Constellations
Podcast

Special Episode – Apollo 11 Revisited, 50 Years Later

Guest: Hugh Blair-Smith, Apollo 11 Software Engineer and Computer Programmer, Draper - 15 minutes

John: Welcome to constellations, the podcast from Kratos. My name is John Gilroy and I'll be your moderator. Our guest today is Hugh Blair-Smith, one of the original engineers who worked on Apollo. Now everyone's listening to this, whether now or six months from now, they've been bombarded with all kinds of front page news and radio spots and everything else about the Apollo mission from 50 years ago. We are celebrating the 50th anniversary of the moon landing by learning more about it and the men and women who made it possible. Our guest on Constellations podcast is Hugh Blair-Smith, who was one of the Apollonauts, as we call those engineers, who worked on the Apollo missions, Hugh worked for 22 years at MIT's instrumentation laboratory. Later renamed Draper laboratory after its founder Charles Draper. Through his career, he worked on several NASA programs and published a book called "Left Brains for the Right Stuff-, Computers, Space, and History" about the Space Race. So Hugh, what was your role in the Apollo program?

Hugh: My main role in Apollo was providing programming tools to the MIT engineers augmented later by hundreds of contractors who wrote the Apollo guidance computer or AGC Code. This included defining two dialects of assembly level programming language and creating the program that funds later the source code in those dialogues to the ones and zeros of object code that the AGC could actually handle, but it also included taking part in the design of the ATC hardware, making sure that its repertoire of instructions would be as easy and efficient as possible for the programmers to use. All this work was equally influential to every AGC program and every flight. To me, the best form of influence was that there was no conscious influence. My tools and other participants tools just worked and were totally taken for granted.

John: AGC Of course means Apollo guidance computer, is that right? Yeah, and I think, you know, there are people, it possibly even my kids who are listening that may be confused and and thrown curve balls and they hear about this machine language and and assembly and what you really did, but I'll give a summary. They took a very, very complex problem threw at you and said you don't have any budget figured out with the limited hardware that you have there and you were working with a very, very small computer that had had very minimal memory and somehow you waved a magic wand and you figured out how to get to the moon.

Hugh: That is certainly correct. And I can help you a little bit with assembly language for people who are not familiar with it. Most of the programming language we use, we say something like C equals A plus B. But the assembly level way to do same thing is to say pick up A, then add B to it, then store the result in C. So it takes three instructions to express that real simple equation.

John: Yeah, that's great. So I'm looking backwards, and historians always like to look backwards and attribute motivations, and someone say, well, the reason that Hugh worked so hard is because he wanted to beat the Russians to the moon. Other people argue that, no, no, you put a problem in front of Hugh and he's going to just go through that brick wall and find it. So, so what was it, the brick wall or going to the moon?

Hugh: My fuller answer is this. Like all Americans, I winced at the hammer blows at the big Soviet lead in space technology. Sputniks and Geigers. They used it to suggest to newly decolonized countries that the decadent west was in decline and that the new Soviet man would carry all before him. President, John F. Kennedy had the vision and the energy to propose how the West, with the USA in the lead, could turn that psychological weapon around and his sense of timing enabled him to put that proposal before Congress when the Congress was ready for it. Like most Americans, and I'd guess just about all American engineers, I embraced this approach as a way to solve the Soviet issue by achieving the coolest thing that has ever been done.

John: "The coolest thing". Well I never thought a guy with your education background would be seeking out the coolest thing. I mean, Harvard and MIT. Wow, that's pretty impressive.

Hugh: So now you see why I answered that, with yes.

John: President John F. Kennedy so famously said, "We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard". What was the most difficult or daunting challenge you remember facing during the Apollo program?

Hugh: Well, the one for me personally, the biggest continuing challenge, was helping programmers to know what was in each Apollo program, what wasn't in it, how the pieces connected and supported each other, and how some of the pieces were in contention with each other. I kept adding cross indexes and other sub tools to help the programmers and the managers find their way around first 12,000, then 24,000, then 36,000 lines of code. What would be not enough information and what would be too much information?

John: 36,000 lines of code. Not as easy to write as today. That code was written with the computer cards, wasn't it?

Hugh: Yes, by today's standards you say it's very carefully hand tailored. Each step.

John: A bespoke at collection of software, very hand tailored.

Hugh: Certainly bespoke. You know, every 15 bit word of program- If you could save one, you did a good thing, every millisecond of time. If you could save it, you did a good thing. Those things were important.

John: When you look at what happened 50 years ago, NASA and contractors such as Draper were brought together to work on this project, sending men to the moon. How do you get all those people, all over the country, to focus on one project at a time? I think project management must've been the biggest challenge.

Hugh: Project management is certainly one of the biggest benefits from the whole program and one that isn't really mentioned very often. But as far as the computer technology goes, the fact that we jumped in and gave a powerful kickstart to the integrated circuits, that probably moved up by a few years, the progress of all the amazing chips and circuits that we use today. Well, I suppose the nearest thing to it was really the Manhattan project, but in some ways it wasn't easy, but it was relatively simple in its structure. We had just thousands of parts. We had hundreds of subsystems and making all this stuff play together was an enormous challenge for MIT and for NASA and for all the contractors.

John: When you look at the 1960s and what happened in 1969 and I'm not talking about Woodstock, what happened in 1969- is there a clear before and after the moon landing? Was that like the big mountain? Was that the line of demarcation?

Hugh: My approach to that point is that it's funny about lots of engineering activities that they are completed years before anybody outside the project is even aware of what is being created. Tool makers like me, for instance, find that our products have to be complete and usable well ahead of much else. That's part of the work I did for Apollo was substantially finished before even the first successful manned Apollo flight. By the time of the moon landing, I'd spent most of two years on future projects, much of which went into the space federal. I was certainly enormously interested in the moon landing and felt that once they've set foot on the moon, we've achieved step one of Kennedy's promise, but step two wasn't complete until the astronauts were back on the carrier. Those were important emotional milestones. As for what I did at work day to day, that wasn't a key date in what I was doing.

John: When you look back at your work that you accomplished in the 1960s what technology from today do you wish you might've had back then?

Hugh: Yeah, well that's a tricky question. There were a lot of features like high level compiler languages to floating point arithmetic hardware that keeps track of the decimal points for you that existed then and were well known, but with the limitations on processing speed and memory capacity that we had to work with, we couldn't use such features and still get the job done.

John: You know, Hugh, thousands of people from all over the world have listened to this Constellations podcast. If you are listening now and you'd like to get alerts when new episodes are available, simply go to Google and type in Constellations podcast. Click on Kratos and sign up. You know, we've had listeners of this podcast from just all over the world. It always amazes me they'd be fascinated with your interview. I know that. Let's bring it up to today. The current US administration earlier this year reinforced its goal to send astronauts to the moon within five years. I think it's 2024 or something like that. Why do you think the human stopped? Why do you think the American government stopped sending people to the moon and what are they attempting to get from landing again?

Hugh: Well, I think we stopped as soon as the major political drivers of moon travel ran out of steam when there was no need to run up the score after 12 and American and zero Russians had walked on the moon and when the project had completed all the scientific exploration that most Americans wanted to hear about. Now however, the moon's low gravity offers an attractive platform for building bigger projects to send humans to Mars or to build enough solar power collectors to replace all fossil fuel use on earth.

John: When I read about the Saturn-5 being manufactured and built, they talk about very, very highly skilled welders at accomplished just miracles with building that, that huge rocket. My guess is that there are still new challenges that are being faced today. So what do you think the challenges are to putting people on the moon today that didn't exist back then?

Hugh: Well, oddly enough, I think one of the biggest challenges is in the reliability of software. Instead of about 75 kilobytes all put together by a close knit group working in the same building, we are now used to many gigabytes of code written by millions of programmers around the world who don't know each other and don't have the kind of mutually supporting culture we had back in the day. It's surprising to me that these vast agglomerations work together as well as they do and it's no longer a shock when they bind up and stopped for reasons that may never be understood. We are too often in a situation where shutting off the system's power and restarting is the only remaining fallback. Having said that, we did use full scale software research to get past the 1202 and 1201 alarms in Apollo 11. Maybe a lot of progress can be made just by rediscovering how to reboot our modern systems in a small fraction of a second as the AGC did instead of multiple tedious minutes as our new machines do.

John: And I think the word is "sloppy". When you only have a certain amount of space to work in, it's like moving with a small truck and moving with a train. You know, with a small truck, you've already carefully pack everything in and it's reliable. If you have gigabytes to work with, you can see that some of that could maybe be a little sloppier than in the past. Would you agree?

Hugh: I agree with that. I think for instance, that when whenever you get a multi gigabyte download for most apps into your phone or your desktop, that you're probably getting a complete copy of all the fonts that were ever used for display or printing. And I'll bet my desktop has got well over a thousand copies of all the world's fonts.

John: And it can be sped up by eliminating redundancy I'll bet, huh?

Hugh: Yeah. I mean we're kind of used to the fact that if your computer isn't running fast enough, well don't worry about it. Just go on Amazon and order a new chip that's 10 times as fast as the one you were using and while you're at it, order, a new memory chip that's 64 times as capacious as the one you were using and there's just no motivation to attempt efficiency, you just spend a little money and get a huge amount of additional hardware capabilities and sometimes that pays off as intended. And sometimes I think it turns out that the stuff you just bought didn't really help you that much. And you have to get smarter.

John: Back in 1969, the United States was racing against the USSR to set foot on the moon first. In recent years, we have seen more and more global cooperation in space, for example, with the ISS. But there are also more companies investing in space, creating further competition. What can you tell us about cooperation and competition when it comes to the success of those projects?

Hugh: Well, this is a very high level subject which would take more time than we've got, but I just have a little thought on the subject. When nations or companies compete, the goals are a mixture of political and economic. Insofar as the airlines still compete for passengers, they seek greater market share by improving the quality of service. That's an economic goal. But when they contemplate the absurdity of having a separate airport for each airline in any given metro area, similar to London railway stations, they get more interested in the political or social benefits of a pooled facility. the hugely ambitious rocket companies are now chasing prestige and recognition, which are competitive political goals and will come later to economic competition.

John: I was reading your biography in the NASA office for logic design, and if I'm not mistaken, you were working on something called MoD1, which was robotic propulsion to Mars. All of a sudden, 2019 we have this guy, Elon Musk talking about the same thing. It seems like you were ahead of the game here weren't you, Hugh?

Hugh: Well, MoD1 computer was supposed to go to Mars, it's on display by the way, in the Draper laboratory lobby for anybody who wants to visit a model of it. But all it was going to do, it was going to fly to Mars, get one close up picture, fly back to Earth and drop off the film cartridge for an airplane to pluck out of the air. Just one photo.

John: How much math would be involved in that?

Hugh: Yeah, but you know, we didn't have any better way to get a close up of Mars in those days. But anyhow, I think, in terms of the engineering limits of the present and future, is that the most difficult engineering problems in such projects is protecting the softer delicate tissues of which we humans are constructed. And that's not only in space vehicles, but on the surface of the hostile worlds we want to explore.

John: You know, a lot of people want to hear more about what happened behind the scenes, those 400,000 people, how they work together, software development and the rocket training of the astronauts themselves. And I think you've written a book that kind of shows what's actually went on. The book's called "Left Brains for the Right Stuff- Computers, Space and History". Tell us about this book please.

Hugh: Well I think the dedication in the book says it and this is short, I'll just read it to you. For the many individuals around the world who cooperated gladly, generously, and passionately to add rocket science, navigation science, and especially computer science. The electronic left brains to the right stuff. They answered the long fought quest for a moral equivalent to war and changed the world.

John: Pretty good for a rocket scientists. How very articulate. Well, Hugh, unfortunately, we are running out of time here. I'd like to thank our guest, Hugh Blair-Smith, one of the original engineers who worked on Apollo. Thank you.

Hugh: Thank you for inviting me.