Oral History of Murray Campbell

Interviewed by:
Dag Spicer

Recorded: September 8, 2005
Mountain View, California

CHM Reference number: X3256.2006

Dag Spicer: I'm Dag Spicer, the Senior Curator at the Computer History Museum, and I'm here today with Murray Campbell. This is September 8, 2005. Murray Campbell was one of the key investigators on the Deep Blue Project, a supercomputer that defeated the world chess champion in 1997, and he has had a long career in computer chess beginning in the 1980s. Welcome, Murray.

Murray Campbell: Thank you.

Spicer: We're delighted to have you here--

Campbell: Glad to be here.

Spicer: I wanted to start fairly early in your life, before you worked on Deep Blue, because I know you're one of the few team members on Deep Blue that was actually a really good chess player, and that came at a fairly early age. I was wondering if you could talk to us a little bit about when you started learning chess, enjoying chess and maybe realizing that you actually had some talent in chess.

Campbell: I guess started playing chess at maybe 11, 12 years old and played while I was a teenager. I generally did pretty well in expert level, which is short of master but was respectable. I managed to win the provincial championship of Alberta once, and the junior championship a couple of times to represent Alberta in the national championship. During those teenage years I had my first chess computer. It was a Fidelity Chess Challenger, a little hand held device, and I remember being very frustrated at how terrible it was. I would let it search on its deepest search mode, which would run overnight, and I would play a game against it. It took months to play a game and it was still pitiful, but even at that stage obviously I was interested.

Spicer: What year would that have been, approximately?

Campbell: Hard to say exactly. I could go back and look when that thing--

Spicer: --the '70s or something?

Campbell: I would say probably mid to late '70s.

Spicer: So the state of knowledge, if you like, built into chess machines at that time -- the little portable chess boards -- was not very good.

Campbell: It was quite poor, yes. Frustratingly poor.

Spicer: Right. For anyone who is a decent chess player.

Campbell: Yes.

Spicer: I read that you actually saw a computer playing chess, I think, at the University of Alberta?

Campbell: Yeah. There was an open house at the Computing Science department -- this was while I was a senior in high school -- and I happened to see a program called Tinkerbell, which was written by Ken Thompson in fact. It was running on an old PDP-11 system, and just the idea of it, being a chess player myself, seeing that computer play chess, was practically the thing that convinced me computer science was the place for me to go as opposed to physics or math or something else.

Spicer: Wow. Were you interested in computer science itself, or specifically in computer chess?

Murray Campbell: Well, computer science in general. But that was enough to tip the balance to make computer science seem more interesting, more exciting.

Spicer: Did you get to interact with Tinkerbell?

Campbell: Later I did, not at that time. I think I was observing a game, but later on when I was actually a student there I did get to play against it. It was certainly better than the Chess Challenger but still not very good.

Spicer: I think it would take quite a bit of time between moves, I guess, or did it—

Campbell: It was programmable, the amount of time it would take.

Spicer: Getting back to this theme in the development of machines that play chess, there seem to be two kinds of communities. One is the chess players who get into computer science, and then there's this sort of straight computer scientists who may have limited chess knowledge. But somehow they're very complementary, it seems, and a lot of the winning chess machines seem to involve groups from both communities. Do you have any sort of insight into—

Campbell: I would say it's interesting that… As you expect in any field there were going to be top performers. But it seems there's very few examples of really top performers in both fields working together, I guess. Hans Berliner created a strong program and he was a very strong player, but he might have been one of the few exceptions. Mostly they're very good in one area and at least adequate in the other area, which allows them to make some progress.

Spicer: I think Hans Berliner is really one of the key people I can even think of that was really strong, and I think you are probably another one.

Campbell: Hans was certainly a much better chess player than I'll ever be.

Spicer: We did a great oral history with him by the way.

Campbell: Oh, good.

Spicer: You might enjoy watching it. Can you tell us about your academic career a bit, and when you really sunk your teeth into the computer chess problem?

Campbell: As an undergraduate I did all of the typical undergraduate things in computer science: math and physics and computing science courses. But I recall that one of my projects as an undergraduate was we decided to build a chess program, a group of us, I think three of us, and this was probably '78, 1978. It was obviously a very simplistic program that didn't play particularly well, but it was my first attempt at this area. During the summer I got a position working with Tony Marsland, who was to later become my thesis adviser for my master's, doing some work for him. He had a chess program, AWIT, which was one of the better known programs back in the '70s, and I moved on doing my master's thesis under Tony. I did some work on parallel game tree search. Obviously we were thinking ahead, that in the long run there will be large scale parallel systems that play chess and we'll need algorithms for using those systems to efficiently search chess game trees.

Spicer: AWIT, just to take a step back, ran on a mainframe I think, an Amdahl—

Campbell: Yes.

Spicer: --mainframe, 470/V6 or something?

Campbell: Yes.

Spicer: The way that researchers in this area exchanged information? I guess starting in 1970 in the ACM tournaments, but obviously you didn't have the web and the quick collaborative tools that we do now.

Campbell: True.

Spicer: Can you comment on how progress occurred?

Campbell: I think progress was driven very much by these annual competitions. People wanted to put their best foot forward and do well in the annual competition. So there would be a flurry of progress in each individual camp prior to the event, and then everybody would discuss and share ideas. That's primarily when the cross pollination occurred. Then there'd be another year and it would happen, but obviously it's happening at a very slow pace. There were obviously publications as well, but at that point it was a fairly open community, fairly academic. Commercial teams hadn't particularly entered much into the area, and of course the commercial teams never published or rarely published. So when that happened it changed the dynamics a little bit. But at that [earlier] point it was pretty open, but at a glacial pace.

Spicer: This is something that Ken Thompson also lamented, and said initially it was so collegial and friendly and open, and as computer chess became a business that sort of vaporized. There was money at stake.

Campbell: Well, that's true. Each year there would be probably more and more commercial competitors trying to sell their programs, and they don't want to give away their competitive secrets. You lost some of that openness.

Spicer: That's kind of the "tragedy of the commons". I don't know if you know that argument, but it relates back to English enclosure laws, in which there used to be common areas in a town which farmers could all share. What would happen is often, human nature being what it is, two or three farmers would start putting up fences and treating the land as their own personal property rather than something shared in common. Kind of an analogy.

Campbell: A good analogy, yes.

Spicer: Moving from master's at Alberta to CMU. What got you interested in CMU?

Campbell: Well, I was interested in CMU because I had seen some of the publications of Hans Berliner and knew of his work. If I was going to continue working in this area, that would be a natural place to go. I was happy that I was able to go there and have Hans as my adviser and continue working in related areas. My PhD research was in the area of chunking, that is, dividing a problem into sub-problems that interact only slightly, and trying to solve each of the sub-problems, and then putting it together and composing a solution that involved both parts. I chose chess endgames -- in particular chess pawn endgames -- as a good domain for this because pawns don't move sideways generally and so they don't interact with each other. You can easily divide up a chessboard with pawns and kings into separate components, and then the kings are the interaction because they move between components. I got some interesting results and some interesting publications out of that, which showed that although the method was interesting, the analysis of the individual components was a brute force type analysis. It gets very good results when it's within range, but it's not terribly scalable. I think that's part of the limitations of the work -- that it wasn't as scalable to other areas as we had hoped.

Spicer: I see a little thread here in your trajectory with parallelizing the chess problem. Does chunking connect with it, or?

Campbell: I think it's a separate thread. The parallelizing work was more concerned with searching a game tree and figuring out how to make that efficient on a parallel computer. As you search through a game tree in a game like chess, if you applied a naïve method to parallelize it you would lose a lot of efficiency. It turns out if you do it in a clever way, and establish some information about the position first off the bat, and then share that information with all the other processors, they can be much more efficient. So finding good ways to do that was that thread. This thread was more closer to traditional AI, that is, taking a problem, decomposing it into sub–problems, and then reconstructing the solution by handling the interactions. In some sense there could be some parallelism there because the separate problems could be solved separately, but that wasn't the main focus. The main focus was: how do you take these solutions to sub-problems and compose them together.

Spicer: Can you explain for viewers why search trees are sort of "parallel friendly", or friendly to parallel architectures?

Campbell: They're actually not. That's the problem. If they were friendly, and it's called "embarrassingly parallel", you would just farm out the jobs to the processors, they would do their work, they'd send the results back, and you'd be done. There would be no real research there, except trying to get the maximum speed possible on the communications. The problem is that it's not one of these "embarrassingly parallel" algorithms as I mentioned. There was a need to establish some bounds on a position -- where the value of the position is likely to lie -- before you farm it out to all the processors. If you don't take care to do that you lose a huge amount of time. So that's why there's actually interesting research issues in how to farm out these jobs to the processors.

Spicer: Did this inform Deep Blue's—

Campbell: Absolutely. I think we moved a long way past what my master's thesis work was in the late '70s, early '80s But it was critical to Deep Blue to have an efficient way to allocate these jobs to the roughly 500 processors that were part of the system.

Spicer: Let's flip back a little to CMU. Can you tell us about your graduate work there? I think you changed projects once or maybe twice?

Campbell: Not really. I worked with Hans Berliner through my graduate career at CMU. I did the work on the chunking, the dividing, the chess pawn and games work, which was my main focus, and spent most of my time. I also was part of a team that developed a machine called Hitech, which was one of the better machines in the mid to early '80s.

Spicer: It's coming here by the way; CMU is donating it.

Campbell: Great. I'm glad—that's where it belongs.

Spicer: Thank you.

Campbell: I was one of the team. There was a team of four or five of us, led by Hans Berliner and Carl Ebeling, that designed and built this machine and then programmed it. I was one of the team that helped program that system and helped test it and get it to work. Somewhat after the Hitech machine had been built, a group of graduate students initially inspired by Feng-Hsiung Hsu -- he had designed a chip that could search through chess moves, a completely different architecture to what Hitech used, and in my mind a better architecture in some ways. That led to a group of us putting together a system called ChipTest which later led to Deep Thought, and the first defeat of a grand master in computer play by that machine, Deep Thought.

Spicer: We're going to explore that a little more. It isn't that you changed projects. I'm sorry.

Campbell: I had one main project and then a couple of side projects.

Spicer: That's quite the workload!

Campbell: Well, when you're a graduate student you have lots of time and energy.

Spicer: Can you evaluate the two machines that grew out of the department? The interactions maybe between the two teams, or the two architectures?

Campbell: Hitech came first and had some good initial results. It competed locally in chess tournaments -- in the Pittsburgh Chess Club and in Pennsylvania tournaments -- as well as in the ACM world computer chess championship computer tournaments, and did quite well. It was, I guess, the first senior master level program, rating above 2400 on the U.S. Chess Federation scale. Almost a competitive program was developed by this other group of students. There was no actual faculty member involved in that network, it was just students, and it was more or less scrounging up parts from around the computer science department to put together, borrowing a work station and putting it all together. There were some shared technologies between them, and some of the searching enhancements that we had come up with in Deep Thought or ChipTest made their way into Hitech Certainly since I was on both teams -- I guess I was the only person that was on both teams -- there was some transfer that way through me into evaluation and into other ideas. But they were relatively independent projects at that point.

Spicer: I don’t want to open any old wounds, but there was some tension -- a little bit maybe -- between the teams.

Campbell: Sure. I think there was some tension between the two groups, and I was sort of caught in the middle of that. That was perhaps uncomfortable at times, but I think that I did help share technologies between the groups to some extent. It eventually became clear that the Deep Thought path was leading to better results than the Hitech, system which was limited in some ways. It was harder to parallelize and grow.

Spicer: That was based on basically a processor per square?

Campbell: A processor per square, so it’s obviously a less efficient way of generating moves. The ChipTest system -- the first system based on Hsu’s chip -- used a very simplistic evaluation function which allowed it to evaluate positions. So it was a very strong searching system but [had] a limited capability to evaluate. Deep Thought addressed some of those concerns and created a more comprehensive evaluation function.

Spicer: The evaluation function is incredibly important, and that’s what you worked on in Deep Blue?

Campbell: To a large extent. I also worked on opening book and search but—

Spicer: I think in one article… you can have all the search in the world, but if your evaluation function is poor….

Campbell: Then you end up with poor decisions.

Spicer: In ’89 there was a huge upset with Larsen and Denker. Can you tell us a bit about that?

Campbell: Sure. Well, these are two separate events. Hitech played a match against grand master Arnold Denker who was then, to be charitable, well past his prime, but he was still a grand master player. He lost this match to Hitech. I don’t recall the exact score. Denker did not actually earn the grand master title in the traditional way, but was awarded the title for previous good work, in a sense. So he was a grand master, but he was not a competing grand master to this same strength as Larsen, who was the grand master that Deep Thought managed to defeat in a tournament here on the West Coast. That was part of a tournament -- a large, open software tool works. It was a large tournament open to many, many strong players who participated in that tournament. Deep Thought got its chance to play its first really strong players there, and a win and a loss against a grand master, and then some defeats with some good, strong international players. So that was a “coming out” party for Deep Thought. It made itself noticed after that tournament.

Spicer: It must have inspired a lot of confidence in the team that they were on the right track.

Campbell: Well, I also think that that result got IBM’s attention as well. There was an IBM employee, Peter Brown, who had taken time off to get his PhD at CMU, and was there at the same time as the Deep Thought team. So he had gone back to his management back at IBM, and pointed out what was going on, and initiated IBM’s interest in hiring the Deep Thought team to build the next generation chess machine.

Spicer: That’s a really interesting topic so let’s start there. When I look at the history, from a corporate perspective, if I were IBM, it seems like one of those kind of “steeples of excellence”, if you like. It’s kind of a point project, and it was very well defined. But it’s unusual, and you think of IBM as a fairly staid, conservative company. I’ve always been intrigued as to whether they saw the public relations potential, or whether they had a genuine research interest, which I suspect is the case; they don’t tend to waste their money. But I’m very interested in the business understanding, the business case, of sponsoring the Deep Blue Project, because it went on for quite a few years and took quite a few resources..

Campbell: It did. It went on for seven years, which is unusual for a project that isn’t leading to direct revenue and income. I think there were at least two factors, that came into play here. One is that at the time, the Watson Research Center was , and still is, one of the top research facilities in the world, industrial research facilities. They were always looking to hire good people, and they saw in the team of us, independent of whatever project we worked on. I believe that they thought we were people that could contribute to IBM research, and so it made sense independent of that. But somebody -- I don't know exactly if it was the head of research, which at that time was Jim McGroddy. or I don't know exactly who -- but somebody had the foresight to say this project is sort of win-win. We can bring some good people to IBM, we can do some research on parallel algorithms, high performance computing, and there is this potential payoff down at the end, if we're successful, where we'll get a lot of recognition. Part of IBM Research's mission is to generate this awareness of IBM -- the term they use is luster -- the prestige of accomplishing scientific goals and awareness in the scientific community. Somebody had the foresight to see that there was a chance for this happening. There was also another event beyond defeating the grand master. But there was the first match between Deep Thought and Kasparov which was also—

Spicer: In New York—

Campbell: In New York, 1989.

Spicer: Kasparov at this time I think was still-- The human race, if you like, was not really in any danger of-- At least from his perspective I think he felt fairly confident that he was going to win—

Campbell: Oh, undoubtedly. I think grand masters could maybe lose an occasional game of speed chess to a computer, but in a serious tournament game I don't think they were in too much trouble.

Spicer: Did you feel when you went to IBM that -- just based on, say, the number of rating points that chess programs seemed to be improving by, every year or every two years, -- that you could almost draw… assuming there were no singularities or strange, bizarre meteors or whatever you want to call them, that you could almost tell, to the year plus or minus a couple of years, when you'd reach certain levels?

Campbell: Well, that's an interesting story. I guess people would draw that line and they'd say "but there's a tailing off effect" and it's going to tail off around, I want to say, 2200, master level. Then when you passed master level they say "oh, it'll keep going up for a while but then there'll be a tailing off effect around 2400", and then "2500". Then eventually as it surpassed each of those levels, people redrew this graph and always said that "there'll be a tailing off effect". I guess now we're seeing that they just keep going up. If you keep taking advantage of faster hardware and improving-- it doesn't even have to be revolutionary improvements in the programming but evolutionary -- things keep improving. There may in fact be a tailing off. I think there is. If you plot the year versus rating, it's slowing down. If you take out outliers, in a sense like Deep Blue, which sort of pushed ahead several years in terms of speed over the curve that present day PCs are on, I think it is continuing to go up. I don't see any reason to believe it won't continue to go up for a while yet.

Spicer: This raises an interesting question for me, which is: Obviously computers or software chess programs are rated by playing humans, or other computer programs of known ratings, but ultimately I think it's all traceable to human play. Is that fair?

Campbell: Yes, I think that's fair. I think if you're going to calibrate a program you have to calibrate it against human players.

Spicer: What happens when your program is better than the best player in the world and no longer has anyone to play against? How do you measure it?

Campbell: You can be better than the best player in the world and still not win every game. If a strong grand master plays [and] doesn't make any mistakes in a particular game, it'll probably end up in a draw no matter how strong you are. So you're relying, to win, on mistakes. You always assume that the natural result of a chess game is a draw unless there [are] mistakes made. As long as it's not a complete wipeout -- 100% for one side or the other -- you can always make some estimated rating out of that.

Spicer: Ok. Can it go past 3,000?

Campbell: That would be hard but possible.

Spicer: Maybe it doesn't mean anything past a certain level?

Campbell: It would just mean that a computer that's rated 3,000 would win three out of four against-- You've got 75% of the points against a Kasparov or a Kramnik or somebody who's close to 2800.

Spicer: So it's not as mysterious as I'm making it out to be.

Campbell: **No.** Once you get up beyond 3200 it would be very difficult, because it's just harder to estimate the parameters of the rating at that point.

Spicer: Only humans accept draws I think, right? Deep Blue was programmed not to accept draws. Only the team could sort of arbitrate—

Campbell: True. We had agreed prior to the match that there would be a human intervention in the case of draw, because if we left it up to Deep Blue it would-- We could have programmed in a draw acceptance function, but there is a problem with that. If it thinks it's winning, it will refuse all draws, even though our human experience tells us that this is not a winnable position. We may know from hundreds of years of grand master practice that rook and two pawns versus rook and one pawn on the same side is a known draw. And without a horrible blunder on the weaker side, it's going to be a draw. Deep Blue doesn't know that, and doesn't know all the exceptional cases that grand masters have discovered over the years where an advantage doesn't lead to a win. So it would, you could say, torture Kasparov in that situation all the way to the 50 move rule. The game could last an extra couple of hours needlessly. So to avoid that embarrassing and sort of unsportsmanlike outcome, we allow human intervention in that case.

Spicer: Some of the crazy endgames, not in Deep Blue probably, but it's possible to actually go over 100 moves I think with games—

Campbell: Certainly. Deep Blue had these endgame data bases where you calculate the exact result for all positions with five or fewer pieces on the board, in some cases with six or fewer pieces on the board. Deep Blue certainly had that, and many programs these days have that, and some positions do indeed take hundreds of moves to win.

Spicer: A really interesting comment you made, in Scientific American '96 I think, when you said something really interesting, that humans and computers have different strengths and different weaknesses, and it's hard to know what strengths are playing against what weaknesses when you put a human and a machine together. That struck me as fairly profound, because even though on the human side you expect a bit of maybe fuzziness and players to play more by gestalt or something very difficult to quantify, but you think of computers as fairly rational in the brute force approach especially if it's just, "calculation". But when you put them together it seems like you get an alchemy almost, a really interesting alchemy that can result in amazing games. Did you see a lot of really interesting games in the '96 and '97 matches? Things that people initially thought "that's a strange move on Deep Blue's part", then later they went "ahh, that's actually…"

Campbell: There were examples of that in the games against Kasparov -- moves that seemed counterintuitive or just wrong, just plain wrong. But there was a logic to them, and you could reconstruct that logic. I saw plenty of examples of that as we were programming and preparing Deep Blue to play the match against Kasparov, the two matches. Many cases where we were very upset about a move that it played, and we would assume it was a bug and conduct a deep investigation and find out it had to play that move, there was no choice. Any other move would have lost, or significantly worsened its position. But intuition sometimes let us down into thinking we knew what was going on, when we didn't always.

Spicer: Also when that kind of thing happens it can be perceived by Kasparov, and I think it may have been, as a bug, and that he may have played accordingly.

Campbell: Yeah. You have to be careful not to make too many assumptions about the opponent and just try and play reasonable moves and not build this model of the computer opponent. Because it's very hard to model what a brute force type program -- not that we're strictly brute force, but a brute force style program -- it's hard to model in the human mind what it's capable of. Because it's coming up with all these continuations that-- If you imagine that human beings have censors, that you look at a position and it says oh, all those moves, we'll censor those out, those make no sense, we'll just look at these ones, and then they add these possible replies, only these make sense, forget all the others. These censors work very well almost all the time, but they have tiny gaps where they rule out something that shouldn't be ruled out. Obviously the better players have smaller gaps than others, but we all have them. The programs don't have these gaps and they find these weird, non-intuitive, strange sequences of moves that afterwards in hindsight [we] say "oh, beautiful", but it's very hard to see them beforehand.

Spicer: When you say beautiful, I sense there's a sort of what scientists call elegance, if you like.

Campbell: Yeah. There's some elegance to, "oh, he's taking two steps back to take three steps forward and I never would have thought of that"

Spicer: You know the joke about: you shouldn't anthropomorphize computers; they hate it when you do that. But that is very interesting when all of a sudden-- Ed Feigenbaum talked to me a bit about that. When you apply search so strongly to a problem that every once in a while you get a result that you could not possibly have predicted, there's a sort of intermediate space, if you like—

Campbell: Yes, and we saw plenty of examples of that throughout the development of Deep Blue and earlier machines as well, where a move "has" to be wrong but it isn't. You see a move that has got to be a bug, but it turns out it's seen something that you don't see. Obviously as the programs got stronger and stronger this happened more and more often. If there's a strange move, it's almost always a good move rather than a bug -- as we debugged our programs, of course.

Spicer: Yeah, and it raises the question too -- I don't want to get too metaphysical here -- but it makes you ask: Is thinking an epiphenomenon of calculation, or is intuition? Because in the computer world it seems like when it comes up with these kind of crazy strategies which have, "elegance" or "beauty" or whatever, it's sort of in the middle. It's almost human. Anyway you'd like to think it is.

Campbell: Yeah. There's been speculation about the kind of processing that's going on in the grand master brain to come up with these judgments or intuition. Is it actually conducting a lot of small searches? What's really going on? It's really obviously difficult to know, because introspective studies are very poor at figuring out what's going on inside a grand master's head.

Spicer: So you are at IBM. Can you maybe go through for us the trajectory of the project from the beginning to ’97?

Campbell: Sure. We had a vision of where we wanted to go, what we wanted Deep Blue--we didn’t have the name Deep Blue yet -- what we wanted it to look like. But we didn’t think we were ready yet to make that big jump over to that system, I think we needed some more experience. We built over the next couple of years “Deep Thought 2”, which was based on the original technology, the chips that Hsu designed at CMU, but parallelized it to a much greater extent [and] enhanced the evaluation. That machine competed quite successfully for a couple of years before it went into semiretirement as we sort of focused on Deep Blue. We brought it out a couple more times after that with mixed results. But at this point we were designing the software and the chip that would form the basis for Deep Blue, and that was a quite long and painful process. Hardware design is something I wasn’t particularly familiar with, and so I certainly wasn’t-- Hsu was absolutely the designer of that system. But there was a lot of testing that needed to be done, and I got some experience with the difficulties in hardware design by helping to design some of the testing for that and run it. That was a long term process. That took a couple years from beginning to end. And then meanwhile we were also working on the software for this system, and deciding on the architecture. We had originally envisioned perhaps a stand alone machine with an off the shelf processor to control it, but we eventually moved towards the IBM SP parallel computers. They seemed like a much more sensible way, rather than designing the entire system from scratch, piggyback onto an existing system. That’s what we did, and designed the printed circuit boards to plug into these commercial computers, which was a relief not to have to design the entire thing. By the time we had the working chips, we also had simulations of them. We’d done some experiments that way, and we had initial software. Then bringing it up and getting it running in time for the ’96 match against Kasparov was very challenging, hard work much later than we had hoped. So there was very limited time for testing. But to our great happiness, it was held together long enough to play that entire match and in fact win a game against Kasparov, which was a tremendous success from our point of view. Even if it did lose the match, it gave us the confidence that we were on the right track, that if we patched some of the obvious flaws in the system and did some better testing we were going to be in much better shape the next time around.

Spicer: Didn’t you get the chip three weeks before the match or something?

Campbell: We had gotten earlier versions of the chip which had flaws in them earlier than that. But the actual chips that we used, yeah, three weeks prior to the match. Which was tremendously difficult to fully debug, and in fact wasn’t fully debugged. There were programming errors that caused crashes during that first match.

Spicer: Which Kasparov kind of interpreted in a negative way.

Campbell: I guess we hadn’t fully explained that possibility to him when we restarted the system a couple of times. I think after explanation he was understanding what was going on there. But it was surprisingly robust, and in fact these chess programs are surprisingly robust. You can discover the most horrendous errors you can imagine in the code, but because of this brute force capability and searching, it seems to always do something reasonable. It may not be ideal, it may not be what you intended, but it seems to always do something reasonable, because it's optimizing -- choosing a continuation that optimizes the material balance and some of the factors you know about. There may be this one evaluation function term that's pulling you in the wrong direction but it still ends up playing reasonable chess.

Spicer: Was it you who said Deep Blue just plays bad chess really fast or was it—

Campbell: I hope not.

Spicer: That must be apocryphal. So tell us, this is without the benefit, in '96, of really much tuning? And no grand masters?

Campbell: There was a limited amount of grand master help but very limited. Nothing like what happened for the '97 match.

Spicer: I wanted to pick up a bit on the evaluation function. That was your main-- one of your two responsibilities…

Campbell: Yes…

Spicer: …on the project. Can you tell us a bit about, without giving any secrets away, maybe how many terms or variables that evaluation function may have considered and how you came up with those?

Campbell: Well, the terms or the patterns that Deep Blue can recognize are actually encoded in the hardware, so those we had worked out over a long period of time. The '96 version had a few thousand, and the '97 version had more, on the order of several thousand -- eight thousand maybe -- patterns that it could recognize. Now there's sort of two steps that you need to take once you've got your set of patterns, because once we've got these patterns that's it, we're stuck with those patterns and if there's something else…

Spicer: …because they're in hardware?

Campbell: Because they're in hardware. There's something else we need to recognize - we just need to make do with what we've got. We have to have some surrogate for it or fake out the machine in some way -- there's various software hacks that can do it,, but really if you don't have a pattern you're out of luck. So given these 8,000 patterns that can be recognized, you have to decide first of all [in] what context are they , and in each of those contexts how important are they, what value they should be given. As an initial pass we just, off the top of our head, came up with some numbers based on our experience from Deep Thought and assign numbers. Some of the hardware terms that-- the patterns we weren't quite sure how to use initially and certainly in '96 we didn't assign any values to them at all. Over time we- we started using them more and more. So we gave them initial values and tried to determine which contexts were appropriate and assign the values appropriate to the context and let it play and observe its play and find out how it does against Grand Masters, how it does against other programs, and find weaknesses, retune the values. We also had a large test set of positions that we would rerun every so often to make sure that nothing was broken, that it was still solving the positions that it should solve and maybe it's got a few new ones that it hasn't been able to solve in the past, maybe not. These were a combination of tactical and positional types of issues because whether or not you find the right move is a complex combination of the search and the evaluation. So evolving those values and contexts we actually had a simpler problem in some sense than software programmers, who in principle can add new patterns any time they want However, of course, each time you add a new pattern it slows down the system a little bit which tends to weaken its tactical ability and there's this delicate balance between valuation and search that you're always playing. We don't have that problem. The evaluation is executed in the fixed amount of time and whatever patterns we have are evaluated in that amount of time or detected in that amount of time. If we needed to add more patterns like we did between '96 and '97 we just put them in and it doesn't take any longer, because it's all executing in parallel on the chip. So the big problem was, given some feedback from the Grand Master Sir Joel Benjamin, that's it playing a stupid move in this position, figuring out what's the cause of the stupid move. Sometimes it isn't a stupid move and it's his mistake. Sometimes it's the fault of the search, sometimes it's the fault of the evaluation, and we had to… Literally everyday he would come in and start playing games, send us notes about moves that if he thought were suspicious. And this wasn't moves that it played but moves that it was considering. It's always printing out as it's playing what it expects to be the normal continuation, and if you see a weird move in that, it may just appear for an instant, for one iteration, and it's gone, it's off on another track. But it's a sign that there's something wrong and you have to zero in on those and track them down. A very slow, very painful process and took the vast majority of my time.

Spicer: Is it fair to say that the evaluation function is really the knowledge half, and the computation is the search half?

Campbell: Yes, well the patterns, the context, the values you could say are the knowledge, but there's a lot of knowledge about search too. When to search more deeply here versus here - there's knowledge there as well -- so it's not a clear-cut distinction between search and knowledge.

Spicer: That's very interesting. Were there some some chess programs – Slate and Akins, I think introduce some really powerful techniques that were probably used by all programs.

Campbell: Yeah, they were the first ones to really do a good job on ‘brute force’. They really said ‘wow let’s just go for it; it doesn’t make sense, it’s impossible but let’s do it anyway’. And they found that it was better than their carefully engineered knowledge based program. And so they inspired a lot of people after that to take that approach and do it one better, and one better, and so on.

Spicer: They always kind of had the fastest Cray or Control Data…

Campbell: Yes exactly, they would use the fastest computer available and have very good results with it.

Spicer: Which could kind of tell other people that in ten years from now we’ll probably have the equivalent of this. You could see that hardware was on the way and that these folks were ahead of the curve because they just had better hardware,

Campbell: Right But they made the leap to pure brute force programs which you could say in some sense it is a step back for AI and the goals that AI had for games as a research vehicle.

Spicer: In terms of encoding human knowledge.

Campbell: Yeah the human based approach of trying to emulate human thought processes. But it supports the more engineering approach of doing whatever it takes to solve a problem, and they took that to the limit.

Spicer: What is the debt of Deep Blue to Belle?

Campbell: There’s a large debt. The original design of the Deep Thought chip, which provided a good part of the Deep Blue chip, was based on the Belle design. A somewhat improved design of the Belle move generator was done for Deep Thought. And of course Ken Thompson was always there at these events. And if you asked Ken the right question, he’ll tell you the answer - it’s figuring out the right question to ask him and that was always the joke. Because Ken was very open, he didn’t keep a lot of secrets, but he didn’t volunteer a lot of information either So if you could figure out the right question to ask Ken you could get a goldmine.

Spicer: The issue of time control, now, how does that affect Deep Blue or how did it affect Deep Blue’s decision process?

Campbell: I don’t think Deep Blue did anything radically different from the way other programs did in terms of time control; it allocated a certain amount of time for a move. If things were going wrong when it reached that limit it would give itself more time until it could try and resolve it. There was a final cutoff saying you can't go beyond this time because if you do you'll just screw yourself for the rest of the game, you won't have enough time. We actually did toy with a radical idea and we never actually did it, probably for the best, and that was to given Deep Blue just a few seconds for every move - maybe five, ten seconds for every move -- and the effect of this would've been that Kasparov would've been under a tremendous pressure the entire match, because he would've not have been able to get up and leave the board which he's accustomed to doing. No human opponent has ever done that to him before and it would've been quite dramatic!. There's nothing illegal about moving quickly. It's stupid for humans because they just don't have enough time to get their head around a position to make a good move, and they would just be mercilessly beaten. But the fact is that in these programs that search like Deep Blue, you get a large part of the benefit in the first few seconds of the search. And then you get to look two moves, three moves, four moves, five moves, six moves, seven moves, eight moves ahead, and only two seconds have passed. And then because of the exponential growth it starts to slow down. You may take the last two minutes to do one additional depth. If you'd sacrifice that one additional ply of search and move more quickly, we speculated that that would more than make up for the loss in playing ability because of the psychological pressure it would put on the opponent. But we were too scared to do it <laugh>.

Spicer: I remember watching a film of Robert Burn -- I guess he was the chess columnist for New York Times -- playing I think it was Deep Blue…

Campbell: …he played Deep Thought…

Spicer: …Deep Thought. I remember him saying that it was like playing an octopus: no matter what you did you just felt this thing had its tentacles around the whole board, and it was very sort of intimidating.

Campbell: You have to latch onto something that the computer doesn't understand very well to beat it. As they search deeper and deeper that's harder and harder to do. If you slip into the kind of position where computers are good, there's just no hope because they seem to… I passed that stage, I don't know, probably in the early 80s, where computers were trivial opponents, computers were tough, computers were way too good So they passed by me, but I could feel that if I could find a concept they didn't understand I could keep a hold of the game and steer the game, But if it slipped into something where it's an open position, lots of tactics, I could just feel it -- there was no hope, I was just trying to keep my head above water. I think that's what the top Grand Masters are feeling now; that unless they find something unusual they're going to be struggling to keep control of the position…

Spicer: …sounds like a boa constrictor or something…

Campbell: It is; it squeezes an squeezes; we might be able to find a weakness but it's not easy.

Spicer: So what happened at move 19 in '97 with Kasparov …

Campbell: Move 19?

Spicer: I'm sorry, maybe that's not right but in '97, the move that Kasparov, in game 2 I think it was kind, of lost his composure and people felt after that point…

Campbell: I think that move…

Spicer: He had lost the…

Campbell: '37…

Spicer: It ended in-- I think it ended in 19 moves maybe?

Campbell: That was Game 6. Game 6 was the short game that Deep Blue won in 19 moves. Game 2 was the longer game that ended in the Deep Blue win, where there was a couple of moves that Kasparov couldn't explain, that's what you're referring to. Somewhere around 35,36, 37 was the area where he had troubles. I think in hindsight we can see that modern day computers can replicate practically every move that Deep Blue played, given enough time. Sometimes it's just the obvious move that they play without even that much thought. So I think it's clear to say that there are moves that are suspicious. Just doesn't have a good understanding of what computers can and can't do.

Spicer: Can you compare for us Deep Blue's strength versus, say, the top commercial chess programs today running on a 2 or 4 way kind fast Pentium.

Campbell: It's pretty hard to do; I know that in 1997 playing against programs running on Pentium 90s or something like that, which was what we had, maybe 120Mhz, they were trivial opponents and we gave up on them because they didn't challenge. That's why we got the Grand Masters to come in and play, because the computer opponents didn't challenge Deep Blue enough to lead to any insights for us. Over the years they've gone a long way since then, the commercial programs. It's hard to speculate exactly how well they would do against Deep Blue, but I think there's been a lot of advances in search and in speed of the machines, and in evaluation. I would suspect the top programs would be as good as Deep Blue was.

Spicer: Here's a strange little question for you: if you had two Deep Blues and they played each other, what would happen? Would it be an eternal draw?

Campbell: No, I guess it depends. If you start from the opening position… In fact Deep Blue wasn't entirely deterministic, and so it might play differently depending on completely random factors.

Spicer: Would white still have an advantage?

Campbell: There's no guarantee. I don't think we ever did it without an opening book directly from the opening position to see what would happen, and it's not terribly illuminating, I don't think. I don't think we would've learned as much as playing against a human opponent. It might be a good debugging technique but doesn't illustrate too much.

Spicer: Let's say you were Kasparov's -- this is a little off the wall -- you were Kasparov's trainer or advisor. How would you have advised him before the rematch in '97?

Campbell: I would've advised him to do to some extent what he did do, which was try and keep the position closed, non-tactical. Avoid opening preparations and standard openings, because the team that's preparing Deep Blue, we would have prepared for Kasparov's repertoire. So play slightly unusual stuff and avoid tactical battles where you're gonna lose. He tried to do that, but I think he didn't stick with that strategy as completely as he might have. Particularly in the last game, which he lost, he went completely in the wrong direction. He went into this… He forced Deep Blue to sacrifice a knight in return for a huge attack. He forces himself to play absolutely accurately move after move, and even then he might still lose. It was a huge risk. I think that was a particularly poor decision on his part to play that opening. If he played a much more closed opening and tried to keep the control of the position, at least the thread of the position so it's not spinning out of control, I think he would've had a much better chance.

Spicer: Can you explain what anti-computer chess is and if this relates to what you just said?

Campbell: I know it when I see it. <laugh> Typically what I try to do when I play against computers is exchange as few pawns as possible, so there's few open lines. of the fewer the open lines the easier it is to calculate, because there aren't as many moves, there aren't as many options, and there's not as much interaction between the pieces. There's the pawns blocking the way and they're looking at each other through the pawns. I found that allows me to keep control of the position for a good long time before I finally fall apart and lose to computers-- modern computers. So anti-computer play is, to some extent, trying to hide behind a wall of pawns and keep things on a strategic level rather than a tactical level. Looking ahead on a long-term I'm going to attack on the king side; there's no rush, I've got 10, 15, 20 moves to do it. Just to try and suppress all the counter play that the computer's trying to generate. If I can close up the position completely, get this locked pawn chain, I'm really happy then because it allows me to stop having to calculate as much and think much more strategically. It's harder these days, because programmers know that people try and do that and lock the pawn chains, and they try and avoid those in their opening books or in their evaluation function. They may even give up some things in order to avoid a locked position.

Spicer: Right. So they get anti-anti computer chess. One thing that I… After this exhibit, which culminates in Deep Blue and so on, I actually had more respect for Kasparov in a sense, because the fact that it took that machine 200 million calculations a second or whatever to beat a human. It actually says a lot about the human. I mean, we shouldn't feel bad.

Campbell: It is amazing that people-- somebody like Kasparov, who in my mind the best human player ever, some of the insights that he has… I've gotten to interact with him and see his thinking through a position prior to the matches. The insights that he has, and he says 'oh, and then white will win'. I can look at it and think for a good long time and maybe get the vaguest hint of what he's talking about but the deepness of what he and the other top Grand Masters are thinking… I got to interact with the Grand Masters on the Deep Blue team and saw some of that as well, and how they struggled to explain this intuition and put it into operational terms so we could program it. What they know is amazing, and it takes not only special talent for the game but a number of years, many years of training to get to that level.

Spicer: Yeah a real genius I think too, something extra special.

Campbell: Yeah, yeah.

Spicer: You've already said in previous things it's not thinking. In fact we have one of your quotes on the exhibit, you probably haven't seen it yet so I won't bother - unless you want to comment on people's perception of Deep Blue thinking.

Campbell: I think thinking implies intelligence. Intelligence implies a broadness of knowledge that Deep Blue, of course, does not have. Deep Blue is extraordinarily good at this one very well defined problem. If, for example, you change the way a pawn moves very slightly, a human Grand Master would have no problem adapting to that. They may not like it, but they could certainly… Deep Blue wouldn't even work any more. It's so brittle that something like that would break it, and you'd have to reprogram it and fix it. It would take some time, particularly if it required new hardware. So in that sense Deep Blue doesn't have the breadth and robustness to be intelligent.

Spicer: So Feng-Hsiung Hsu also gave a good little analogy, where he said if the building that the match was in caught on fire, all the humans would run out but Deep Blue would still keep calculating.

Campbell: Right. My favorite cartoon is the one where you see Kasparov and Deep Blue facing off across the chess board, and there's this thought bubble above Deep Blue. You can see pieces moving around - rooks and bishops and knights moving around on the chess board. And there's this thought bubble above Kasparov and he's thinking about taking his foot and pulling the plug out of Deep Blue. And that's in a nutshell why Kasparov's intelligent and Deep Blue's not: because he can step outside the domain of chess and say 'oh I just pulled the plug, I win'.

Spicer: You'll see a very similar cartoon in our exhibit actually; it says "Kasparov beats Deep Blue in one move" and it's him pushing a big switch. It's the same idea. One of the museum's missions, of course, is to inspire youngsters. I'm sure a lot of people look at your work and are quite inspired, and by Deep Blue as well. Is there anything you'd like to say to them, and people, kids that are interested in computer science or even chess?

Campbell: I think I would say that 50 years ago or more computer scientists in this new field said 'chess would be a great grand challenge -- if we could get a computer to play chess we've really done something at a high level'. There are a lot of grand challenges out there right now, around intelligence, and getting a computer to understand human language, represent knowledge about the world, simple things like common sense which the computer doesn't have. There are many, many challenges out there that I think in the next 20 years we'll see amazing progress on, and I'm really looking forward to seeing them.

Spicer: Great, well I think we'll leave it there and thanks again Murray that was a wonderful interview.

Campbell: Thank you.

Spicer: Thank you.

