Chapter 6

Cognitive conflicts in dynamic systems

Denis Besnard[1] and Gordon Baxter[2]

[1]University of Newcastle upon Tyne, [2]University of York

1 Introduction

In this chapter, we focus on dynamic systems i.e. systems whose state can change without direct action from the operator, such as transport and process control. Within these systems, we will adopt a psychological standpoint to address some HMI problems. We are particularly interested in cognitive conflicts, i.e. situations in which the way a system is mentally represented by its user shows some incompatibility with the system's actual behaviour. Namely, we investigate the extent to which a discrepancy between the operator's understanding of the system and what the system actually does can lead to a degraded interaction. We will restrict our discussions to flightdeck systems, based on examples from accidents in commercial aviation.

After defining cognitive conflicts, we provide two examples of aviation accidents that can be interpreted using this concept. We analyse these accidents in cognitive terms and explain how the mismatch between the crew's expectations and the actual behaviour of the aircraft contributed to the mishap. We then discuss two possible dimensions related to the remediation of cognitive conflicts (namely assistance tools and transparent systems) and provide some general guidelines on the structure of HMI in dynamic, critical, computer-based systems.

2 Critical systems in aviation

[This section was omitted]

3 What is a cognitive conflict?

Conflicts have been characterised by Dehais [3] in terms of the impossibility for a number of cooperating agents to reach a goal, for reasons including lack of resources or knowledge, contradictory objectives, or lack of agreement. In this chapter, we focus on the cognitive aspects of conflicts. A cognitive conflict results from an in-

compatibility between an operator's mental model and the process under control. The conflict often materialises as a surprise on the part of the operator. Automation surprises (e.g. [6]) for instance, are cognitive conflicts that arise when the automation (e.g. the autopilot) behaves unexpectedly.

For the scope of this paper, cognitive conflicts can be categorised using two dimensions (see Fig. 1):

Nature. An unexpected event occurred or an unexpected non-event occurred (i.e. nothing happened when the operators were expecting something to happen);

Status. The conflict is detected (the operator is alerted by a system state) or hidden (the operator is not aware of the system state).

	STATUS	
	Detected	Hidden
NATURE — Unexpected non-		
Unexpected event		

Fig. 1. A classification of cognitive conflicts

4 Examples of cognitive conflicts in dynamic systems

4.1 Unexpected non-events

[This section was omitted]

4.2 Unexpected events

In December 1995, a Boeing B757 flying at night from Miami (Florida) crashed into a 12,000ft mountain near Cali, Colombia, killing nearly all of the 163 people on board [1]. This Controlled Flight Into Terrain (CFIT) accident was attributed to the crew losing position awareness after they had decided to reprogram the FMC to implement a switch to the direct approach suggested by air traffic control (ATC).

The crew was performing a southbound approach, preparing to fly south-east of the airport and then turn back for a northbound landing. Because wind conditions were calm and the aircraft was flying from the north, ATC suggested that the aircraft could instead land directly on the southbound runway (see Fig. 2). The approach for this landing starts 63 nautical miles from Cali at a beacon called TULUA, followed by another beacon called ROZO (subsequently re-named PALMA). Because the crew knew they had missed TULUA when the direct approach was suggested, they attempted to proceed directly to ROZO. They therefore reprogrammed the FMC and intended to enter ROZO as the next waypoint to capture the extended runway centre-line. However, when the crew entered the first two letters of the beacon name

("RO") in the FMC, ROMEO was the first available beacon in the list, which the crew accepted. Unfortunately, ROMEO is located 132 miles east-northeast of Cali. It took the crew over a minute to notice that the aircraft was veering off on an unexpected heading. Turning back to ROZO put the aircraft on a fatal course, and it crashed into a mountain near Buga, 10 miles east of the track it was supposed to be following on its descent into Cali.
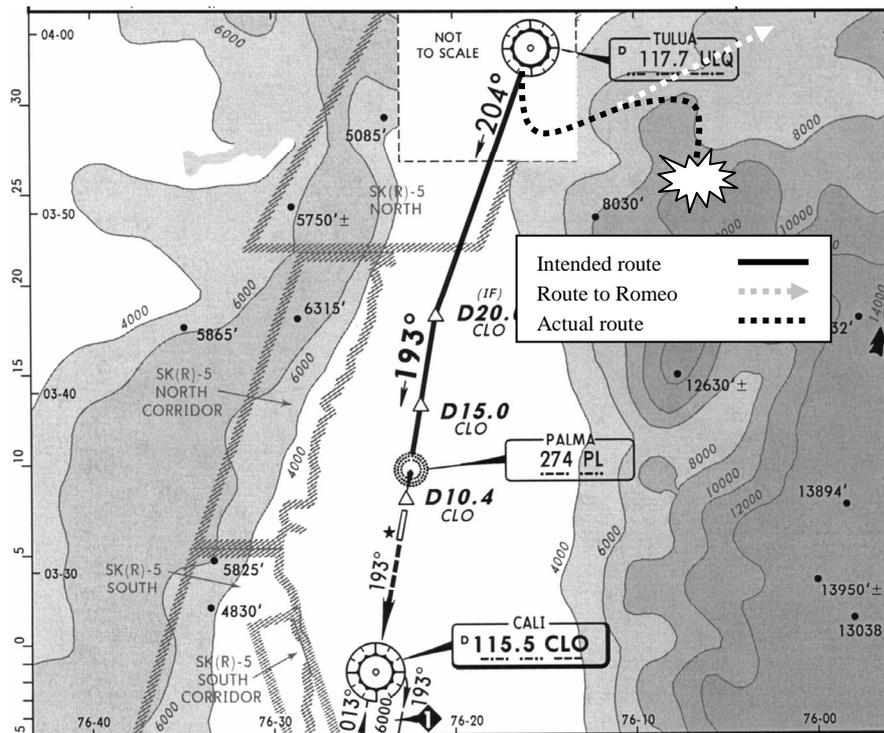

Fig. 2. Partial, amended chart of the approach to runway 19 (southbound) at Cali.
© Reproduced with permission of Jeppesen Sanderson, Inc.

The inquiry commission noticed several failures in the crew's performance, most notably:

- in the acceptance of ATC guidance without having the required charts to hand;
- in continuing the initial descent while flying a different flight plan;
- in persisting in proceeding with the (new) southbound approach despite evidence of lack of time.

After erroneously entering the co-ordinates of the ROMEO beacon into the FMC, there was a delay before the crew noticed the aircraft's unexpected behaviour. This created the need to re-evaluate and correct the aircraft position and trajectory. The time it took the crew to perform these actions, combined with the erroneous follow-

ing of the initial descent plan, put the aircraft on a collision course with a mountain. This case highlights the criticality of delays between an action and the detection of its inappropriate outcomes. The crew were in a very difficult situation in that they were trying to reach a beacon without knowing (as recorded on the CVR) what their precise position was.

The Cali accident was exceptional in that beacon names are supposed to be unique in the first two characters for any particular airspace. However, an aspect of the selection mistake is related to the frequency gambling heuristic [4]. People operating in situations perceived as familiar tend to select actions based on previous successes in similar contexts. Because the workload was extremely high when the flight path was being reprogrammed, and because of the exceptional problem with the beacons database, the crew did not immediately detect their mistake. The confusion between the ROMEO and ROZO beacons postponed the detection of the cognitive conflict, thereby delaying recovery from the mistake and worsening the consequences. This simple analysis illustrates that the longer the delay between an action and (the detection of) its outcome, the more difficult it is to recover if that action is subsequently judged as being erroneous.

5 Discussion

These cases provide an initial basis for characterising the situations in which cognitive conflicts occur. The common features are:
- a complex dynamic system;
- the occurrence of an undetected technical problem or the occurrence of an undetected human error;
- the poor predictability of the system's behaviour (albeit for different reasons in the cases considered here);
- failure to reject initial plans in a timely manner.

We have defined what we mean by a cognitive conflict and illustrated the concept using some examples. We now consider what technical solutions could be used to help manage conflicts on the flightdeck.

6 Human cognition and modern cockpits' evolution

[This section was omitted]

7 Guidelines

7.1 Better assistant tools

Any assistant tools that are developed need to take appropriate account of the following features, if they are to help increase the dependability of HMI in critical systems:

- **Timeliness** The advice delivered by assistant tools has to be timely. The span of the anticipation of the system is a matter of trade-off. The longer the span, the earlier events can be forecast. However, more competing hypotheses will then have to be analysed and eventually brought to the operator's attention.
- **Intention** Capturing what the operator wants remains a very important issue. This would help in avoiding pilots misinterpreting symptoms when they cannot easily be interpreted meaningfully. Current assistant systems only match the operator's actions against reference plans, the downside being that safe violations [2] cannot receive support. These would require operators to turn the assistant off, which is what line pilots sometimes do with autopilots.
- **Integration** Today, most of the advice given to pilots uses the visual and aural channels. Using more of the kinaesthetic channel, e.g. through vibrations (as for stall warnings via the control column), would help to diminish the visual processing load and de-clutter flightdeck displays. On another dimension, how the various functions and subsystems are integrated with one another (e.g. using similar interface principles for multifunction displays) is of importance.
- **Troubleshooting support** The information needed by operators to control a process is different from that needed to troubleshoot it. Therefore, beyond advising on forecast events, assistant tools should provide support for troubleshooting. Namely, assistants need to provide more than raw data about the system's status. Instead, operational help including a holistic view of available resources and deadlines, relative likelihood of causes of problems, technical solutions and associated risks should be available to operators.
- **Evolution** Any new automation needs to take account of existing automation and related practices, and planned future automation. Each piece of equipment will have an impact on the flight crew's cognitive resources.

7.2 Supporting transparent flightdecks

Designers should consider the following issues when developing flightdeck systems:
- **Predictable systems** Systems should be intuitive and predictable. Automation provides reliability only when its actions are understood by the human operators. In other words, there is a need to reduce the operational complexity induced by the way technology is deployed.
- **Systems with direct understanding of inner structure required** Today, modern cockpits are of the indirect interaction type. Pilots program flight systems and then let the software perform the command. In some cases, software systems even filter human actions, to the extent of sometimes preventing them. Such an evolution was driven by safety concerns. However, the initial intention has sometimes been stretched beyond human understanding capabilities, thereby impairing the viability of pilots' mental models. What is needed is not to revert to the classical cockpit but to design a computer-based cockpit that provides the same level of transparency.
- **Computers should mainly be monitoring/advisory systems** The automation should take last resort emergency decisions (e.g. pull up in front of an

obstacle) only if the corresponding situation can be unambiguously characterised as time-critical and not requiring any human intervention. Responsibility for such decisions should remain with human operators as late as possible. This is actually the case for e.g. the Airborne Collision Avoidance System but some mode changes and reversions occur in non-critical situations and unduly take the operator out of the control loop, thereby contributing to losses in situation awareness.

8 Conclusion

Cognitive mismatches are a generic mechanism that is potentially involved in any control and supervision activity. They reveal an incompatibility between the mental structure of the system that the operator maintains and the actual system's structure. The occurrence of cognitive mismatches can be facilitated by over-computerised environments if opaque automation's decision rules trigger misunderstood system behaviours. Of course, computer-based critical systems do not necessarily trigger errors but given the increased complexity of the situations the software controls, they increase the likelihood of cognitive conflicts. Because the failure of complex socio-technical systems is rarely a mere technical issue, we hope that the cognitive approach adopted in this paper is a contribution to a better understanding of the contribution of HMI to dependability in critical environments, and of potential research avenues. Also, our guidelines might offer starting points for a new reflection on further integration of cognitive features into the structure of computer-based systems in critical environments.

References

[1] Aeronautica Civil of the Republic of Colombia (1996) Controlled flight into terrain American Airlines flight 965 Boeing 757-233, N651AA near Cali, Colombia, December 20, 1995 (Aircraft Accident Report).
[2] Besnard, D and Greathead, D (2003) A cognitive approach to safe violations. Cognition, Technology & Work, 5, 272-282.
[3] Dehais F (2004) Modélisation des conflits dans l'activité de pilotage. Doctoral dissertation, ONERA, France.
[4] Reason J (1990) Human error. Cambridge University Press, Cambridge, UK.
[5] Sarter N, Woods DD (1995) How in the world did we ever get into that mode? Mode error and awareness in supervisory control. Human Factors, 37:5-19.