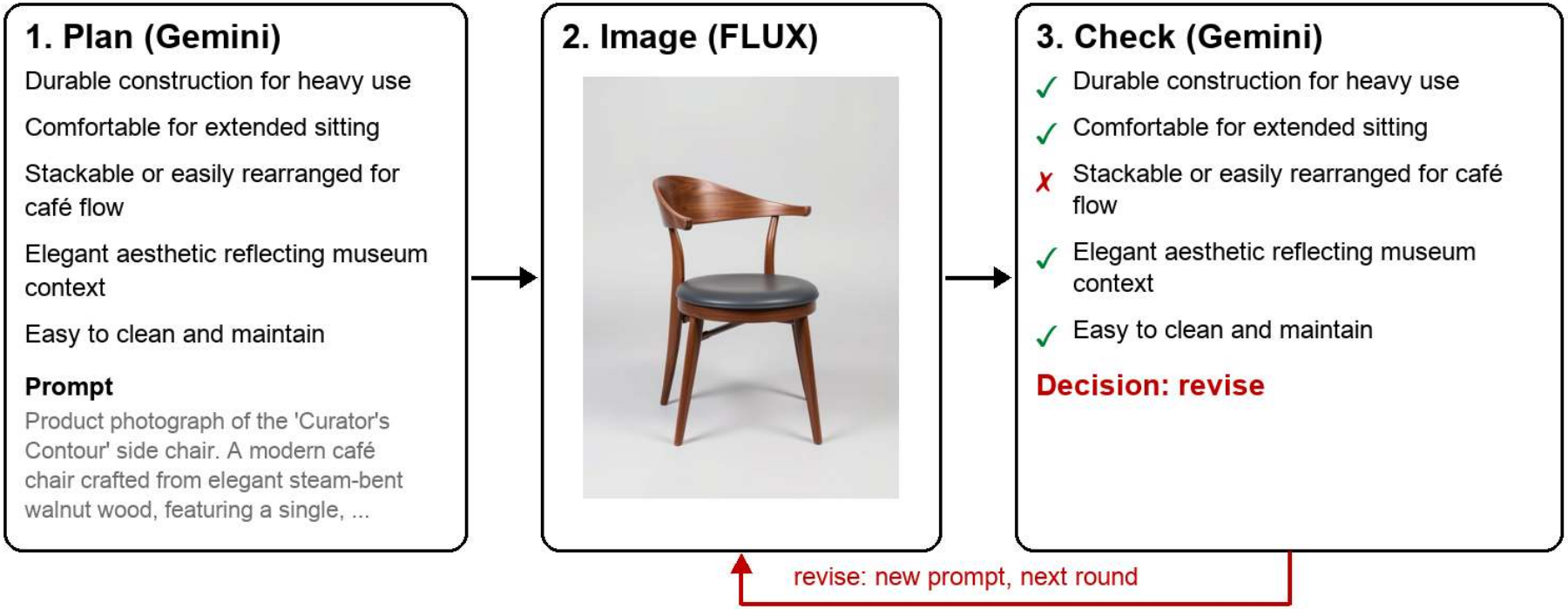
Parasuraman et al., IEEE SMC-A 2000, Fig. 3

2. Four runs of our agent

The brief and the 56 museum chairs from week 4, run four times



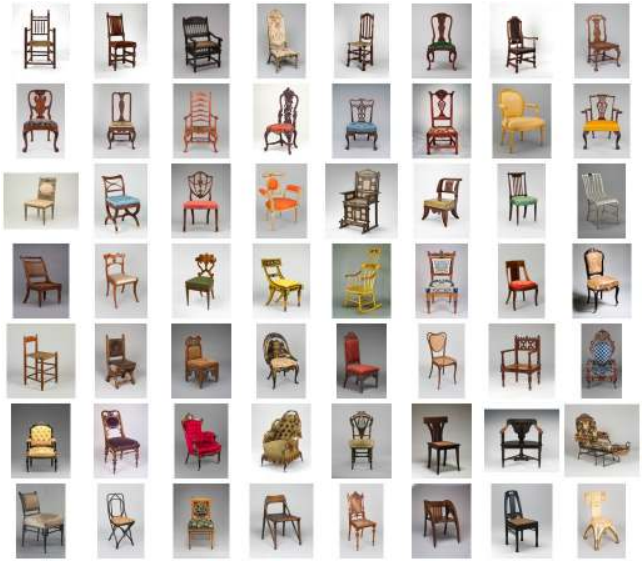
How the agent works



Run 1, round 1. Every step is saved to agent-log.jsonl.

Run 1: the agent's checklist

1. Durable construction for heavy use
2. Comfortable for extended sitting
3. Stackable or easily rearranged for café flow
4. Elegant aesthetic reflecting museum context
5. Easy to clean and maintain



Brief: “draw on the 56 historical chairs”

Run 1. Right: the reference sheet the agent was given.

Run 1: round 3 check



✓ Stackable: “As a folding chair, it is easily rearranged and can be stored compactly, fulfilling the requirement for adaptable café flow, even if not traditionally stackable.”

✗ Durable ✗ Comfortable ✗ Elegant ✗ Easy to clean

From agent-log.jsonl, run 1, round 3

Run 1: results



Round 1

1 of 5 failed, revise



Round 2

2 of 5 failed, revise



Round 3

4 of 5 failed, revise



Round 4

1 of 5 failed, revise

4 rounds took 54 seconds, and **8** of the 20 checks failed. It stopped at my limit of 4 rounds, with the last round still on revise.

Run 2: same brief and settings



Round 1
1 of 5 failed, revise



Round 2
2 of 5 failed, revise



Round 3
0 of 5 failed, done

A new checklist, with “Subtly references historical chair forms from the museum's collection”

Done at round 3

Same brief, settings and code as run 1

Run 3: my checklist



Round 1
1 of 4 failed, revise



Round 2
1 of 4 failed, revise



Round 3
1 of 4 failed, revise



Round 4
0 of 4 failed, done

1. Takes a specific shape or detail from one of the 56 chairs on the reference sheet
2. Stacks with other chairs of the same design
3. Has a seat and back that can be wiped clean
4. Light enough for café staff to carry

Run 3: checks of item 1



✗ Round 2: “The reference sheet of 56 historical chairs was not provided, making it impossible to verify if a specific shape or detail was drawn from the collection.”

✓ Round 4: “The intricate, ornate floral pattern on the chair's backrest is a clear decorative detail that draws inspiration from historical furniture styles, fulfilling the requirement to draw on the museum's collection.”

From agent-log.jsonl, run 3. Left: round 4.

Run 4: eight rounds



Round 1
1 of 5 failed, revise



Round 2
2 of 5 failed, revise



Round 3
2 of 5 failed, revise



Round 4
1 of 5 failed, revise



Round 5
2 of 5 failed, revise



Round 6
2 of 5 failed, revise



Round 7
2 of 5 failed, revise



Round 8
2 of 5 failed, revise

“Subtly references historical forms” failed in rounds 2 to 8.
Rounds 5 to 8 switched between a fabric seat for comfort and a wipe-clean seat.

The agent wrote its checklist; limit raised to 8 rounds

Four runs

	Checklist by	Images	Failed checks	Ended by
	Run 1: the agent	4	8 of 20	my round limit
	Run 2: the agent	3	3 of 15	agent said done
	Run 3: me	4	3 of 16	agent said done
	Run 4: the agent	8	14 of 40	my round limit

Done or revise: **19** decisions by the agent, none reviewed while it ran

Left: the last chair of each run

Our runs at each function

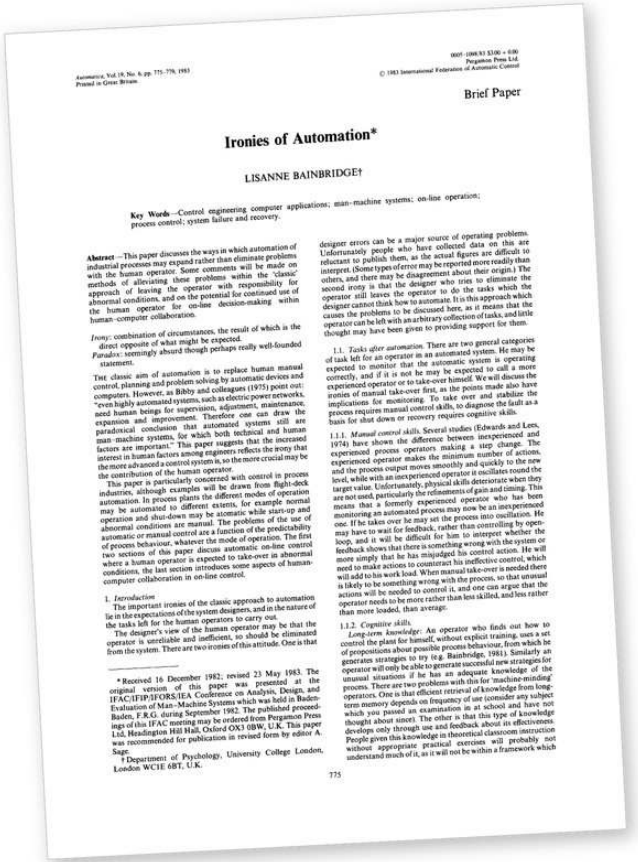


Drawn like Parasuraman et al.'s Fig. 2

My reading of the four logs

3. Ironies of automation

Today's reading: Bainbridge, “Ironies of Automation,” Automatica 1983



Automatica, Vol. 19, No. 6, pp. 775–779, 1983
Printed in Great Britain.

0005–1098/83 $3.00 + 0.00
Pergamon Press Ltd.
© 1983 International Federation of Automatic Control

Brief Paper

Ironies of Automation*

LISANNE BAINBRIDGE†

Key Words—Control engineering computer applications; man–machine systems; on-line operation; process control; system failure and recovery.

Abstract—This paper discusses the ways in which automation of industrial processes may expand rather than eliminate problems with the human operator. Some comments will be made on methods of alleviating these problems within the 'classic' approach of leaving the operator with responsibility for abnormal conditions, and on the potential for continued use of the human operator for on-line decision-making within human–computer collaboration.

Irony: combination of circumstances, the result of which is the direct opposite of what might be expected.
Paradox: seemingly absurd though perhaps really well-founded statement.

THE classic aim of automation is to replace human manual control, planning and problem solving by automatic devices and computers. However, as Bibby and colleagues (1975) point out: "even highly automated systems, such as electric power networks, need human beings for supervision, adjustment, maintenance, expansion and improvement. Therefore one can draw the paradoxical conclusion that automated systems still are man–machine systems, for which both technical and human factors are important." This paper suggests that the increased interest in human factors among engineers reflects the irony that the more advanced a control system is, so the more crucial may be the contribution of the human operator.

This paper is particularly concerned with control in process industries, although examples will be drawn from flight-deck automation. In process plants the different modes of operation may be automated to different extents, for example normal operation and shut-down may be automatic while start-up and abnormal conditions are manual. The problems of the use of automatic or manual control are a function of the predictability of process behaviour, whatever the mode of operation. The first two sections of this paper discuss automatic on-line control where a human operator is expected to take-over in abnormal conditions, the last section introduces some aspects of human–computer collaboration in on-line control.

1. *Introduction*

The important ironies of the classic approach to automation lie in the expectations of the system designers, and in the nature of the tasks left for the human operators to carry out.

The designer's view of the human operator may be that the operator is unreliable and inefficient, so should be eliminated from the system. There are two ironies of this attitude. One is that designer errors can be a major source of operating problems. Unfortunately people who have collected data on this are reluctant to publish them, as the actual figures are difficult to interpret. (Some types of error may be reported more readily than others, and there may be disagreement about their origin.) The second irony is that the designer who tries to eliminate the operator still leaves the operator to do the tasks which the designer cannot think how to automate. It is this approach which causes the problems to be discussed here, as it means that the operator can be left with an arbitrary collection of tasks, and little thought may have been given to providing support for them.

1.1. *Tasks after automation*

There are two general categories of task left for an operator in an automated system. He may be expected to monitor that the automatic system is operating correctly, and if it is not he may be expected to call a more experienced operator or to take-over himself. We will discuss the ironies of manual take-over first, as the points made also have implications for monitoring. To take over and stabilize the process requires manual control skills, to diagnose the fault as a basis for shut down or recovery requires cognitive skills.

1.1.1. *Manual control skills*

Several studies (Edwards and Lees, 1974) have shown the difference between inexperienced and experienced process operators making a step change. The experienced operator makes the minimum number of actions, and the process output moves smoothly and quickly to the new level, while with an inexperienced operator it oscillates round the target value. Unfortunately, physical skills deteriorate when they are not used, particularly the refinements of gain and timing. This means that a formerly experienced operator who has been monitoring an automated process may now be an inexperienced one. If he takes over he may set the process into oscillation. He may have to wait for feedback, rather than controlling by open-loop, and it will be difficult for him to interpret whether the feedback shows that there is something wrong with the system or more simply that he has misjudged his control action. He will need to make actions to counteract his ineffective control, which will add to his work load. When manual take-over is needed there is likely to be something wrong with the process, so that unusual actions will be needed to control it, and one can argue that the operator needs to be more rather than less skilled, and less rather than more loaded, than average.

1.1.2. *Cognitive skills*

Long-term knowledge: An operator who finds out how to control the plant for himself, without explicit training, uses a set of propositions about possible process behaviour, from which he generates strategies to try (e.g. Bainbridge, 1981). Similarly an operator will only be able to generate successful new strategies for unusual situations if he has an adequate knowledge of the process. There are two problems with this for 'machine-minding' operators. One is that efficient retrieval of knowledge from long-term memory depends on frequency of use (consider any subject which you passed an examination in at school and have not thought about since). The other is that this type of knowledge develops only through use and feedback about its effectiveness. People given this knowledge in theoretical classroom instruction without appropriate practical exercises will probably not understand much of it, as it will not be within a framework which

* Received 16 December 1982; revised 23 May 1983. The original version of this paper was presented at the IFAC/IFIP/IFORS/IEA Conference on Analysis, Design, and Evaluation of Man–Machine Systems which was held in Baden-Baden, F.R.G. during September 1982. The published proceedings of this IFAC meeting may be ordered from Pergamon Press Ltd, Headington Hill Hall, Oxford OX3 0BW, U.K. This paper was recommended for publication in revised form by editor A. Sage.

† Department of Psychology, University College London, London WC1E 6BT, U.K.

Bainbridge on the human operator

man–machine systems, for which both technical and human factors are important." This paper suggests that the increased interest in human factors among engineers reflects the irony that the more advanced a control system is, so the more crucial may be the contribution of the human operator.

This paper is particularly concerned with control in process industries, although examples will be drawn from flight-deck

Bainbridge, Automatica 1983, p. 775

Bainbridge on the tasks left to the operator

reluctant to publish them, as the actual figures are difficult to interpret. (Some types of error may be reported more readily than others, and there may be disagreement about their origin.) The second irony is that the designer who tries to eliminate the operator still leaves the operator to do the tasks which the designer cannot think how to automate. It is this approach which causes the problems to be discussed here, as it means that the operator can be left with an arbitrary collection of tasks, and little

Bainbridge, Automatica 1983, p. 775

Bainbridge on manual skills

experienced operator makes the minimum number of actions, and the process output moves smoothly and quickly to the new level, while with an inexperienced operator it oscillates round the target value. Unfortunately, physical skills deteriorate when they are not used, particularly the refinements of gain and timing. This means that a formerly experienced operator who has been monitoring an automated process may now be an inexperienced one. If he takes over he may set the process into oscillation. He may have to wait for feedback, rather than controlling by open-loop, and it will be difficult for him to interpret whether the

Bainbridge, Automatica 1983, p. 775

Bainbridge on monitoring

We know from many 'vigilance' studies (Mackworth, 1950) that it is impossible for even a highly motivated human being to maintain effective visual attention towards a source of information on which very little happens, for more than about half an hour. This means that it is humanly impossible to carry out the basic function of monitoring for unlikely abnormalities, which therefore has to be done by an automatic alarm system connected to sound signals. (Manual operators will notice

Bainbridge, Automatica 1983, p. 776

Levels of autonomy for AI agents



Feng, McDonald & Zhang, “Levels of Autonomy for AI Agents,” 2025, Fig. 1 (optional reading)

Discussion

1. Which of the agent's decisions would you take back?
2. At which of the four functions should your tool stop and ask?
3. What does your tool leave for the designer to do?

Led by this week's discussion leads, with questions from the reading posts

Lab: read an agent's log

1. Run the agent on your own brief
2. Mark one decision you disagree with
3. Change the checklist or the round limit, and run it again

Working prototype

After the break, another team tests your tool and looks for failures

Crit

1. Each team shows its tool for 3 minutes
2. Two teams and I answer these three questions
3. Write down three changes, and report back at the next crit

Could I direct it? Could I see why it made what it made? Could I revise what it made?

Next: Week 8

Simulating Users

Read before the studio:
- Park et al., “Generative Agents,” UIST '23
- Wang et al., “Large language models that replace human participants can harmfully misportray and flatten identity groups,” Nature Machine Intelligence 2025