19

Architectures and Ethics for Robots
Constraint Satisfaction as a Unitary Design Framework

Alan K. Mackworth

Introduction

Intelligent robots must be both proactive and responsive. That requirement is the main challenge facing designers and developers of robot architectures. A robot in an active environment changes that environment in order to meet its goals and it, in turn, is changed by the environment. In this chapter we propose that these concerns can best be addressed by using *constraint satisfaction* as the design framework. This will allow us to put a firmer technical foundation under various proposals for codes of robot ethics.

Constraint Satisfaction Problems

We will start with what we might call Good Old-Fashioned Constraint Satisfaction (GOFCS). Constraint satisfaction itself has now evolved far beyond GOFCS. However, we initially focus on GOFCS as exemplified in the constraint satisfaction problem (CSP) paradigm. The whole concept of constraint satisfaction is a powerful idea. It arose in several applied fields roughly simultaneously; several researchers, in the early 1970s, abstracted the underlying theoretical model. Simply, many significant sets of problems of interest in artificial intelligence can each be characterized as a CSP. A CSP has a set of variables; each variable has a domain of possible values, and there are various constraints on some subsets of those variables, specifying which combinations of values for the variables involved are allowed (Mackworth 1977). The constraints may be between two variables or among more than two variables. A familiar CSP example is the Sudoku puzzle. The puzzle solver has to fill in each square in a nine by nine array of squares, with a digit chosen from one through nine, where the constraints are that every row, every column, and every three by three subgroup has to be a permutation

Based, in large part, on Mackworth, Alan. "Agents, Bodies, Constraints, Dynamics, and Evolution." *AI Magazine*, Volume 30, Issue 1, Spring 2009, pp. 7–28. Association for Advancement of Artificial Intelligence, Menlo Park, CA.

of those nine digits. One can find these solutions using so-called arc consistency constraint satisfaction techniques and search; moreover, one can easily generate and test potential Sudoku puzzles to make sure they have one and exactly one solution before they are published. Constraint satisfaction has its uses.

Arc consistency is a simple member of the class of algorithms called *network consistency algorithms*. The basic idea is that one can, before constructing global solutions, efficiently eliminate local nonsolutions. Because all of the constraints have to be satisfied, if there is any local value configuration that does not satisfy any of them, one can throw that tuple out; that is called a "no good." The solver can discover (that is, learn) those local inconsistencies, once and for all, very quickly in linear, quadratic, or cubic time. Those discoveries give huge, essentially exponential, savings when one does start searching, constructing global solutions, using backtracking, or other approaches. The simplest algorithm is arc consistency, then path consistency, then k-consistency, and so on. For a detailed exposition and historical perspective on the development of those algorithms, see Freuder and Mackworth (2006). Since those early days, network consistency algorithms have become a major research industry. In fact, it has now evolved into its own field of computer science and operations research called *constraint programming*. The CSP approach has been combined with logic programming and various other forms of constraint programming. It is having a major impact in many industrial applications of AI, logistics, planning, scheduling, combinatorial optimization, and robotics. For a comprehensive overview, see Rossi, van Beek, and Walsh (2006). Here we will consider how the central idea of constraint satisfaction has evolved to become a key design tool for robot architectures. This development, in turn, will allow us to determine how it could underpin proposals for codes of robot ethics.

Pure Good Old-Fashioned AI and Robotics (GOFAIR)

The way we build artificial agents has evolved over the past few decades. John Haugeland (Haugeland 1985) was the first to use the phrase Good Old-Fashioned AI (GOFAI) when talking about symbolic AI using reasoning and so on as a major departure from earlier work in cybernetics, pattern recognition, and control theory. GOFAI has since come to be a straw man for advocates of subsymbolic approaches, such as artificial neural networks and evolutionary programming. AI at the point when we discovered these symbolic techniques tended to segregate itself from those other areas. Lately, however, we see a new convergence. Let me quickly add here that there was a lot of great early work in symbolic programming of robots. That work can be characterized, riffing on Haugeland, as Good Old-Fashioned AI and Robotics (GOFAIR) (Mackworth 1993).

GOFAIR Meta-Assumptions

In a cartoon sense, a pure GOFAIR robot operates in a world that satisfies the following meta-assumptions:

- Single agent
- Serial action execution order
- Deterministic world
- Fully observable, closed world
- Perfect internal model of infallible actions and world dynamics
- Perception needed only to determine initial world state
- Plan to achieve goal obtained by reasoning and executed perfectly open loop

There is a single agent in the world that executes its actions serially. It does not have two hands that can work cooperatively. The world is deterministic. It is fully observable. It is closed, so if I do not know something to be true, then it is false, thanks to the Closed World Assumption (Reiter 1978). The agent itself has a perfect internal model of its own infallible actions and the world dynamics, which are deterministic. If these assumptions are true, then perception is needed only to determine the initial world state. The robot takes a snapshot of the world. It formulates its world model. It reasons in that model, then it can combine reasoning that with its goals using, say, a first-order theorem-prover to construct a plan. This plan will be perfect because it will achieve a goal even if it executes the plan open loop. So, with its eyes closed, it can just do action A, then B, then C, then D, then E. If it happened to open its eyes again, it would realize "Oh, I did achieve my goal, great!" However, there is no need for it to open its eyes because it had a perfect internal model of these actions that have been performed, and they are deterministic and so the plan was guaranteed to succeed with no feedback from the world.

CSPs and GOFAIR

What I would like you, the reader, to do is to think of the CSP model as a very simple example of GOFAIR. There are no robots involved, but there are some actions. The Sudoku solver is placing numbers in the squares and so on. In pure GOFAIR there is a perfect model of the world and its dynamics in the agent's head, so I call the agent then *an omniscient fortune-teller*, as it knows all and it can see the entire future because it can control it, perfectly. Therefore if these conditions are all satisfied, then the agent's world model and the world itself will be in perfect correspondence – a happy state of affairs, but, of course, it doesn't usually obtain. However, when working in this paradigm we often failed to distinguish the agent's world model and the world itself, because there really is no distinction in GOFAIR. We confused the agent's world model and the world, a classic mistake.

A Robot in the World

Now we come to think about the nature of robots. A robot acts in a world. It changes that world, and that world changes the robot. We have to conceive of a

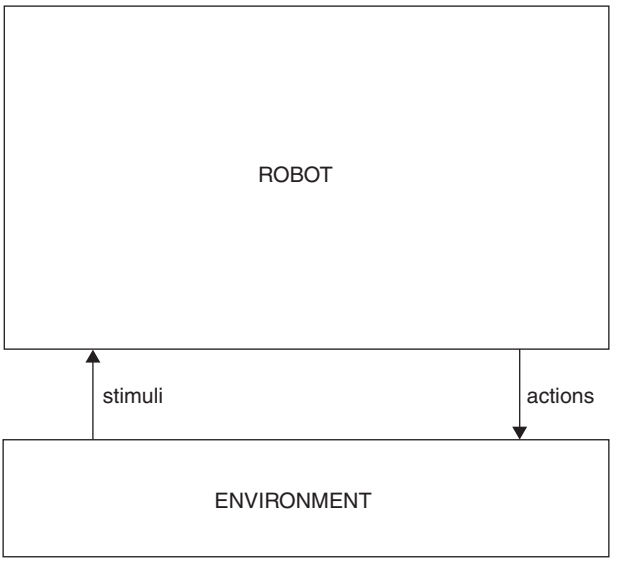

Figure 19.1. A Robot Co-evolving with its Environment.

robot in an environment and performing actions in that environment; and the environmental stimuli, which could be sensory or physical stimuli, will change the robot. Therefore, think of the robot and its environment as two coupled dynamical systems operating in time, embedded in time, and each changing the other as they co-evolve, as shown in Figure 19.1.

They are mutually evolving perpetually or to some future fixed point state, because, of course, the environment could contain many other agents who see this robot as part of their environment.

Classic Horizontal Architecture

Again, in a cartoon fashion, consider the so-called three-boxes model or the horizontal architecture model for robots. Because perception, reasoning, and action are the essential activities of any robot, why not just have a module for each?

As shown in Figure 19.2, the perception module interprets the stimuli coming in from the environment; it produces a perfect three-dimensional model of the world that is transmitted to the reasoning module, which has goals either internally generated or from outside. Combining the model and the goals, it produces a plan. Again, that plan is just a sequence of the form: Do this, do this, do this, then stop. There are no conditionals, no loops in these straight-line plans. Those actions will, when executed, change the world perfectly according to the goals of the robot. Now, unfortunately for the early hopes for this paradigm, this

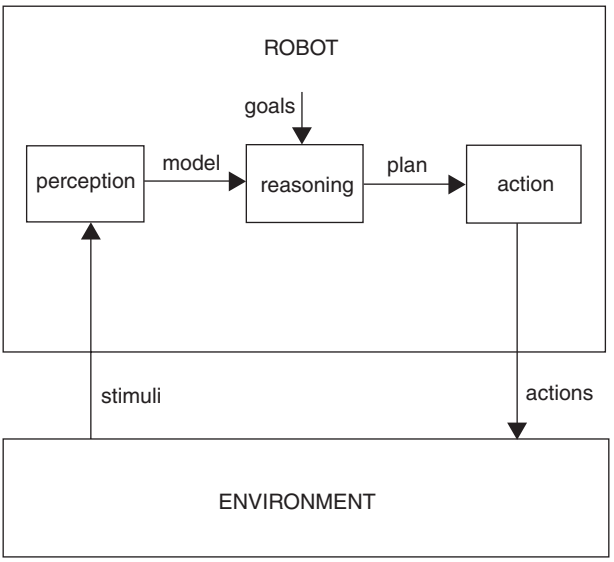

Figure 19.2. A Horizontal Architecture for a GOFAIR Robot.

architecture can only be thought of as a really good first cut. You know that if you wanted to build a robot, it is a really good first thought. You want to push it as hard as you can, because it is nice and simple, it keeps it clean and modular, and all the rest of it. It is simple but, unfortunately, not adequate. Dissatisfaction with this approach drove the next stage of evolution of our views of robotic agents.

The Demise of GOFAIR

GOFAIR robots succeed in controlled environments such as block worlds and factories, but they cannot play soccer! GOFAIR does work as long as the blocks are matte blocks with very sharp edges on black velvet backgrounds. It works in factories if there is only one robot arm and it knows exactly where things are and exactly where they are going to go. The major defect, from my point of view, is that they certainly cannot, and certainly never will, play soccer. I would not let them into my home without adult supervision. In fact, I would advise you not to let them into your home, either.

It turns out that John Lennon, in retrospect, was a great AI researcher: In one of his songs he mused, "Life is what happens to you when you're busy making other plans" (Lennon 1981). The key to the initial success of GOFAIR is that the field attacked the planning problem and came up with really powerful ideas, such as GPS, STRIPS, and back-chaining. This was revolutionary. Algorithms were now available that could make plans in a way we could not do before. The book

Plans and the Structure of Behaviour (Miller 1960) was a great inspiration and motivation for this work. In psychology there were few ideas about how planning could be done until AI showed the way. The GOFAIR paradigm demonstrated how to build proactive agents for the very first time.

Yet planning alone does not go nearly far enough. Clearly, a proactive GOFAIR robot is indeed an agent that can construct plans and act in the world to achieve its goals, whether short term or long term. Those goals may be prioritized. However, "There are more things in heaven and earth, Horatio, than are dreamt of in your philosophy." In other words, events will occur in the world that an agent does not expect. It has to be able to react quickly to interrupts from the environment, to real-time changes, to imminent threats to safety of itself or humans, to other agents, and so on. An intelligent robot must be both proactive *and* responsive. An agent is proactive if it acts to construct and execute short-term and long-term plans and achieve goals in priority order. An agent is responsive if it reacts in real-time to changes in the environment, threats to safety, and to other agents' actions.

Beyond GOFAIR to Soccer

So that was the real challenge to the GOFAIR cartoon worldview that was before us in the 1980s. How could we integrate proactivity and reactivity? In 1992, I made the proposal (Mackworth 1993) that it is fine to say robots must be proactive and reactive (or responsive), but we needed a simple task domain in order to force us to deal with those kinds of issues. I proposed robot soccer as that domain in that paper. Actually, I proposed it after we had actually already built the world's first robot soccer players using cheap toy radio-controlled monster trucks and made them work in our lab. The first two players were named after Zeno and Heraclitus. You can see videos of the first robot soccer games on the Web.[1]

A single color camera looking down on these trucks could see the colored circles on top of the trucks so that the perceptual system could distinguish Zeno from Heraclitus. It could also see the ball and the goals. Each truck has its own controller. Because they cannot turn in place – they are nonholonomic – it is actually a very tricky problem to control this kind of steerable robot. The path planning problems have to be solved in real time. Of course, one is trying to solve a path planning problem as the ball is moving and the opponent is moving in order to get that ball; that is very tricky computationally. We were pushing the limits both of our signal processing hardware and the CPUs in order to get this to work in real time: We were running at about 15Hz cycle time. The other problem was that our lab was not big enough for these monster trucks. So we were forced to go to smaller robots, namely 1/24th scale radio-controlled model Porsches, which we called Dynamites. These cars ran on a ping-pong table with a

[1] URI: http://www.cs.ubc.ca/~mack/RobotSoccer.htm

little squash ball. In the video online, one can see the players alternating between offensive and defensive behaviors. The behaviors the robots exhibit are clearly a mix of proactive and responsive behaviors, demonstrating the evolution of our models of agents beyond the GOFAIR approach.

Incidentally, there was the amazing and successful contemporaneous effort to get chess programs to the point where they could beat the world champion (Hsu 2002). However, from the perspective presented here, it changes only the single agent Sudoku puzzle into a two agent game; all the other aspects of the Sudoku domain remain the same – perfect information, determinism, and the like. Chess loses its appeal as a domain for driving AI research in new directions.

We managed to push all our soccer system hardware to the limit so that we were able to develop two-on-two soccer. The cars were moving at up to 1 m/s and autonomously controlled at 30 Hz. Each had a separate controller off board and they were entirely independent. The only thing they shared is a common front-end vision perceptual module. We were using transputers (a 1MIP CPU) because we needed significant parallelism here. You can see a typical game segment with the small cars on the Web.[2] We were able to do the real-time path planning and correction and control at about 15–30Hz, depending, but that was really the limit of where we could go at that time (1992–4) because we were limited by the hardware constraints.

RoboCup

As happens, shortly thereafter some Japanese researchers started to think along similar lines. They saw our work and said, "Looks good." Instead of using steerable robots, as we had, they chose holonomic robots that can spin in place. Hiroaki Kitano and his colleagues in Japan proposed RoboCup (Kitano 1997). In Korea, the MiroSot group[3] was also intrigued by similar issues. It made for an interesting international challenge.

The first RoboCup tournament was held in Nagoya in 1997. Our University of British Columbia (UBC) team participated; it was a great milestone event. Many researchers have subsequently made very distinguished contributions in the robot soccer area, including Peter Stone, Manuela Veloso, Tucker Balch, Michael Bowling and Milind Tambe, and many others. It has been fantastic. At RoboCup 2007 in Atlanta, there were approximately 2,700 participant agents, and of those about 1,700 were people and 1,000 were robots. A review of the first ten years of RoboCup has recently appeared (Visser and Burckhard 2007), showing how it has grown in popularity and influenced basic research.

It has become incredibly exciting – a little cutthroat and competitive, with perhaps some dubious tactics at times, but that is the nature of intense

[2] URI: http://www.cs.ubc.ca/~mack/RobotSoccer.htm
[3] URI: http://www.fira.net/soccer/mirosot/overview.html

Figure 19.3. Humanoid Robot Soccer Player.

competition in war and soccer. More importantly, robot soccer has been incredibly stimulating to many young researchers, and it has brought many people into the field to do fine work, including new competitions such as RoboRescue and RoboCup@Home. The RoboCup mission is to field a team of humanoid robots to challenge and beat the human champions by 2050, as suggested by Figure 19.3.

From Sudoku to Soccer and Beyond

Now let us step back a bit and consider our theme of evolutionary development of robot architectures. If one thinks of the Sudoku puzzle domain as the exemplar of GOFAIR in a very simple-minded way, then soccer is an exemplar of something else. What is that something else? I think it is situated agents, and so we are transitioning from one paradigm to another. As shown in Figure 19.4, we can compare Sudoku and Soccer as exemplar tasks for each paradigm, GOFAIR and Situated Agents respectively, along various dimensions.

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

	Sudoku	Soccer
Number of agents	1	23
Competition	No	Yes
Collaboration	No	Yes
Real time	No	Yes
Dynamics	Minimal	Yes
Chance	No	Yes
Online	No	Yes
Planning Horizons	No	Yes
Situated Perception	No	Yes
Partially Observable	No	Yes
Open World	No	Yes
Learning	Some	Yes

Figure 19.4. Comparison of Sudoku and Soccer along Various Dimensions.

I shall not go through these dimensions exhaustively. In soccer we have twenty-three agents: twenty-two players and a referee. Soccer is hugely competitive between the teams obviously, but also of major importance is the collaboration within the teams, the teamwork being developed, the development of plays, and the communications systems, signaling systems between players, and the protocols for them. Soccer is real-time. There is a major influence of dynamics and of chance. Soccer is online in the sense that one cannot compute a plan offline and then execute, as one can in GOFAIR. Whenever anything is done, a plan almost always must be recomputed. There exists a variety of temporal planning horizons, from "Can I get my foot to the ball?" through to "Can I get the ball into the net?" and "Can I win this tournament?" The visual perception is very situated and embodied. Vision is now onboard the robots in most of the leagues, so a robot sees only what is visible from where it is, meaning the world is obviously only partially observable. The knowledge base is completely open because one cannot infer much about what is going on behind one's back. The opportunities for robot learning are tremendous.

From GOFAIR to Situated Agents

How do we make this transition from GOFAIR to situated agents? There has been a whole community working on situated agents, building governors for steam

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

engines and the like, since the late nineteenth century. Looking at Maxwell's classic paper, "On Governors" (Maxwell 1868), it is clear that he produced the first theory of control, trying as he was to understand why Watt's feedback controller for steam engines actually worked, under what conditions it was stable, and so on. Control theorists have had a great deal to say about situated agents for the last century or so. Thus, one way to build a situated agent would be to suggest that we put AI and control together: Stick an AI planner, GOFAIR or not, on top of a reactive control-theoretic controller doing proportional-integral-derivative (PID) control. One could also put in a middle layer of finite state mode control. These are techniques we fully understand, and that is, in fact, how we did it for the first soccer players that I described earlier. There was a two-level controller. However, there are many problems with this approach, not the least being debugging it and understanding it, let alone proving anything about it. It was all very much "try it and see." It was very unstable as new behaviors were added: It had to be restructured at the higher level and so on. Let me just say that it was a very graduate-student-intensive process requiring endless student programming hours! So rather than gluing a GOFAIR planner on top of a multilayer control-theoretic controller, we moved in a different direction.

I argued that we must abandon the meta-assumptions of GOFAIR but keep the central metaphor of *constraint satisfaction*. My response was that we just give up on those meta-assumptions of GOFAIR, but not throw out the baby of constraint satisfaction with the bathwater of the rest of GOFAIR. Constraint satisfaction was, and is, the key in my mind, because we understand symbolic constraints as well as numerical. We understand how to manipulate them. We understand even first-order logic as a constraint solving system, thanks to work on that side, but we also understand constraints in the control world. We understand that a thermostat is trying to solve a constraint. We have now a uniform language of constraint solving or satisfaction, although one aspect may be continuous whereas the other may be discrete or even symbolic. There is a single language or single paradigm to understand it from top to bottom, which is what we need to build clean systems. The constraints now though are dynamic: coupling the agent and its environment. They are not like the timeless Sudoku constraint: Every number must be different now and forever. When one is trying to kick a ball. the constraint one is trying to solve is whether the foot position is equal to the ball's position at a certain orientation, at a certain velocity, and so on. Those are the constraints one is trying to solve, and one really does not care how one arrives there. One simply knows that a certain point in time, the ball will be at the tip of the foot, not where it is now, but where it will be in the future. So this is a constraint, but it is embedded in time and it is changing over time as one is trying to solve it, and clearly, that is the tricky part.

Thus, constraints are the key to a uniform architecture, and so we need a new theory of constraint-based agents. This has set the stage. I shall leave you in

suspense for a while for a digression before I come back to sketch that theory. Its development is part of the evolutionary process that is the theme of this article.

Robot Friends and Foes

I digress here briefly to consider the social role of robots. Robots are powerful symbols; they have a very interesting emotional impact. One sees this instinctively if one has ever worked with kids and Lego robotics or the Aibo dogs that we see in Figure 19.5, or with seniors who treat robots as friends and partners. We anthropomorphize our technological things that look almost like us or like our pets – although not too much like us; that is the "uncanny valley" (Mori 1982). We relate to humanoid robots very closely emotionally. Children watching and playing with robot dogs appear to bond with them at an emotional level.

But, of course, the flip side is the robot soldier (Figure 19.6), the robot army, and the robot tank.

Robots, Telerobots, Androids, and Cyborgs

Robots really are extensions of us. Of course, there are many kinds of robots. One uses the word "robot" loosely but, technically, one can distinguish between strictly autonomous robots and telerobots; with the latter, there is human supervisory control, perhaps at a distance, on a Mars mission or in a surgical situation, for example. There are androids that look like us and cyborgs that are partly us and partly machine. The claim is that robots are really reflections of us, and that we project our hopes and fears onto them. That this has been reflected in literature and other media over the last two centuries is a fact. I do not need to bring to mind all the robot movies, but robots do stand as symbols for our technology.

Dr. Frankenstein and his creation, in *Frankenstein; or, The Modern Prometheus* (Shelley 1818), stood as a symbol of our fear, a sort of Faustian fear that that kind of power, that kind of projection of our own abilities in the world, would come back and attack us. Mary Shelley's work explored that, and Charlie Chaplin's *Modern Times* (Chaplin 1936) brought the myth up to date. Recall the scene in which Charlie is being forced to eat in the factory where, as a factory worker, his entire pace of life is dictated by the time control in the factory. He is a slave to his own robots and his lunch break is constrained because the machines need to be tended. He is, in turn, tended by an unthinking robot who keeps shoving food into his mouth and pouring drinks on him until finally, it runs amok. Chaplin was making a very serious point that our technology stands in real danger of alienating and repressing us if we are not careful.

I'll conclude this somewhat philosophical interjection with the observations of two students of technology and human values. Marshall McLuhan argued

Figure 19.5. Robot Friends Playing Soccer.

Figure 19.6. . . . and Robot Foes.

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

(although he was thinking of books, advertising, television, and other issues of his time, though it applies equally to robots), "We first shape the tools and thereafter our tools shape us" (McLuhan 1964). Parenthetically, this effect can be seen as classic projection and alienation in the sense of Feuerbach (Feuerbach 1854).

The kinds of robots we decide to build will change us as they will change our society. We have a heavy responsibility to think about this carefully. Margaret Somerville is an ethicist who argues that the whole species *Homo sapiens* is actually evolving into *Techno sapiens* as we project our abilities out (Somerville 2006). Of course, this is happening at an accelerating rate. Many of our old ethical codes are broken and do not work in this new world, whether it is in biotechnology or robotics, or in almost any other area of technology today. As creators of some of this technology, it is our responsibility to pay serious attention to that problem.

Robots: One More Insult to the Human Ego?

Another way of thinking about our fraught and ambivalent relationship with robots is that this is really one more insult. How much more can humankind take? Robotics is only the latest displacement of the human ego from center stage. Think about the intellectual lineage that links Copernicus, Darwin, Marx, Freud, and Robots. This may be a stretch, but perhaps not.

Humans thought they were at the center of the universe until Copernicus proposed that the earth was not at the center, but rather that the sun was. Darwin hypothesized we are descended from apes. Marx claimed that many of our desires and goals are determined by our socioeconomic status, and, thus, we are not as free as we thought. Freud theorized one's conscious thoughts are not freely chosen, but rather they come from the unconscious mind. Now I suggest that you can think of robots as being in that same great lineage, which states: You, *Homo sapiens*, are not unique. Now there are other entities, created by us, that can also perceive, think, and act. They could become as smart as we are. Yet this kind of projection can lead to a kind of moral panic: "The robots are coming! The robots are coming! What are we going to do?" When we talk to the media the first questions reporters ask are typically: "Are you worried about them rising up and taking over?" and "Do you think they'll keep us as pets?" The public perception of robots is evolving as our models of robots and the robots themselves evolve.

Helpful Robots

To calm this kind of panic we need to point to some helpful robots. The University of Calgary NeuroArm is actually fabricated from nonmagnetic parts so it can operate within an MRI field. It allows a surgeon to do neurosurgery telerobotically, getting exactly the right parts of the tumor while seeing real time feedback as the surgery is performed.

An early prototype of our UBC smart wheelchair work is shown in Figure 19.7. This chair can use vision and other sensors to locate itself, map its environment, and allow its user to navigate safely.

RoboCars: DARPA Urban Challenge

Continuing with the helpful robot theme, consider autonomous cars. The original DARPA Challenges in 2004 and 2005 and the Urban Challenge in 2007 have catalyzed significant progress. Sebastian Thrun and his team at Stanford developed Junior (Figure 19.8[a]), loaded with sensors and actuators and horsepower and CPUs of all sorts, who faced off against Boss (Figure 19.8[b]) and the Carnegie Mellon/General Motors Tartan racing team in the fall of 2007. Boss took first place and Junior took second in the Urban Challenge.[4] The media look at these developments and see them as precursors to robot tanks, cargo movers, and automated warfare, naturally because they know that DARPA funded them. However, Thrun (Thrun 2006) is an evangelist for a different view of such contests. The positive impact of having intelligent cars would be enormous. Consider the potential ecological savings of using highways much more efficiently instead of paving over farmland. Consider the safety aspect, which could reduce the annual carnage of 4,000 road accident deaths a year in Canada alone. Consider the fact that cars could negotiate at intersections: Dresner and Stone (Dresner 2008) have simulated to show you could get potentially two to three times the throughput in cities in terms of traffic if these cars could talk to each other instead of having to wait for stop signs and traffic lights. Consider the ability of the elderly or disabled to get around on their own. Consider the ability to send one's car to the parking lot by itself and then call it back later. There would be automated warehouses for cars instead of using all that surface land for parking. Truly, the strong positive implications of success in this area are enormous. Yet can we trust them? This is a real problem and major problem. In terms of smart wheelchairs, one major reason why they do not already exist now is liability. It is almost impossible to get an insurance company to back a project or a product. This clarifies why the car manufacturers have moved very slowly and in an incremental way to develop intelligent technology.

Can We Trust Robots?

There are some real reasons why we cannot yet trust robots. The way we build them now, not only are they not trustworthy, they are also unreliable. So can they do the right thing? Will they do the right thing? Then, of course, there is the fear that I alluded to earlier – that eventually they will become autonomous, with free will, intelligence, and consciousness.

[4] URIs: http://www.tartanracing.org, http://cs.stanford.edu/group/roadrunner

Figure 19.7. Prototype Smart Wheelchair (UBC, 2006).

(a) "Junior"
(Stanford Racing Team, 2007)

(b) "Boss"
(CMU-GM Tartan Racing Team, 2007)

Figure 19.8. Two competitors in the DARPA Urban Challenge.

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

Ethics at the Robot/Human Interface

Do we need robot ethics, for us and for them? We do. Many researchers are working on this (Anderson and Anderson 2007). Indeed, many countries have suddenly realized this is an important issue. There will have to be robot law. There are already robot liability issues. There will have to be professional ethics for robot designers and engineers just as there are for engineers in all other disciplines. We will have to factor the issues around what we should do ethically in designing, building, and deploying robots. How should robots make decisions as they develop more autonomy? How should we behave and what ethical issues arise for us as we interact with robots? Should we give them any rights? We have a human rights code; will there be a robot rights code?

There are, then, three fundamental questions we have to address:

1. What should we humans do ethically in designing, building, and deploying robots?
2. How should robots decide, as they develop autonomy and free will, what to do ethically?
3. What ethical issues arise for us as we interact with robots?

Asimov's Laws of Robotics

In considering these questions we will go back to Asimov (Asimov 1950) as he was one of the earlier thinkers about these issues; he put forward some interesting, if perhaps naïve, proposals. His original three Laws of Robotics are:

1. A robot may not harm a human being, or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence, as long as such protection does not conflict with the First or Second Laws.

Asimov's Answers

Asimov's answers to those questions I posed are: First, by law, manufacturers would have to put those laws into every robot. Second, robots should always have to follow the prioritized laws. He did not say much about the third question. His plots arise mainly from the conflict between what the humans intend the robot to do and what it actually does do, or between literal and sensible interpretations of the laws stemming from the lack of codified formal language. He discovered many hidden contradictions but they are not of great interest here. What is of interest and important here is that, frankly, the laws and the assumptions behind them are naïve. That is not to blame Asimov – he pioneered the area – but we can

say that much of the ethical discussion nowadays remains naïve. It presupposes technical abilities that we just do not have yet.

What We Need

We do not currently have adequate methods for modeling robot structure and functionality, of predicting the consequences of robot commands and actions, and of imposing requirements on those actions such as reaching the goal but doing it in a safe way and making sure that the robot is always live, with no deadlock or livelock. Most important, one can put those requirements on the robot, but one has to be able to find out if the robot will be able to satisfy those requirements. We will never have 100 percent guarantees, but we do need within-epsilon guarantees. Any well-founded ethical discussion presupposes that we (and robots) do indeed have such methods. That is what we require.

Theory Wanted

So, finally coming back to the constraint-based agent theory, it should help to satisfy those requirements. In short, we need a theory with a language to express robot structure and dynamics, a language for constraint-based specifications, and a verification method to determine if a robot described in the first language will (be likely to) satisfy its specifications described in the second language.

Robots as Situated Agents

What kind of robots, then, are we thinking about? These are *situated* robots tightly coupled to the environment; they are not universal robots. Remember *Rossum's Universal Robots* (Capek 1923)? We are not going to build universal robots. We are building very situated robots that function in particular environments for particular tasks. However, those environments are typically highly dynamic. There are other agents. We have to consider social roles. There is a very tight coupling of perception and action, perhaps at many different levels. We now know that the human perceptual system is not a monolithic black box that delivers a three-dimensional model from retinal images. There are many visual subsystems dealing with recognition, location, orientation, attention, and so forth. Our robots will be like that as well.

It is not "cheating" to embody environmental constraints by design, evolution, or learning. It was cheating in the old GOFAIR paradigm that did aim at universal robots. Everything had to be described in, say, the logic, and one could not design environmental constraints into the robots. We think just following biology and natural evolution is the way to go, and learning will play a major part. Evolution is learning at the species level. Communication and perception are very situated. The architectures are online, and there is a hierarchy of time

scales and time horizons. Critically, we want to be able to reason about the agent's correctness. We do not require the agents to do reasoning – they may not – but certainly we want to be able to reason about them. When we think back to the GOFAIR model, we never actually did that. The reasoning was in the agent's head alone, and we assumed that if it was correct, everything else was correct. Finally, as I mentioned earlier, one cannot just graft a symbolic system on top of a signal-control-based system and expect the interface to be clean, robust, reliable, debuggable, and (probably) correct. So the slogan is "No hybrid models for hybrid systems."

Vertical Architecture

To satisfy those requirements for situated agents, we have to throw away the horizontal three boxes architectural model and move to a vertical "wedding cake" architecture. As shown in Figure 19.9, as one goes up these controllers, each controller sees a virtual body below it, modularizing the system in that way. Each controller, as one goes higher, is dealing with longer time horizons but with coarser time granularity and different kinds of perception. Each controller will only know what it needs to know. This architectural approach was advocated by Albus (Albus 1981) and Brooks (Brooks 1986). It corresponds quite closely to biological systems at this level of abstraction.

A Constraint-Based Agent

We are interested in constraint-based agents. They are situated; they will be doing constraint satisfaction but in a more generalized sense, not in the GOFCS sense. These constraints may be prioritized. Now we conceive of the controller of the agent or robot as a *constraint solver*.

Dynamic Constraint Satisfaction

Consider the generalization of constraint satisfaction to *dynamic constraint satisfaction*. A soccer example will serve us.

Imagine a humanoid robot trying to kick a soccer ball. In Figure 19.10, we can see the projection into a two-dimensional space of a complex phase space that describes the position and velocity of the limbs of the robot and the ball at time t. Each flow line in the figure shows the dynamics of the evolution of the system from different initial conditions. The controller has to be able to predict where the robot should move its foot to, knowing what it knows about the leg actuators, the ball and where it is moving, how wet the field is, and so on to make contact with the ball to propel it in the right direction. That corresponds to the 45° line $y = x$. So x here is the ball position on the horizontal axis, and y is the foot position on the vertical axis. That is the constraint we are trying to solve. If

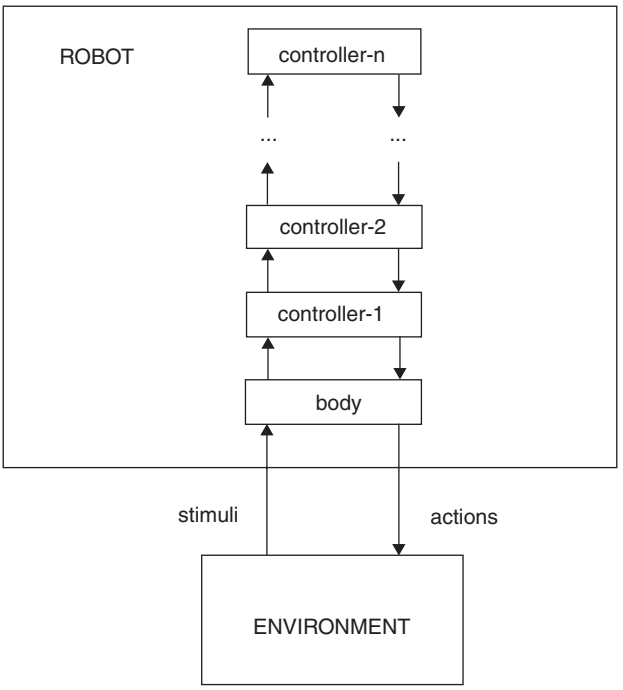

Figure 19.9. A Vertical Robotic System Architecture.

the controller ensures the dynamical system always goes to (or approaches, in the limit) that constraint and stays there, or maybe if it doesn't stay there, but it always returns to it soon enough, then we say that this system is solving that constraint, $FootPosition(t) = BallPosition(t)$. In hybrid dynamical systems language, we say the coupled agent environment system *satisfies the constraint* if and only if the constraint solution set, in the phase space of that coupled hybrid dynamical system, is an *attractor* of the system as it evolves. Incidentally, that concept of online hybrid dynamical constraint satisfaction subsumes the entire old discrete offline GOFCS paradigm (Zhang and Mackworth 1993).

Formal Methods for Constraint-Based Agents

The Constraint-Based Agent (CBA) framework consists of three components:

1. Constraint Net (CN) for system modeling
2. Timed for-all automata for behavior specification
3. Model checking and Liapunov methods for behavior verification

These three components correspond to the tripartite requirement for the theory we said we wanted earlier. Ying Zhang and I developed these formal methods

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

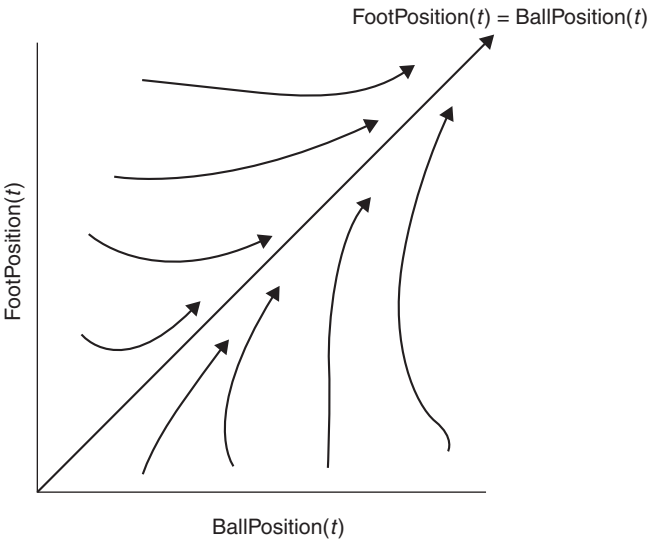

Figure 19.10. Solving a Dynamic Soccer Constraint.

for constraint-based agents (Zhang and Mackworth 1995; Mackworth and Zhang 2003). First, there is an architecture for distributed asynchronous programming languages called Constraint Nets (CN). Programs in CN represent the robot body, the controller, and the environment. In them are represented constraints that are local on the structure and dynamics of each system. For behavior specification we either use temporal logics or timed for-all automata. For verification techniques we have used model checking or generalized Liapunov techniques taken from the standard control literature but generalized for symbolic as well as numerical techniques. Rather than present any technical detail here, I shall sketch a case study, again using soccer.

A Soccer Case Study with Prioritized Constraints

Suppose we want to build a robot soccer player that can move around the world and repeatedly find, track, chase, and kick the soccer ball. The setup is shown in Figure 19.11. Pinar Muyan-Özçelik built a controller for this robot to carry out the task using the Constraint-Based Agent methodology (Muyan-Özçelik and Mackworth 2004). The detailed view of the robot in Figure 19.12 shows a robot base that can only move in the direction it is facing, but it can rotate in place to move in a new direction. There is a pan-tilt unit that serves as a neck and a trinocular color camera on top of it that can do stereo vision, but in this experiment we used monocular color images only.

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

Figure 19.11. A Robot and a Human Kick the Ball Around.

This is a very simple, almost trivial example, but even here you get a rich complexity of interaction with emergent behavior. Imagine that you have got very simple controllers that can solve each of these constraints: (1) get the ball in the image; (2) if the ball is in the image, center it; and (3) make the base heading equal to the pan direction. Imagine that you are a robot and you can only move forward in the direction you are facing with these robots. If you turn your head to the left and acquire the ball in the image over there, then you have to turn your body to the left toward it, and as you are tracking the ball in the image you have to turn your head to the right in the opposite direction. This is analogous to the well-known vestibulocular reflex (VOR) in humans. Now you are looking at the ball and facing toward it, so now you can move toward it and hit the ball. The last constraint is for the robot to be at the ball. If we can satisfy these constraints, in the correct priority order, this unified behavior will emerge: acquire, track, chase, and kick the ball. If at any time one is satisfying a lower priority constraint and a higher priority constraint becomes unsatisfied, the controller must revert to resatisfying it.

The prioritized constraints are: Ball-In-Image (I), Ball-In-Center (C), Base-Heading-Pan (H), Robot-At-Ball (A). The priority ordering is: I > C > H > A. We want to satisfy those prioritized constraints. The specification for this system is that one has to solve those four constraints with that priority. That is all one would say to the system. It is a declarative representation of the behavior we want the system to exhibit. We can automatically compile that specification into a controller that will in fact exhibit that emergent behavior. We can conceptualize these prioritized constraint specifications as generalizations of the GOFAIR linear sequence plans.

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

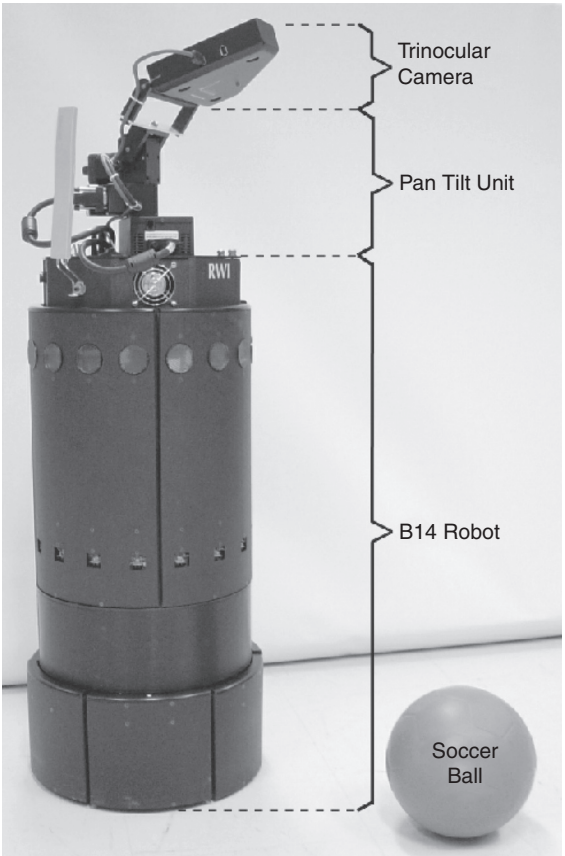

Figure 19.12. A Simple Soccer Player.

Constraint-Based Agents in Constraint Nets

Suppose we are given a prioritized constraint specification for a controller at a certain level in the controller hierarchy as in Figure 19.13. The specification involves Constraint1, Constraint2, and Constraint3. It requires this priority order: Constraint1 > Constraint2 > Constraint3.

We assume we have a simple solver for each constraint, Constraint Solver-1, -2, and -3. Constraint1 is the highest priority, so if it is active and not satisfied, its solver indicates, "I'm not satisfied now, I'd like you to do this to satisfy Constraint1." It might be a gradient descent solver, say. Its signal would go through Arbiter-1. The arbiter knows this is higher priority, and its signal passes it all the way through to Arbiter-2 as well, to the motor outputs. If Constraint-1 is satisfied, Arbiter-1 will let ConstraintSolver-2 pass its outputs through and so

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

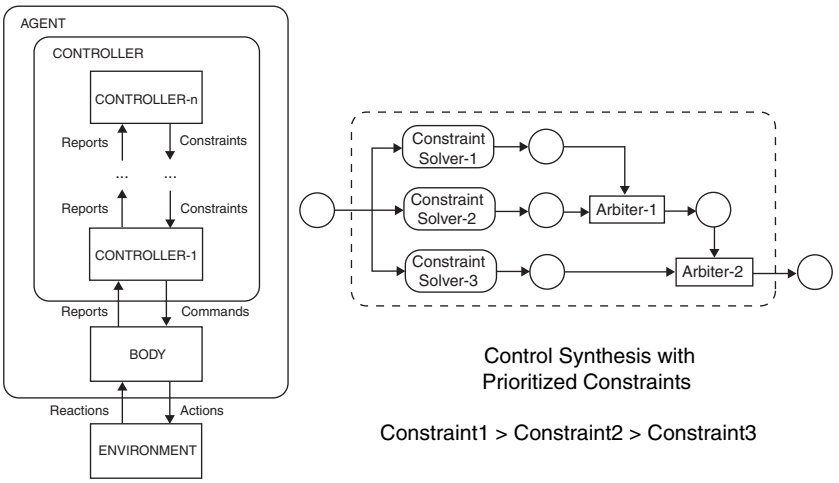

Figure 19.13. Synthesizing a Controller from a Prioritized Constraint Specification.

on. If there is any conflict for the motors, that is how it is resolved. If there is no conflict, then the constraints can be solved independently because they are operating in orthogonal spaces. Using that architecture we built a controller for those constraints in the soccer player, and we tested it all in simulation and it works. It works in a wide variety of simulated conditions; it works with the same controller in a wide variety of real testing situations. We found that the robot always eventually kicks the ball repeatedly, both in simulation and experimentally. In certain circumstances, we can *prove* that the robot always eventually kicks the ball repeatedly. We conclude that the Constraint-Based Agent approach with prioritized constraints is an effective framework for robot-controller construction for a simple task.

Just as an aside, this is a useful way to think about a classic problem in psychology. Carl Lashley, in a seminal paper called "The Problem of the Serial Ordering of Behavior" (Lashley 1951), was writing about speech production, but the sequencing problem arises for all behaviors. Suppose one has a large number of goals one is attempting to satisfy. Each goal is clamoring to be satisfied. How do they get control of the actuators? How does one sequence that access? How does one make sure it is robust? For prioritized constraints, it is robust in the sense that if the controller senses the loss of satisfaction of a higher-priority constraint, it will immediately resume work on resatisfying it. One can also think of what we have done as a formalization of subsumption (Brooks 1991). So this is *a* solution; I am not saying it is *the* solution. It is a very simple-minded solution, but it is a solution to the classic problem of the serial ordering of behavior. It demonstrates that prioritized constraints can be used to build more reliable dynamic agents.

https://doi.org/10.1017/CBO9780511978036.024 Published online by Cambridge University Press

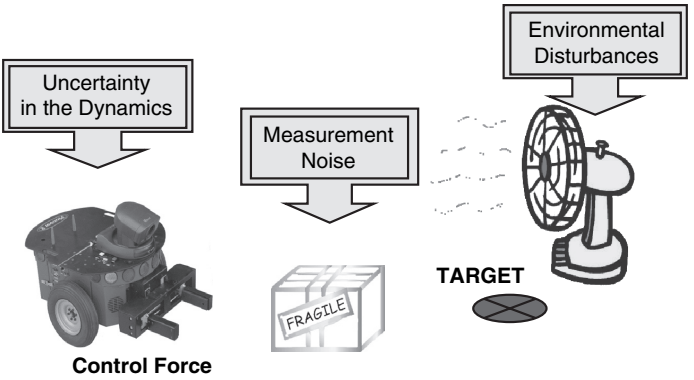

Figure 19.14. Uncertainty in Robotic Systems.

Modeling Uncertainty

So far, nothing has been said about the element of chance, but of course, in real robots with real environments there will be much noise and uncertainty. Robert St-Aubin and I have developed Probabilistic Constraint Nets using probabilistic verification (St-Aubin 2006). As shown in Figure 19.14, there will be uncertainty in the model of the dynamics, in the dynamics themselves, and in the robot's measurement of the world. Moreover, one will be unable to fully model the environment so there will be environmental disturbances.

Observations and Conclusion

Stepping back we observe that a very simple idea, constraint satisfaction, allows us to achieve intelligence through the integration of proactive and responsive behaviors; it is uniform top to bottom. We see in the formal prioritized constraint framework the emergence of robust goal-seeking behavior. I propose it as a contribution to the solution of the problem of a lack of technical foundation to many of the naïve proposals for robot ethics. So if one asks, "Can robots do the right thing?" the answer so far is "Yes, sometimes they can do the right thing, almost always, and we can prove it, sometimes."

Acknowledgments

I am most grateful to all of the students, colleagues, and collaborators who have contributed to some of the work mentioned here: Rod Barman, Le Chang, Pooyan Fazli, Gene Freuder, Joel Friedman, Stewart Kingdon, Jim Little, David Lowe, Valerie McRae, Jefferson Montgomery, Pinar Muyan-Özçelik, Dinesh Pai, David Poole, Fengguang Song, Michael Sahota, Robert St-Aubin, Pooja

Viswanathan, Bob Woodham, Suling Yang, Ying Zhang, and Yu Zhang. This chapter is based, in large part, on the article documenting my AAAI presidential address (Mackworth 2009); David Leake and Mike Hamilton helped with that article. Funding was provided by the Natural Sciences and Engineering Research Council of Canada and through the support of the Canada Research Chair in Artificial Intelligence.

References

Albus, J. S. 1981. *Brains, Behavior and Robotics.* NY: McGraw-Hill.

Anderson, M. and Leigh Anderson, S. 2007. Machine Ethics: Creating an Ethical Intelligent Agent. *AI Magazine*, 28(4):15–26.

Asimov, I. 1950. *I, Robot*. NY: Gnome Press.

Brooks, R. A. 1986. A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, RA-2(1): 14–23.

Brooks, R. A. 1991. Intelligence without Reason. In Proc. of Twelth International Joint Conference on Artificial Intelligence, 569–595. San Mateo, CA: Morgan Kaufmann.

Capek, K. 1923. *R.U.R. (Rossum's Universal Robots): A Fantastic Melodrama in Three Acts and an Epilogue*. Garden City, NY: Doubleday.

Dresner, K. and Stone, P. 2008. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research* 31:591–656.

Feuerbach, L. A. 1854. *The Essence of Christianity*. London: John Chapman.

Freuder, E. C. and Mackworth, A. K. 2006. Constraint Satisfaction: An Emerging Paradigm. In *Handbook of Constraint Programming*, ed. F. Rossi, P. Van Beek and T. Walsh, 13–28. Amsterdam: Elsevier.

Haugeland, J. 1985. *Artificial Intelligence: The Very Idea*. Cambridge, MA: MIT Press.

Hsu, F. 2002. *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*, Princeton, NJ: Princeton University Press.

Kitano, H . (ed.) 1998. *RoboCup-97: Robot Soccer World Cup I*. Lecture Notes in Computer Science 1395, Heidelberg: Springer.

Lashley, K.S., 1951. The Problem of Serial Order in Behavior. In *Cerebral Mechanisms in Behavior*. Ed. L.A. Jeffress, 112–136. New York: Wiley.

Lennon, J. 1980. Beautiful Boy (Darling Boy). Song lyrics. On album *Double Fantasy*.

McLuhan, M. 1964. *Understanding Media: The Extensions of Man*. New York: New American Library.

Mackworth, A. K. 1977. Consistency in Networks of Relations, *Artificial Intelligence* 8(1), 99–118.

Mackworth, A. K. 1993. On Seeing Robots. In *Computer Vision: Systems, Theory and Applications* eds. A. Basu and X. Li, 1–13. Singapore: World Scientific Press.

Mackworth, A. K. 2009. Agents, Bodies, Constraints, Dynamics, and Evolution. *AI Magazine*, 26(30):7–28, Spring 2009.

Mackworth, A. K. and Zhang, Y. 2003. A Formal Approach to Agent Design: An Overview of Constraint-Based Agents. *Constraints* 8 (3) 229–242.

Maxwell J. C. 1868. On Governors. In *Proceedings of the Royal Society of London*, 16, 270–283. London: The Royal Society.

Miller, G. A., Galantner, E., & Pribram, K. H. 1960. Plans and the Structure of Behavior. New York: Holt, Rinehart & Winston.

Mori, M. 1982. *The Buddha in the Robot*. Tokyo: Charles E. Tuttle Co.

Muyan-Özçelik, P., and Mackworth, A. K. 2004. Situated Robot Design with Prioritized Constraints. In *Proc. Int. Conf. on Intelligent Robots and Systems* (IROS 2004), 1807–1814.

Reiter, R. 1978. On Closed World Data Bases. In *Logic and Data Bases* eds. H. Gallaire and J. Minker, 119–140. New York, NY: Plenum.

Rossi, F ., van Beek, P . **and** Walsh, T . (eds.) 2006. *Handbook of Constraint Programming.* Amsterdam: Elsevier Science.

Sahota, M. and Mackworth, A. K. 1994. Can Situated Robots Play Soccer? In *Proc. Artificial Intelligence '94*, 249–254. Toronto ON: Can. Soc. for Comp. Studies of Intelligence.

Shelley, M. W. 1818. *Frankenstein; or, The Modern Prometheus.* London: Lackington, Hughes, Harding, Mavor and Jones.

Somerville. M. 2006. *The Ethical Imagination: Journeys of the Human Spirit*. Toronto: House of Anansi Press.

St-Aubin, R., Friedman, J. and Mackworth, A. K. 2006. A Formal Mathematical Framework for Modeling Probabilistic Hybrid Systems. *Annals of Mathematics and Artificial Intelligence* 37(3–4) 397–425.

Thrun, S. 2006. Winning the DARPA Grand Challenge. Invited Talk at Innovative Applications of Artificial Intelligence (IAAI-06), Boston, Massachusetts, July 16–20.

Visser, U. and Burkhard, H. D. 2007. RoboCup: 10 years of Achievements and Challenges. *AI Magazine* 28(2) 115–130.

Waltz, D. L. 1975. Understanding Line Drawings of Scenes with Shadows. In *The Psychology of Computer Vision*, ed. P.H. Winston, 19–92. New York, NY: McGraw-Hill.

Zhang, Y. and Mackworth, A. K. 1993. Constraint Programming in Constraint Nets. In *Proc. First Workshop on Principles and Practice of Constraint Programming*. 303–312. Padua: Assoc. for Constraint Programming.

Zhang, Y. and Mackworth, A. K. 1995. Constraint Nets: A Semantic Model for Dynamic Systems. *Theoretical Computer Science* 138 211–239.