# DOWNLOAD PDF THE FIFTH GENERATION COMPUTER PROJECT

## Chapter 1 : The Fifth Generation Project in Japan

*The Fifth Generation Computer Systems [Present and Future] (FGCS) was an initiative by Japan's Ministry of International Trade and Industry, begun in , to create a computer using massively parallel computing/processing. It was to be the result of a massive government/industry research project in Japan during the s.*

Vacuum Tubes The first computer systems used vacuum tubes for circuitry and magnetic drums for memory , and were often enormous, taking up entire rooms. These computers were very expensive to operate and in addition to using a great deal of electricity, the first computers generated a lot of heat, which was often the cause of malfunctions. First generation computers relied on machine language , the lowest-level programming language understood by computers, to perform operations, and they could only solve one problem at a time. It would take operators days or even weeks to set-up a new problem. Input was based on punched cards and paper tape, and output was displayed on printouts. Census Bureau in  Transistors The world would see transistors replace vacuum tubes in the second generation of computers. The transistor was invented at Bell Labs in but did not see widespread use in computers until the late s. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, more energy-efficient and more reliable than their first-generation predecessors. Though the transistor still generated a great deal of heat that subjected the computer to damage, it was a vast improvement over the vacuum tube. Second-generation computers still relied on punched cards for input and printouts for output. From Binary to Assembly Second-generation computers moved from cryptic binary machine language to symbolic, or assembly , languages, which allowed programmers to specify instructions in words. These were also the first computers that stored their instructions in their memory, which moved from a magnetic drum to magnetic core technology. The first computers of this generation were developed for the atomic energy industry. Integrated Circuits The development of the integrated circuit was the hallmark of the third generation of computers. Transistors were miniaturized and placed on silicon chips , called semiconductors , which drastically increased the speed and efficiency of computers. Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system , which allowed the device to run many different applications at one time with a central program that monitored the memory. Computers for the first time became accessible to a mass audience because they were smaller and cheaper than their predecessors. An integrated circuit IC is a small electronic device made out of a semiconductor material. Microprocessors Present The microprocessor brought the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip. What in the first generation filled an entire room could now fit in the palm of the hand. In IBM introduced its first computer for the home user, and in Apple introduced the Macintosh. Microprocessors also moved out of the realm of desktop computers and into many areas of life as more and more everyday products began to use microprocessors. As these small computers became more powerful, they could be linked together to form networks, which eventually led to the development of the Internet. Fourth generation computers also saw the development of GUIs , the mouse and handheld devices. Artificial Intelligence Present and Beyond Fifth generation computing devices, based on artificial intelligence , are still in development, though there are some applications, such as voice recognition , that are being used today. The use of parallel processing and superconductors is helping to make artificial intelligence a reality. Quantum computation and molecular and nanotechnology will radically change the face of computers in years to come. The goal of fifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

## Chapter 2 : Who Killed Prolog? | A Programmers Place

*The Japanese Fifth Generation Project in computer technology was an attempt to leapfrog Western computer expertise and create an entirely new computer technology. Although the generation terminology is a bit murky, there was the general perception that there had been a a number of generations of computer design and the accompanying operating methods.*

There are a thousand programming languages out there Literally, it seems, according to people who actually count such things. A classification of so many species is bound to be complex and subject to much debate. I attach to the name of the class of programming language what I consider to be the first exemplar of the class, in chronological order: Compared to these three, Prolog has fallen far behind. Hence the title of this article. Even for those who were aware of the project at the time, it is now worth reviewing its fascinating history in the context of its time. This article is such a review as well as a theory of how Prolog was killed and how Lisp was saved from this fate. A convenient starting date is  The military and political stand-off between the US and the USSR had long occupied centre stage, but is now replaced by the industrial and commercial rivalry between the US and Japan. Japan, devastated and dirt-poor in , had, while nobody was looking, transformed itself into a gleaming model of everything enviable in a modern industrial society. Not only good at watches, cameras and consumer electronics, but also at bullet trains, industrial robots, cars, steel, and mainframe computers admittedly, plug-compatible with IBM machines. Though Japan was commercially daunting in the extreme, it was a consolation that it could be belittled as being imitative rather than innovative. Another way in which Japan was seen to be competing unfairly was the way in which Japanese companies especially the keiretsu could get away with anti-competitive behaviour not allowed for their American counterparts. Unfair competition, because so un-American. Neither ten years earlier, nor ten years later would such a book idea have been viable. In it hit the sweet spot. With the stage set in this way, imagine the impact of the news that MITI had orchestrated a project to initiate the development of an entirely new kind of computer system. On the software side it embodied just about everything that had been a goal of AI research. On the hardware side, it was to be massively parallel. The marketers at IBM had taught the world to think about progress in computer hardware in terms of generations. They said that the use of vacuum tubes relegated a computer to First Generation, that the use of discrete transistors indicated Second Generation. So, when the IBM came out it was not just a new type of computer, it was a new Generation, the Third! During the s things got muddled, as there did not seem to be a clear criterion for Fourth Generation. The project was associated with two words that seemed calculated to make Westerners nervous: MITI for the reason mentioned above. AI because it is one of those things that cannot be contemplated dispassionately: In between these normal periods there are episodes in which AI is embraced with wildly unrealistic expectations. The year was the beginning of one of these. Japan was seen to be taking off from its current platform, already of daunting power, to shake off any remaining shackles, start innovating, and continue on to world domination. In the corridors of power around the world there was much scurrying around. The question that reverberated in the minds of ministers in charge of such things as Industry, Technology, Trade, Commerce, Skilled Manpower, or what not, was: What is the Appropriate Response? In the US things could not be as simple as the government allocating a pot of money and then handing it out to researchers presenting themselves as worthy recipients of largesse. As a result the US response was more interesting. If the government could not respond, could not industry form a consortium to ensure that the US would stay ahead of the rest of world in Fifth Generation Computer Systems? No, such formations were illegal under anti-trust law. Mere lobbying would probably not have been enough for such a complete and timely legislative outcome. Feigenbaum and Pamela McCorduck and published in  After proclaiming how expert systems were going to give rise to Knowledge Industry causing Knowledge itself to become the new Wealth of Nations, Feigenbaum and McCorduck continue with: To implement this vision the Japanese have both strategy and tactics. Their strategy is simple and wise: It is unlikely to be completely successful in the ten-year period. Even partially realized concepts that are superbly engineered can have great economic value, pre-empt the market, and give the Japanese the dominant position they seek. In the atmosphere that gave this book a warm reception, a judicious amount of lobbying was sufficient for the National Cooperative Research Act, which weakened anti-trust legislation sufficiently to make the response consortium legal. As leader a suitable admiral was found, perhaps inspired by the Manhattan Project under the leadership of a general. There was plenty of opposition to the FGCS project and the various responses. A common argument was that the FGCS project was not to be taken seriously. The volume, edited by T. Moto-Oka, still lingers in many a library. The conference officially kicked off the project. Some of the papers are by steering committee types and describe how breakthroughs in AI, software, and hardware were going to lead to computer systems transforming society to new levels of harmony and prosperity. But there are also papers by computer scientists, notably by K. Furukawa later to become a group leader in ICOT. Prolog is the language and logic programming the methodology. Parallelism was seen as the hardware imperative, and Prolog with inference as basic computing step seemed to have much potential in this direction. Hence, FGCSs were to be parallel inference machines. Fast forward to  The world looks very different. In the Nikkei Index, which had risen strongly for an unprecedentedly long period, from the beginning of the FGCS project, was about to breach 40, But instead of continuing its rise, it started a decline and was down to half the peak value by  The Lisp machine companies were either totally dead, or surviving only as something else than a Lisp machine company. The rapid increase in speed of the commodity processor helped to kill interest in parallelism, which had been found harder to exploit anyway. The parallel Prolog version of a Lisp machine, so exciting a prospect in , had become a relic. The lesson is that outside the waffling steering committees, people have to choose between technologies and they choose what they fall in love with. In my next article I plan to review the history of how Prolog came to appear on the radar of the Japanese when the sky was cluttered with Lisp echoes; echoes caused by people who fell in love with Lisp. I will describe as best as I can what causes people to fall in love with Lisp and how the same thing can happen for Prolog. Postscript February 11, Richard Grigonis made the following comment: The funny thing about this is that, in , as I recall, Fifth-Generation Computer Systems project director, Kazuhiro Fuchi, came to Columbia University in New York, along with Edward Feigenbaum, to give a speech speech and answer questions of students. Feigenbaum was railing about how the Japanese were going to take over AI and the world, and we should better fund AI researchers in America or we would all be left behind in the dust. It was as if he was using Fuchi as a prop to get more excitement in America for AI. A sort of embodiment of the first-order predicate calculus? BUT Americans need more support because the Japanese are advancing the field! It was the strangest moment. He was a bit miffed at me, but when he discovered we both had just purchased Atari computers, he warmed up a bit, and began asking me questions about its graphics, as his wife was working on a multi-colored quilt and wanted to use the Atari to help design it. My friend was more into Atari graphics at that point and answered his questions. The top American researchers knew the FGCS was completely flawed, but we were humoring them and making a big deal of it so we could get better funding for other, LISP-based projects in the U. FGCS was special in that for the first time Japan was not in copycat mode, but struck out in an original direction. AI researchers were only too happy to supply suggestions. In the UK the Thatcher government was especially sensitive to perceived lack of industrial virility. This resulted in the lavishly funded Alvey Programme. The incident you report is a nice example of the duplicity of many of the researchers funded in the US and UK. I agree that FGCS was destined to fail. Warren shows that in the four years of Prolog implementation development an efficiency in terms of execution speed and memory use was reached that equalled what was reached by a quarter of a century of Lisp implementation development. This is remarkable for a language that in some repects is more high-level than Lisp. The Japanese were smarter than researchers like Feigenbaum in that they took the trouble to discover that Prolog was a different animal from resolution-based automatic theorem provers, where the search space was pruned by the paramodulation technique you mention and by several others. Prolog is also based on resolution logic, but its inference is restricted to mimicking the function definition and the function call mechanism that has been the mainstay of conventional programming since Fortran. As Lisp also relies on this it is not surprising that since their performances are similar. In applications where Lisp need not search, Prolog does not search either. My present opinion is that the difference in the strengths of the two languages does not make one of them overall superior to the other. Other things being equal I might now recommend Lisp because its development has steadily continued since the time when interest in Prolog plummeted with the demise of FGCS. I believe that FGCS was a plausible idea and was similar to the idea behind the Lisp machines of the time. FGCS failed because it failed to come up with a convincing demonstration. Such a demonstration should have come in the form of at least a prototype comparable to a Lisp machine. It could have been clear from the start that a project headed by government bureaucrats and staffed with technical people on loan from big companies had little chance of coming up with a convincing prototype. A Prolog version of a Lisp machine was at least as promising as the Lisp machine itself. I believe that the failure of the Lisp machines was not predictable. Around everybody was caught off-guard by the rapid drop in price and rise in performance of commodity PCs. There were a few exceptions: Larry Ellison and Linus Thorvalds come to mind.

## Chapter 3 : Fifth Generation Computers : Free Download, Borrow, and Streaming : Internet Archive

*The Fifth Generation Computer Project is a two-part book consisting of the invited papers and the analysis. The invited papers examine various aspects of The Fifth Generation Computer Project. The analysis part assesses the major advances of the Fifth Generation Computer Project and provides a balanced analysis of the state of the art in The Fifth Generation.*

It was originally started after much debate about the necessity for significantly more usable computers which should proliferate "like air" throughout society and among other things take care of the ageing population and life-long learning. The MITI people who sponsored the project must have had a good marketing consultant to pick the name of the project, for just its very name has inspired a lot of interest around the world. This conference was held to report on the results from the intermediate phase of the project just before the final phase of integrating together all the innovations to a complete computer system. Unfortunately, the results were somewhat disappointing from the perspective of whether something new could be done with the fifth generation computer. Most of the applications presented at the conference were interesting because they were "X done in logics programming" â€” not because they were "X done better than before. We will have to wait and see. The project director, Kazuhiro Fuchi gave the keynote speech and compared the three project stages to "hop, step, and jump," saying that they had now taken the step and were getting ready to jump in the final part of the project when they will produce a massively parallel machine. Fuchi was also very enthusiastic about natural language processing which he said would be the link between human and machine in their project. Many of the Japanese scientists and engineers I talked to from outside the Fifth Generation project were actually quite skeptical about the direction of the project and not very hopeful about spectacular results. Maybe even some of the project leaders themselves have had this feeling since they had started a new laboratory for applications-oriented research within the project. The purpose of this laboratory is to verify whether the parallel computers and systems software built by the other laboratories can be used for the next generation of expert systems. The Sputnik Effect Actually, to some extent, the biggest result of the Fifth generation project came by before they even started on their own research, since the very fact that the Japanese were doing a big computer project scared a lot of European and American decision makers half to death. Fortunately they were not scared completely to death but instead decided to "counter-attack" by funding a lot of new research in different computer fields. Representatives from these three initiatives were invited to the conference to report on their own progress which has been fairly substantial. Timothy Walker from the U. Information Engineering Directorate told of the varying British information technology projects which had succeeded in getting a number of key scientists to return to the U. Several changes had been made in the projects over the years, including some name changes, such as changing "AI" to "knowledge based systems" because of lowered expectations. On the other hand, the area of human-computer interaction was receiving more emphasis now: At the start of the Alvey project 5 years ago, HCI might have been seen as important but not so much was done about it, whereas now they realized that they had to make a serious effort to ensure usability. Walker said that HCI could either be done as an independent field of study or integrated with other topics and that they had chosen to base their projects mostly on the latter view. His main these was that we are moving towards a society where research takes on a productive force in itself, or a Wissenschaftsgesellschaft as he called it something impressive about these long German words. This is in analogy with the industrial revolution where agriculture suddenly could be done by a small part of the population and the major force became industrial production. Now this production can also be handled by a small number of people, while the number of scientists on the other hand is growing rapidly. Siekmann reported that the number of scientific journals doubles every 15 years, the number of books in university libraries doubles every 10 years, and the number of scientific publications doubles every 5 years. The knowledge explosion has already reached a level where in chemistry it is often cheaper to conduct a possible duplication of an experiment than to search the literature to find whether the result of a previous experiment can be used. As an example, traditionally you could think of a bank as a building; now it is a computer network and the nature of its world-wide services can be changed by a handful of programmers at the head office. This view is to a large extent true when it comes to user interface research. Perhaps having the heavy focus on hardware development in the fifth generation project in mind, Simon asked whether hardware speeds had really been the bottleneck to progress in the cognitive science field. In some areas like chess playing it clearly had, but in general Simon felt that hardware was not a real issue when it came to the ideas pushing forward cognitive science since it was not normally necessary to have a program perform in real time. Actually even for chess playing, Simon stressed that the best human players are quite slow and only look at maybe states before making a move. A computer may have to look at 3 million positions to make the same move, so probably the right way of looking at the problem would remove the need for fast hardware. In many particular domains programs exist which surpass human intelligence. To get knowledge into a system, there are basically two methods: Currently we cannot use programming on humans, but maybe it would solve many of the problems in AI if we could shift to using learning for computers. Currently we use a few such methods such as showing which files belong in the same folder by showing their icons within the same window. But most of these visual languages are quite primitive in their expressive power compared to verbal languages in spite of the prevalence of picture-like representations in human reasoning. Computer-Supported Jazz One of the most interesting presentations at the conference from the user interface perspective was given by Keiji Hirata from NTT who talked and played about computer-supported jazz. The goal of the work was to produce a workstation for musicians called ICOTone and to have it generate jazz music. Hirata had encoded the music theory for tension and other concepts in bebop-style jazz on the PSI Personal Sequential Inference machine, thereby allowing users to construct a jazz performance by interactively specifying different aspects of the piece. Users could specify different amounts of jazz information depending on their level of skill such that expert users can get a high degree of control over the result while novices can still get the computer to play jazz. Hirata finished his presentation by playing a tape of a performance by his system which was fairly good and did sound jazzy. I am not going to throw away my Dizzy Gillespie CDs just yet to replace them with this system, though, since the jazz workstation is more in the nature of an interesting piece of research showing hope for the future. Dynaware was introduced to me as a small start-up software company. I visited the company fairly late in the evening after having been out for dinner and to the home of my host for wonderful green tea, but even at 10 PM the Dynaware offices were crowded. Unfortunately, with the exception of the Dynaware designs, most of the user interfaces I saw in Japan were not very polished or visually attractive. This is especially strange considering that the country is famous for its high sense of aesthetics, even in small things. And the blandly looking designs can not be blamed on poor graphics capabilities either since everybody seemed to have big, high-resolution displays. My guess is that the emphasis in Japanese user interface design has been on accommodating their special character sets and language including heavy emphasis on natural language and knowledge based systems. If a system is good at dealing with Japanese and will do something useful, it will sell no matter how it looks. So probably the most important aspect of Japanese user interfaces is one which I am not qualified to judge. I was showed lots of menus and other interface elements in Kanji but can only report on one small point: In one system where the other commands were listed in Kanji, the undo command was still listed in English. I asked about this and was told that they had not been able to come up with a good translation of the concept, so they were perfectly happy with having that one word in European characters. My own experience confirms that it can be difficult to translate "undo. Except for the not very impressive graphic appearance of the user interfaces, there were several impressive aspects of the Japanese work in user interfaces. Advanced hardware is one area where the big Japanese electronics companies are doing extremely well, and I was shown impressive advances in optical media and also saw a nice hypermedia system using videodisks and a knowledge based interface. Another interesting system was a multi-media translation system which would scan a page from an English magazine and after an OCR phase translate the text to Japanese but keep the same page layout for the text in relation to the illustrations. Just the translation in itself had been a major project one person-year but had been parceled out to user interface professionals at several companies. The hypertext presentation system had been built somewhat faster using dBase III as an engine. Considering that at least two other hypertext systems has been built over the same information base, it would be interesting to perform a comparative experiment to look at the effectiveness of the different hypertext approaches. If such an experiment is to include the Japanese system, it must either be done entirely in Japan or as an intercontinental collaboration which would probably be a first in our field. Electronic Stationary Goods One of the more spectacular user interface ideas I was shown in Japan was the "electronic stationary goods" in the Tron project at the University of Tokyo managed by Ken Sakamura. This has of course been done in many other systems, but the Tron people went several steps further and also included other kinds of desktop items as computer peripherals. I was especially impressed by their electronic eraser which looked like an ordinary eraser used to erase pencil marks but which could be used on the computer tablet to delete marks on the computer screen. It was an interesting experience to use a paint program by physically changing between using a pen and an eraser instead of using a mouse to select a soft mode from a list of icons in a screen panel. Of course the weakness of this new approach may be that users would be confused by having the wide variety of tools used in some of the more modern paint programs as physical objects: Their office would end up looking like the studios of a graphics artist with lots of pens, knives, etc. But the machine itself is good enough with a nice display and they had designed some good user interfaces on it, using scanned color photographs as well as all the standard trappings of pop-ups, icons, etc. Of course there was a few things I would have done differently and I also saw one case of mixed navigational metaphors â€” but then I find that problem in almost half of the laboratories I visit: In general, the user interfaces I saw in Singapore looked good and also demonstrated interesting theoretical concepts such as generation of hypertext structures from a frame-based knowledge representation. The ISS-people gracefully invited me to their annual institute party which was a fun event with multilingual jokes and puns in English, Arabic, Tamil, and Mandarin as well as other Chinese dialects. There is no doubt that Singapore is one of the countries of the future and that the ISS can lead the way locally as well as contribute to the international user interface community. And since I had been embarrassed to have to say "no," I was glad to finally get a chance to see this machine after several years of rumors on the nets when I stopped over in the US on the way back to Europe. The feeling that this computer looked different from the ones I had been used to. It is amazing how much just the careful use of four level of grayscale can do to the visual appearance of computer screens. Dialog boxes have an almost 3-dimensional effect where input fields seem to be chiseled out. Things are still thrown away by dragging to the bottom right corner of the screen where instead of a trash can there is a strange icon which I could not understand. Upon asking I was told that it was a black hole. Cute â€” cute indeed â€” and it does avoid lawsuits. But the black hole it not the kind of user interface metaphor which your average neighborhood grocer is likely to relate to. The poor guy from Adobe who was going to use the NeXT for his demo of Display Postscript was swamped with requests for demonstrations of the NeXT interface itself rather than his rendering software. After all, for most people it is much more interesting what is written on the computer screen than how the pixels get turned black.

## Chapter 4 : Fifth generation computer - Infogalactic: the planetary knowledge core

*The Fifth Generation Computer Project is a two-part book consisting of the invited papers and the analysis. The invited papers examine various aspects of The Fifth Generation Computer Project.*

By Ruth Bednar Posted on July 22, The Fifth generation of computers Is the name of the stage that covers from to the present, a period where computers are fast and efficient and have a modern software developed. The electronic devices are small, thin and easy to carry, allowing the storage of multimedia information on solid devices and large memory cards. The name refers to an ambitious Japanese project carried out in the s by the Ministry of International Trade and Industry. The aim of this project was to develop a new type of computer capable of using artificial intelligence techniques and technologies, as well as implementing advanced hardware and software. The goal was to develop machines that were able to process natural language and achieve logical human-like abilities by solving complex coding problems. In April , Japan being the main pioneer, the fifth generation project was launched integrating a system composed of the use of artificial intelligence, parallel hardware and logic programming. A system that although for the time could not be fully developed, if it has been maintained until today, being improved and perfected more and more trying to design programs similar to the system of human reasoning. This fifth generation encompasses the use of not only artificial intelligence but robotics, expert systems, communications networks, nanotechnology, quantum computing and voice recognition. Thanks to fifth generation technology the world is getting closer and closer to the so-called"sixth generation"which will be based on artificial brains. The five generations of computers While today computers are part of everyday life being vital in this globalized world, its appearance and way of functioning was not always as it is today. For the generation from to , known as the first generation, computers were huge machines that used valves to operate and occupied large spaces. Magnetic cylinders were used to store the information and punched cards to exchange data, being implemented only in the military and scientific field. For the decade of to , one enters the second generation of computers. During this period, transistors are being used, which contributed significantly to the reduction of the size of the equipment. However, these continued to be slow and emanating enough heat. Then, between and , the third generation appears with the use of integrated circuits that miniaturized the electronic parts of the computers making them smaller and allowing the emergence of the first mini-computer. Then, from to , the Fourth generation of computers. In this generation the integration of the parts of the computers is maximized and the microprocessor appears, a small processor that improved the performance of the equipment making them faster and more efficient, maximizing the capacity to perform tasks and simultaneous programs. During this period the two most famous operating systems of the world, namely Macintosh and Windows, began to emerge from until today to the fifth generation of computers. The fifth generation of computers At this stage the PC emerges as we know it today, having its beginnings in the personal computer of IBM, International Business Machine, the recognized multinational company of technology of the United States; And the computer with a high-speed chip, developed in by Cal Tech, called"Touchstone Delta Supercomputer"which executed 8. It is in this generation that international competition comes to dominate the computer and technology market, and microcomputers and supercomputers are booming. In the s, laptops and notebooks began to emerge as well as the first mini-computers, and by the s the first smartphones or smartphones appeared on the market. In this generation computers can perform thousands of simultaneous operations, being able to perform a trillion operations per second. Computers have a higher speed, more miniaturization of the elements, and a considerable increase in memory capacity. In addition, other technologies have begun to form part of the market such as holography, optical fibers and biotechnology. Large companies such as Apple, Dell, HP and Samsung have created computers capable of performing more than a million million arithmetic operations per second. Main features of the fifth generation of computers Development of artificial intelligence Artificial intelligence has become an important branch of computing capable of providing hardware and software machines with elements that simulate human behavior, looking for computers to develop intelligent systems that allow them to find complex solutions to problems, in such a way Which the human brain does. The objective of the use of artificial intelligence in the fifth generation is that the reasoning, learning and interaction of the machines with the external world is achieved without these having to be programmed for it. Using Parallel Hardware Fifth-generation computers contain a large number of microprocessors that work in parallel, that is, they are capable of executing thousands of different instructions simultaneously. This has allowed computers and smartphones to have voice and image recognition. Application of heuristic programming It refers to the type of programming that has been implemented in the fifth generation, which gives the computer the ability to recognize patterns and processing sequences it has previously encountered. The idea is that the machine can remember previous results and include them in the process through the application of expert systems based on artificial intelligence seek to make the computer alone make a decision-making process. The five generations of computers. Retrieved on July 16, from btob. Retrieved on July 17, from culturacion. Retrieved on July 16, from cs. What is artificial intelligence? Retrieved on July 16, from computerworld. Generations of the computer. Retrieved on July 17, from informatica-hoy. Retrieved on July 16, from biblio3. Fifth generation of the computer. University of San Carlos of Guatemala. Retrieved on July 16, from academia. Background and applications of computer science. Retrieved on July 16, from leonelmartinez. A view of the fifth generation and its impact. Retrieved July 16, from pdfs.

## Chapter 5 : Fifth Generation Computing Conference Report: Article by Jakob Nielsen

*The Fifth Generation Computer Project is a two-part book consisting of the invited papers and the analysis. The invited papers examine various aspects of The Fifth Generation Computer Project. The analysis part assesses the major advances of the Fifth Generation Computer Pro.*

Domain-specific high-level programming languages such as SQL for database access and TeX for text formatting Background and design philosophy Throughout these multiple generations up to the s, Japan had largely been a follower in the computing arena, building computers following U. The Ministry of International Trade and Industry MITI decided to attempt to break out of this follow-the-leader pattern, and in the mids started looking, on a small scale, into the future of computing. They asked the Japan Information Processing Development Center JIPDEC to indicate a number of future directions, and in offered a three-year contract to carry out more in-depth studies along with industry and academia. It was during this period that the term "fifth-generation computer" started to be used. Prior to the s, MITI guidance had successes such as an improved steel industry, the creation of the oil supertanker , the automotive industry, consumer electronics, and computer memory. MITI decided that the future was going to be information technology. However, the Japanese language, in both written and spoken form, presented and still presents major obstacles for computers. These hurdles could not be taken lightly. So MITI held a conference and invited people around the world to help them. The primary fields for investigation from this initial project were: The aim was to build parallel computers for artificial intelligence applications using concurrent logic programming. The FGCS project and its vast findings contributed greatly to the development of the concurrent logic programming field. The chosen tool to implement this goal was logic programming. Logic programming approach as was characterized by Maarten Van Emden â€" one of its founders â€" as: The use of logic to present problems to a computer. The use of logical inference to solve these problems. More technically, it can be summed up in two equations: The Axioms typically used are universal axioms of a restricted form, called Horn-clauses or definite-clauses. The statement proved in a computation is an existential statement. The proof is constructive, and provides values for the existentially quantified variables: Logic programming was thought as something that unified various gradients of computer science software engineering , databases , computer architecture and artificial intelligence. It seemed that logic programming was the "missing link" between knowledge engineering and parallel computer architectures. In , during a visit to the ICOT, Ehud Shapiro invented Concurrent Prolog , a novel concurrent programming language that integrated logic programming and concurrent programming. Concurrent Prolog is a logic programming language designed for concurrent programming and parallel execution. It is a process oriented language , which embodies dataflow synchronization and guarded-command indeterminacy as its basic control mechanisms. It also inspired the concurrent logic programming language Guarded Horn Clauses GHC by Ueda, which was the basis of KL1, the programming language that was finally designed and implemented by the FGCS project as its core programming language. The project imagined a parallel processing computer running on top of massive databases as opposed to a traditional filesystem using a logic programming language to define and access the data. At the time typical workstation machines were capable of about k LIPS. In the government decided to go ahead with the project, and established the Institute for New Generation Computer Technology ICOT through joint investment with various Japanese computer companies. Implementation So ingrained was the belief that parallel computing was the future of all performance gains that the Fifth-Generation project generated a great deal of apprehension in the computer field. After having seen the Japanese take over the consumer electronics field during the s and apparently doing the same in the automotive world during the s, the Japanese in the s had a reputation for invincibility. The project also produced applications to run on these systems, such as the parallel database management system Kappa, the legal reasoning system HELIC-II , and the automated theorem prover MGTP , as well as applications to bioinformatics. The highly parallel computer architecture was eventually surpassed in speed by less specialized hardware for example, Sun workstations and Intel x86 machines. The project did produce a new generation of promising Japanese researchers. A primary problem was the choice of concurrent logic programming as the bridge between the parallel computer architecture and the use of logic as a knowledge representation and problem solving language for AI applications. This never happened cleanly; a number of languages were developed, all with their own limitations. In particular, the committed choice feature of concurrent constraint logic programming interfered with the logical semantics of the languages. The project also suffered from being on the wrong side of the technology curve. During its lifespan, GUIs became mainstream in computers; the internet enabled locally stored databases to become distributed; and even simple research projects provided better real-world results in data mining. The workstations had no appeal in a market where general purpose systems could now take over their job and even outrun them. This is parallel to the Lisp machine market, where rule-based systems such as CLIPS could run on general-purpose computers, making expensive Lisp machines unnecessary. In the early 21st century, many flavors of parallel computing began to proliferate, including multi-core architectures at the low-end and massively parallel processing at the high end. When clock speeds of CPUs began to move into the gigahertz range, CPU power dissipation and other problems became more important. Ordinary consumer machines and game consoles began to have parallel processors like the Intel Core , AMD K10 , and Cell microprocessor. Also in Concurrent Prolog: Includes pictures of prototype machines broken, but available at archive.

## Chapter 6 : What is the Fifth Generation of Computers? | Life Persona

*But I was still intrigued by the Fifth Generation computer project and expert systems. So, I bought the book from one of the Amazon Used and New sellers. I figured I might enjoy the story and learn something about expert systems at the same time.*

Domain-specific high-level programming languages such as SQL for database access and TeX for text formatting Throughout these multiple generations up to the s, Japan had largely been a follower in the computing arena, building computers following U. The Ministry of International Trade and Industry MITI decided to attempt to break out of this follow-the-leader pattern, and in the mids started looking, on a small scale, into the future of computing. They asked the Japan Information Processing Development Center JIPDEC to indicate a number of future directions, and in offered a three-year contract to carry out more in-depth studies along with industry and academia. It was during this period that the term "fifth-generation computer" started to be used. Prior to the s, MITI guidance had successes such as an improved steel industry, the creation of the oil supertanker , the automotive industry, consumer electronics, and computer memory. MITI decided that the future was going to be information technology. However, the Japanese language, in both written and spoken form, presented and still presents major obstacles for computers. These hurdles could not be taken lightly. So MITI held a conference and invited people around the world to help them. The primary fields for investigation from this initial project were: The aim was to build parallel computers for artificial intelligence applications using concurrent logic programming. The FGCS project and its vast findings contributed greatly to the development of the concurrent logic programming field. The chosen tool to implement this goal was logic programming. Logic programming approach as was characterized by Maarten Van Emden â€" one of its founders â€" as: The use of logic to present problems to a computer. The use of logical inference to solve these problems. More technically, it can be summed up in two equations: The Axioms typically used are universal axioms of a restricted form, called Horn-clauses or definite-clauses. The statement proved in a computation is an existential statement. The proof is constructive, and provides values for the existentially quantified variables: Logic programming was thought as something that unified various gradients of computer science software engineering , databases , computer architecture and artificial intelligence. It seemed that logic programming was the "missing link" between knowledge engineering and parallel computer architectures. In , during a visit to the ICOT, Ehud Shapiro invented Concurrent Prolog , a novel concurrent programming language that integrated logic programming and concurrent programming. Concurrent Prolog is a logic programming language designed for concurrent programming and parallel execution. It is a process oriented language , which embodies dataflow synchronization and guarded-command indeterminacy as its basic control mechanisms. It also inspired the concurrent logic programming language Guarded Horn Clauses GHC by Ueda, which was the basis of KL1, the programming language that was finally designed and implemented by the FGCS project as its core programming language. The project imagined a parallel processing computer running on top of massive databases as opposed to a traditional filesystem using a logic programming language to define and access the data. At the time typical workstation machines were capable of about k LIPS. In the government decided to go ahead with the project, and established the Institute for New Generation Computer Technology ICOT through joint investment with various Japanese computer companies. Implementation[ edit ] So ingrained was the belief that parallel computing was the future of all performance gains that the Fifth-Generation project generated a great deal of apprehension in the computer field. After having seen the Japanese take over the consumer electronics field during the s and apparently doing the same in the automotive world during the s, the Japanese in the s had a reputation for invincibility. The project also produced applications to run on these systems, such as the parallel database management system Kappa, the legal reasoning system HELIC-II , and the automated theorem prover MGTP , as well as applications to bioinformatics. The highly parallel computer architecture was eventually surpassed in speed by less specialized hardware for example, Sun workstations and Intel x86 machines. The project did produce a new generation of promising Japanese researchers. A primary problem was the choice of concurrent logic programming as the bridge between the parallel computer architecture and the use of logic as a knowledge representation and problem solving language for AI applications. This never happened cleanly; a number of languages were developed, all with their own limitations. In particular, the committed choice feature of concurrent constraint logic programming interfered with the logical semantics of the languages. The project also suffered from being on the wrong side of the technology curve. During its lifespan, GUIs became mainstream in computers; the internet enabled locally stored databases to become distributed; and even simple research projects provided better real-world results in data mining. The workstations had no appeal in a market where general purpose systems could now take over their job and even outrun them. This is parallel to the Lisp machine market, where rule-based systems such as CLIPS could run on general-purpose computers, making expensive Lisp machines unnecessary. It appears, however, that these new technologies reinvented rather than leveraged approaches investigated under the Fifth-Generation initiative. In the early 21st century, many flavors of parallel computing began to proliferate, including multi-core architectures at the low-end and massively parallel processing at the high end. Ordinary consumer machines and game consoles began to have parallel processors like the Intel Core , AMD K10 , and Cell microprocessor. Again, however, it is not clear that these developments were facilitated in any significant way by the Fifth-Generation project. In summary, a strong case can be made that the Fifth-Generation project was ahead of its time, but it is debatable whether this counters or justifies claims that it was a failure. Also in Concurrent Prolog: Archived from the original PDF on 12 February  Includes pictures of prototype machines broken, but available at archive.

## Chapter 7 : The Five Generations of Computers - Webopedia Reference

*Knowledge information processing forming the main part of applied artificial intelligence is expected to be one of the important fields in s information processing and the dedicated computers for this have been selected as the main theme of the national project of the Fifth Generation computers.*

## Chapter 8 : Fifth-generation jet fighter - Wikipedia

*Back in , Japan's Ministry of International Trade and Industry, begun the project "The Fifth Generation Computer Systems project".*

## Chapter 9 : Fifth generation computer - Wikipedia

*The Fifth Generation project, started in , aimed to develop computers with reasoning capabilities, rather than the ability to merely perform calculations.*