Securing the Interface:
Safety-Critical Interaction between Humans and Mobile Robots

Extended Abstract: 4th IET System Safety Conference, 27.10.2009

Peter Bernard Ladkin
Faculty of Technology and CITEC, University of Bielefeld
Causalis Limited

Introduction

Forty years ago, if you wanted to park your car, you had to control the speed, acceleration and manoeuvring yourself. Today, you can buy a car that parks itself. Forty years ago, if you wanted to remove dust from your house, you had to move around it with a suction device, pointing the opening here and there. Today, you turn a flat round object on and go make a cup of coffee. Forty years ago, if you wanted some behavioral-safety principles for robots, there were Asimov's laws. Today, if you want some behavioral-safety principles, there are Asimov's laws. (And some health-and-safety laws requiring interlocks to prevent people being within the reach-radius of fixed-base industrial robots. And there are people working on principles of engagement for artifical warriors, which we may imagine are not likely to cover the full range of robotic safety needs.) To my mind, the principles have not kept pace with the practice. I want to make a start on catching up.

Some Abstract Technical Preliminaries

Analytic philosophers and mathematicians often begin by defining the terms they are about to use. So shall I. By an “agent”, I understand some entity which exhibits behavior. By “behavior” of an agent A, I understand that the state of the world changes in some way describable only through referring to A. By a “change”, I mean a pair of states S, T, with T later than S, where S and T are incompatible (that is, they cannot co-occur). By a “state”, I mean a snapshot of the world, taken at a moment in time: whatever you can see in this snapshot belongs to the state, and what you cannot, doesn't. This is all more or less a verbal description of the ontology of temporal logic.

A system is a collection of agents. There are objects which do not belong to the system, but which exhibit joint behavior with the system (objects in the system). These objects and their properties constitute the “environment” of the system. The “world” consists of the system and (read: union) its environment.

A more complete set of definitions of terms necessary for the analysis of safety-critical systems may be found at URL:

www.causalis.com/DefinitionsForSafetyEngineering.pdf

Engineered Multiagent Cooperative Functions

There are human agents, called operators (or air traffic controllers, or drivers, or pilots) and artifical agents. Artificial agents (AA) are machines designed and built by engineers. We can make a further distinction between:

- independent artificial agents (IAA) such as a robotic vacuum cleaner, those which encompass a self-contained system with its own goals, as seen by the engineers which designed them, and

- component artificial agents (CAA), such as an autopilot on a modern highly-automated commercial aircraft, which are designed to perform tasks in cooperation with other agents in a system designed to serve a purpose going beyond that of the CAA.

We often call IAAs robots, and think of them as humanoid, but they can be airplanes, cars or trains. These agents, both IAAs and CAAs, are machines situated in a real-world environment. Engineers usually build artificial agents, CAAs and IAAs, because they want to achieve some behavioral goal with them. These agents are “teleological systems”, systems with a purpose, to be contrasted with systems without explicit purpose, such as predator-prey systems in ecology. Not only CAAs but also IAAs may sometimes interact with human agents to achieve the goal. An autopilot CAA in a commercial aircraft is designed to interact with human pilots with whom (we hope) it cooperates and vice versa. However, a humanoid robot such as an artifical servant, an IAA, must also cooperate with you in order to pour coffee into the cup you are holding. So the distinction between IAA and CAA is rather more fuzzy than it might at first appear.

When building a system, an engineer has in mind the goal, the function, of the system; that it must generally work with, cooperate with, other agents in order to achieve the goal. Hence the concept of “engineered, multiagent, cooperative” function, or EMC function, which I hope is self-explanatory. I am concerned here with the safety of EMC functions.

Safety of EMC Functions

While executing an EMC function there arises the safety question: can a

situated agent behave in such a way as to cause damage (whatever we may think of as damage)? There is a second, related, question: can we, or how can we, assure the integrity of the agent, or the system in which many agents are cooperating, so that its intended behavior is indeed how it will behave?

The attitudes towards safety vary widely amongst areas in which EMC functions occur. In aviation, safety is dealt with in detailed standards for the reliability of critical components, in overall risk-based goals for systems and subsystems and through the identification of operational hazards and their mitigation through airworthiness directives pursuant upon careful investigation of incidents and accidents. For industrial fixed-base robots, measures are often based upon motion interlock systems, so that a human may only be present within the reach-volume when the robot is disabled. For mobile robots such as the anthropomorphic robots on which CITEC researchers work, safety measures in place are as yet relatively sparse. I was told a story by Ludwig Seitz in which an acquaintance was present at a technical meeting (not at CITEC!), when the android in the corner of the meeting room began to move rapidly. It turned out that a researcher elsewhere was loading new software into the robot and this software was being directly executed without her realising it.

Both of the above questions gain importance as the behavior of humanoid robots becomes more complex, interacting with humans through gestures and speech. But they are by no means solved in more traditional areas such as commercial aviation, either.

Humanoid Robots: Due Diligence and Beyond

The incident with the lab robot above may be regarded as a moderately straightforward issue: there is a duty of care of the facility provider towards the personnel interacting with the robots and some sort of interlock could be provided to ensure that the robots don't activate without the explicit intent of some person, upon whom a duty of care would also fall to clear the area. There is also a medium-term integrity issue, also concerned with due diligence, in protecting the robots against running malware, and detecting a circumstance in which malware has become installed. Addressing these issues may be tricky, however, they are not directly related to the EMC functionality of the human and machine, which is my main concern here.

Consider that a robot able to move around and pour you coffee when you ask for it may also be capable of pouring the hot liquid on your head. There are, broadly, two reasons why it may do so. One is that it mistakes the space

which your head occupies for space in which it expects a cup to be. The other is that it is programmed so as (it "intends", if Bert Dreyfus and John Searle will forgive me the anthropomorphism) to pour coffee on your head.

This is a question of EMC functionality. Human and machine are supposedly cooperating in getting a cup of coffee poured. In the first case, the human and the machine have conflicting views on the state of the world: the machine "imagines" the cup is where the head is, and we may be moderately sure that the human thinks the cup is elsewhere. In the second case, there is a conflict of intent: the human imagines the robot would be pouring into the cup and the robot has a different "intent".

Such intuitive analysis points towards some relatively abstract principles which might help us to ensure safety in EMC functions. Or at least to keep our heads free from coffee.

Principles of Design for Safety: the RCMC

The first case, in which IAA and human have different ideas about where the cup is, leads to what I call the

> ***Rational Cognitive Model Coherence (RCMC) Criterion**: all participants in an EMC function maintain mutually coherent views of the state of the world (the "global state")*

Another way of saying this is that the union (or conjunction) of the partial states of the world is coherent (that is, there is no contradiction in the conjunction).

RCMC arose first from my analysis of the Überlingen mid-air collision, in which the commander of the Russian crew was searching for a third airplane with which he believed they might also have a conflict, which third airplane did not exist. See *crpit.com/confpapers/CRPITV47Ladkin.pdf* in Volume 47 of the CRPIT series, in which I formulated the notion of Rational Cognitive Model (RCM). An RCM is something associated with an agent. It has an associated notion of state:

> ***RCM state:** The RCM state of an agent A at a time T consists of what the agent thinks/believes/stores is the relevant (partial) state of the world.*

Its importance lies in that, if the RCM state reflects the true state of the

world, RCMC says that all agents in an interaction shall have coherent RCM states. My experience so far with the concept of RCMs suggests that investigations largely involve (small-) finite-state-machine engineering techniques.

Violations of RCMC have occurred in a number of recent accidents to highly automated commercial aircraft, for example the 1992 Lufthansa landing accident in Warsaw, the 1998 Philippine Air Lines Bacolod runway overrun, the 2004 Transasia Taipei-Sungshan runway overrun, the 2007 TAM Sao Paolo landing overrun, the 2002 Überlingen midair collision as I mentioned, and (it appears from the preliminary report) the recent 2009 Turkish Airlines Amsterdam accident, amongst others.

Indeed, it could also be argued that a violation of RCMC is involved in all instances of “mode confusion”, in which an aircraft crew do not understand, or are mistaken about, the “mode” in which the autopilot and autothrottle are functioning. Mode confusion is generally regarded as a “new” type of human-machine error which arose with the first generation of complex digital cockpit automation. Consider that autopilots can be designed with such hierarchical state machines as “Statecharts”, devised by David Harel for use in digital avionics. Mode confusion arises when the high-level state of the autopilot is different from what the pilot, the human agent, thinks it is. Hence a violation of RCMC. However, as soon as the human becomes unsure (“*What is it doing now?”)* and realises that heshe no longer knows what mode the autopilot is in, the RCMC is again fulfilled: knowing that you don't know regains coherence. But it doesn't make the piloting job much easier: one still has to figure out what the airplane is doing (it is often best to disengage the autopilot!). I shall return to this below.

RCMC bears close analogy to cache coherence in parallel computing. People designing memory management systems for highly parallel computers regard cache coherence as a design principle. (There is to my knowledge one correct memory-management algorithm which violates it, designed by Afek, Brown and Merritt in the early 1990's.) However, there is one safety-critical multi-agent algorithm implemented in kit required on every high-performance commercial aircraft flying, namely the collision-avoidance system TCAS II, which is guaranteed to violate RCMC, and which violation was demonstrably a causal factor in the 2002 Überlingen collision (there was also a non-related TCAS II phenomenon, known since 2000, which also played a direct causal role and which one could arguably characterise as an algorithm-integrity issue).

RCMC is one criterion that one could use as a design principle for situated

interaction with robots, that will ensure that certain kinds of safety problems do not arise. There are others.

Principle: The Bounded-Rationality Criterion

Consider, for example, that agents, both human and robotic, have bounded rationality. It takes a certain amount of time to think, or compute, and there is only a certain finite amount of working memory available for such thinking. Bounded rationality has more components than time and space, however. One is bounded perception. Humans, for example, can discriminate up to 7 or 8 different sounds concurrently, but more are perceived as cacophony. This is an important restriction in the design of warning tones in sophisticated aircraft, for example. A similar principle for visual perception of arrays of warning lights is not yet known. Another component is bounded reasoning and decision-making, which has been comparatively well-studied in AI since it was first addressed by Herb Simon 50 years ago. One can formulate a

> ***Bounded-Rationality Criterion (BRC)****: in context A there shall arise no state in which a safety-related decision or action to be made by agent A requires more reasoning/decision/executive capability that that available to agent A.*

I do not know at this point which techniques are appropriate to address BRC questions.

For an example of the importance of bounded-rationality considerations, consider the recent accident to a British Airways Boeing 777, Flight 038, on short-final approach to London Heathrow airport in January 2008. The aircraft lost significant thrust to both engines a few hundred feet above landing elevation, and with skilled piloting made it “over the fence” and flopped on to the grass preceding the runway. Except for the poor guy with the broken leg, all walked away. The pilots had an exceptionally short time in which to decide what to do and execute it, and very limited information with which to decide and act. So it could well be that BRC was violated.

Of course, the failure of BA 038 was not only unanticipated by the pilots, but also by designers, indeed it is not yet determined what the failure was, although there appears to be one strong candidate, in the form of a previously-unknown fuel-system-icing phenomenon. Adherence to BRC cannot protect you from unanticipated phenomena, but that observation does not preclude its use during hazard analysis at design time (at which by

definition only anticipated phenomena are considered).

Principle: the Mutual Cognisance of Relevant Parameters

Let us return to the case of the autopilot mode confusion, in which the pilot has recognised the discrepancy: "*What is it doing now?*". I noted above that this situation fulfilled the RCMC, but it is still not a satisfactory situation to be in. One wants to know the true values of the important parameters, not to be unsure of them. Hence the:

> ***Mutual Cognisance of Relevant Parameters (MCRP) Criterion****: Let R be a set of parameters of which knowledge is required by all agents in a set S cooperating to reach goal G. Then all agents in S must be cognisant of (must have "cached") the true values of R.*

"True values" here means the actual values that those parameters have in the real world. Also note that MCRP may apply many times in the cooperation behavior of the EMC function of attaining G: if V is the set of all agents engaging in the function, there might be a subset S of V for which knowledge of the values of parameters P are necessary, and another subset S_1 of V for which knowledge of parameters P_1 are necessary. MCRP would apply to both S concerning P and S_1 concerning P_1.

It should be clear, I hope, how MCRP precludes a "*What is it doing now?*" situation. What is not so clear is how one may ensure that MCRP is always fulfilled!

Another point to consider is that the exact real-world value of a parameter may not be needed: it is not necessary to know one's airspeed to within five decimal places in any regime of flight. It suffices to know it to within a knot or two. So the granularity of the parameters also enters into the MCRP, although I have not explicitly formulated it.

Some accidents and incidents that involved violation of MCRP are 1990 Indian Airlines in Bangalore, 1992 Air Inter at Mt. St.-Odile near Strasbourg, 1993 Lufthansa in Warsaw (again), the 2005 Malaysian Airlines in-flight near-upset off Perth, according to the preliminary report the 2009 Qantas accident near Learmonth in Western Australia, and another apparently similar 2009 incident also to Qantas noted in the same report.

And again the 2002 Überlingen mid-air collision. Indeed, not only the collision. It is interesting to note that TCAS interactions in controlled airspace are guaranteed to violate MCRP. One causally-relevant agent, the ATC, is not

informed of the relevant state (two aircraft performing an altitude change) until well in to the interaction, and is likely to believe the state to be other than it is (that is, likely to think that aircraft are maintaining assigned altitude) until heshe is informed by the participating aircraft of a resolution manouvre.

Other circumstances in which MCRP could play a role are in take-off and landing with modern highly-automated aircraft. For example, landing uses three braking systems: ground spoilers (to dump lift and thereby put most of the aircraft weight on the wheels, enabling heavier wheel braking; also some aerodynamic-drag braking), thrust reverse (useful in the first few seconds of landing, when the airplane is rolling fast; or on icy or otherwise slippery runways in which wheel braking is not as effective), and wheel brakes. In one modern highly-automated aircraft, the spoilers and thrust reverse, when activated/armed in flight, deploy without pilot involvement when the aircraft touches down and has sufficient weight on its wheels (called “weight on wheels” or WoW, verified usually by a squat switch activated through compression of the oleo struts on the main landing gear). If the aircraft does not have sufficient WoW, ground spoilers will not deploy and thus wheel braking is not as effective. This happened during the 1993 Warsaw landing accident. One can imagine that had the Warsaw pilots known for all of those interminable seconds that they did not have WoW, and that thereby TR, for example, was inhibited and could not be activated, then they could have decided to violate their Standard Operating Procedures in the interests of safety, firewall the throttles and go around. If they had done that, one could imagine that all would have been well.

There are other landing accidents with this type of aircraft in which the pilots have not been cognisant of relevant parameters. (Nevertheless, I should point out that this aircraft type has fewer landing accidents per operational hours flown than any other comparable aircraft in history!) The list of relevant parameters R for landing in this type of aircraft could, for example, contain:

- Weight on Wheels (WoW) /no WoW
- Spoiler enabled/activated/not activated
- Thrust Reverse enabled/activated/not activated
- Current rate of deceleration
- Autobraking enabled/activated/not activated

Compiling a similar list for take-offs is more tricky. Information would be needed about the systems (all engines developing suitable thrust), a suitable rate of acceleration, and whether the aircraft will be able to depart the runway safely from the current state, amongst other things. However, discussion amongst pilots, for example recently on the professional-pilots

forum *pprune.org,* indicates that decision-making on take-off can be subtle, and it would be premature for me to offer a parameter list until the justifications for it are better understood.

Principle: Procedural Completeness Criterion (PCC)

There is a criterion which may seem obvious when stated:

> ***Procedural Completeness Criterion (PCC):*** *for every reachable combination of R parameters, there is an explicit procedure for each agent involved in the task.*

As obvious as this criterion may seem, I know of at least two cases in which it has not been fulfilled, as follows.

In his recent Ph.D. thesis, Bernd Sieker performed an Ontological Hazard Analysis of the protocol for train dispatching on German non-state railways, a protocol defined in German administrative law. Train dispatching is a purely verbal protocol used on "lightly travelled" single-track lines on which there is no signalling. Sieker produced a sequence of steadily more refined state machines representing the intermediate states in a successful train dispatch at different levels of abstraction, down to SPARK code implementing the communications (liveness requires that all train dispatching eventually succeeds; safety requires that dispatched trains do not collide). He found that, with a number of those machines, the train-dispatching protocol did not explicitly require transitions out of all non-final states. This situation violates PCC. (He was able to add such transitions using a certain amount of interpretation and common sense: it could be argued, I suppose, that the laws contained these implicitly.)

Causalis Limited has been involved in an analysis of a high-profile incident in which the PCC appeared to be violated. We are pleased to report that this situation no longer pertains.

Completeness?

Are there more such principles? First, at a more refined level of detail than that with which I have been concerned here, Harold Thimbleby has proposed design principles, in his recent book *Press On* (MIT Press, 2007), for certain sorts of interactions with programmable-digital devices, say, user-programmable so-called "smart" medical devices. Thimbleby has recently proposed "UI model discovery" as a verification technique, pointing to what

one might argue are failures in the design of devices on the market. Thimbleby's observations are acute and obviously pertinent.

Second, consider an avatar, of which we have one called MAX at CITEC. MAX exists on a computer screen, but interacts with people standing before the screen in real time. He also greets visitors to the Heinx Nixdorf Computer Museum in Paderborn. The safety principles I have enumerated are limited in application to an avatar. They obviously concern physical behavior. The physical behavior of an avatar consists fundamentally only in changing pixels on a computer screen and actuating loudspeakers. Its semantic behavior – its gestures and speech, and the effect they have on you - is something else! If the avatar is giving advice to an operator performing safety-critical operations, the consequences of inappropriate semantic behavior could be consequential. It is not clear how far the principles I have formulated may suffice for semantic behavior.

Third, I have briefly mentioned the question of how we secure the integrity of a robot's behavior. This question is broader that that of physical safety - it arises also for an avatar. There may be many types of integrity failure that could result in coffee being poured on your head by a robot. It could be that two incompatible versions of agent control modules were inadvertently loaded. It could be that an unverified and dangerously faulty rapid-prototype module was loaded. It could be that malware had infiltrated the loaded control system.

These integrity issues fall within the general area of dynamic access control for robot systems. One can imagine that it should be made impossible to load conflicting control modules simultaneously, or the robot be incapacitated when it happens. One can imagine that only verified modules and configurations be loaded. One can imagine procedures for verifying the continued integrity of loaded modules.

Dealing with such integrity issues requires that a comprehensive “threat model” (of integrity violations) exists, and that effective principles and verification methods exist for ensuring integrity according to this model. The formulation of such a threat model, and principles to counter the threats, lies above and beyond the principles I have enumerated above.

From these considerations it should be clear that the principles enumerated above are far from complete and it would be premature to think we might now know what the high-level principles of user interaction should be.

Too Strong?

It might well be considered that some of these principles are too strong. I mentioned the January 2008 accident to a British Airways Flight 038, and that the BRC was violated. I also noted that the circumstances in which this happened, whatever they might turn out to be (the causal inquiry is still running) likely could not reasonably have been anticipated, because even with hindsight they are proving elusive.

Second, there appears to be an issue with ice collecting on the face of the Fuel/Oil Heat Exchanger (FOHE) under certain circumstances. This is a phenomenon which was only discovered during the causal investigation into the accident, and could cause restriction of fuel flow, although it is currently not known whether it is causal to the accident to BA 038. It would be fatuous to hold designers in retrospect to a violation of MRCP for not providing the pilots with some means to detect this then-unknown condition.

Hindsight is a wonderful thing, and will be shown to be a wonderful thing when the investigation into BA 038 is concluded. To my mind, it follows from these considerations not that the principles I have formulated are too strong, but that they are analysis principles primarily for use during design and ongoing operations. They can be applied at the least to those conditions which arise during the hazard analysis of a safety-critical system which is being built.

Acknowledgements

This paper contains some joint work with Bernd Sieker, and has benefitted greatly from discussions with Jan Sanders. Bernd is working on the safety principles and detailed application in the mobile-robot domain.