senter for teknologi og samfunn

STS

Jan Rune Holmevik

THE CONSTRUCTION OF SIMULA

The World's First Object Oriented Programming Language

STS-arbeidsnotat 1/92

ISSN 0802-3573-46

arbeidsnotat
working paper

senter for teknologi og samfunn
universitetet i trondheim
7055 dragvoll

tlf: +47 73 59 17 88
fax: +47 73 59 13 27

university of trondheim
centre for technology and society
n-7055 dragvoll, norway

The Construction of SIMULA

The World's First Object Oriented Programming Language [1]

Jan Rune Holmevik
Centre for Technology and Society
University of Trondheim

It has been implied that SIMULA is the most important single Norwegian contribution to international software development, and an important aspect of the paradigm of object oriented programming. From an historical point of view SIMULA can also be perceived as an important fragment of human, social and technological endeavour. In this perspective, social context and technical content are intertwined to such an extent that both become essential aspects in the analysis of the development process. This article surveys the development of SIMULA from the conception of the initial ideas, through the various controversies related to the construction of the language, and finally to the establishment of SIMULA as a black box safely concealed within the opaque framing of a computer. The question I address is simply how the research scientists managed to enroll various influential actors, capture new strongholds and thus constitute a network strong enough to withstand and overcome challenges imposed by other networks. In this respect the history of SIMULA reveals how social and political, as well as technical, elements impose their influence on the outcome of a scientific endeavour, and how the subject matter is not given à priori, but is subjected to continuous interpretations and modifications by the actors involved.

• Introduction

An extensive topic such as the SIMULA development cannot be adequately dealt with within the limits of this article. Consequently my objective has not been to give an exhaustive account of the history of SIMULA, but rather to concentrate on highlighting a few major events of significant impact to the development process. The broader scope of this article cover the period 1961 to 1975, focusing in particular on the formative years between 1962 and 1967. My main objective is to discuss how the language developed, how it was shaped and transformed throughout the period in question, incorporating the important transition from intellectual construct to physical artifact. In the last sections of the article I will further make an effort to shed some light on the dissemination of SIMULA, and also try to establish how this particular language has affected the development of more recent programming languages.

In this article I approach the SIMULA development from an historian's point of view. The historical perspective implies, among other things, that the socio-political context embedding the development of a technical artifact must be taken into account when attempts are made to reveal the secrets of history. A classic approach to the analysis of technological development has been to separate technology from society and read them as some sort of symmetrical constituted entities, or twin artifacts. The French anthropologist of science Bruno Latour argues that this notion, an artificial construct of the analysts' own interpretation, in fact obscures, and makes it difficult to grasp the realities of history.[2] In an attempt to render history more comprehensible, he puts forward an approach which aims at re-joining technology and society into a coherent whole, or as the American historian of technology Thomas P. Hughes neatly puts it; "a seamless web".[3] In Latour's approach it is asserted that techno-science can only be understood through it's practice, which in turn is principally considered as the building of networks. He claims that scientists constantly struggle to enroll, and keep in line, as many venues as possible in order to achieve their goals. The scientists' strive for support will eventually constitute a network of actors, human and non-human alike, tied to the artifact itself. The extent and strength of this network will, according to Latour, finally become the ultimate criterion for success or failure for any given scientific venture. In this survey I will employ a modified version of this actor-network approach as a guide line to the understanding of the SIMULA development.

Let us begin our short journey through the history of SIMULA with a brief overview of the setting. The Norwegian Computing Center (NCC) in Oslo was the heart of the SIMULA-development in the 1960s and 70s. This semi-governmental institution was first established in 1952 as a department at the Central Institute for Industrial Research (Norwegian abbreviation: SI). It's main purpose was to co-ordinate and supply Norwegian industry and various research institutions with computing power. Among NCC's other assignments during the fifties, was also the running of Norway's first digital computer called NUSSE. In 1958 a working committee appointed by the Norwegian Treasury Department decided that the Central Bureau of Statistics should acquire a DEUCE computer from English Electric. The intention was that NCC should be responsible for the running of this computer. In July 1958 NCC was therefore re-organised and became an independent institute under the Royal Norwegian Council for Scientific and Industrial Research (Norwegian abbreviation: NTNF).

Around 1960 it became apparent, however, that NCC was not capable of fulfilling it's obligations. The amount of work imposed on NCC increased rapidly, whereas the utilisation of the DEUCE computer did not live up to expectations. One important reason for this was the critical lack of qualified personnel.[4] It was evident that something had to be done to alter this unhappy situation, and as a first step in fortifying NCC, the board decided, during the spring of 1960, to restate the institute's objectives by inflicting a more research-like profile. This was the situation in May 1960 when Kristen Nygaard entered the stage.

He had previously been affiliated with the Norwegian Defence Research Establishment (NDRE). From 1952 onwards he had been among the leading operations research workers in Norway, and from 1958, head of the operations research (OR) department at NDRE. Toward the end of the fifties a controversy between Nygaard and NDRE's director Finn Lied arose over the way operations research should be conducted and developed. As this growing controversy made working conditions more and more difficult, Nygaard eventually decided to leave NDRE for a new post offered by NCC.[5] In general, Nygaard's new assignment was to build up NCC as a recognised research institute in operations research, computer science and related fields. To assist him in accomplishing this task he brought along six members of his previous OR-team from NDRE. The OR activities at NCC developed rapidly, and by the end of the year several large OR-jobs had been successfully accomplished.

Nearly a decade of OR experiences from NDRE and NCC showed that simulation (Monte Carlo manual simulation methods) was a very useful and efficient tool in analysing a wide range of complex real world systems. Furthermore, the extensive development in computer science, based on semi-conductor technology during the fifties, had opened up for a new generation of digital computers with very powerful computing capabilities. The development of computer programming languages,[6] starting off with FORTRAN i 1957, also made it quite evident that substantial new knowledge could be gained on a number of fields by utilising modern computer technology. These three elements, together with Nygaard's impressive ability to seize new opportunities and couple them with his own professional experiences from OR and computer science produced, during the spring of 1961, a set of powerful new ideas on how to describe, analyse and subsequently simulate complex real-world systems. In a strictly technical context this fragmentary and rather vague set of ideas was to mould the basis for the SIMULA concept.[7]

• SIMULA – a Language for the Description of Discrete Event Networks

From the very outset, back in 1952, Kristen Nygaard had been constantly concerned with questions related to the understanding of real world systems. The ability to fully understand a given system in all it's complexity, and further being able to precisely identify and analyse the operational qualities of the various interconnected entities constituting this system, was in his opinion, of vital importance for the outcome of any OR effort. In light of this the conception of SIMULA must accordingly be understood as a way of providing OR workers with a notation tool which would allow easy and precise description of standardised concepts, comprising all relevant aspects of a given system.

There might be many reasons why Nygaard decided to make this tool a programming language and not just an ordinary symbol notation system. I believe, however, that an important reason for this crucial choice stems from the extraordinary and highly fascinating development in programming language design from the mid 50s onwards. Around 1960 this field was, by leaps and bounds, constituting a new era of information processing which projected new and far-reaching perspectives on future computer utilisation, in scientific research as well as society in general. These emerging prospects undoubtedly had an important impact on Nygaard's reasoning in connection with the conception of SIMULA. His intention was to build the language around a general mathematical structure consisting of a few basic concepts. This would provide the researcher with a standardised approach suitable for description and simulation of practically any given OR problem.[8]

As we can see, it was evident from the very outset that SIMULA should simultaneously be a system description and a programming language. This implied that systems reasoning and programming skills were both needed in the making of such a language. Even though Nygaard had had some experience from computer programming at NDRE before he was assigned to OR in 1952, he had not the sufficient experience and knowledge to undertake an extensive task like this on his own. This meant that he had to recruit such expertise from outside, so he turned to one of his old acquaintances from NDRE, Ole-Johan Dahl. During the fifties Dahl had been assistant to Jan V. Garwick, a pioneer in Norwegian computer science, and held by many to be the father of informatics in Norway. When NDRE acquired a Ferranti MERCURY computer in 1954, Garwick and Dahl initiated work in language design and implementation, and by 1960 Dahl had become one of the leading computer scientists in Norway. Among other things Dahl was, at the time, working on an ALGOL like compiler called

MAC, and was probably, besides Garwick, the best professional ally Nygaard could get. During the spring of 1962 they joined in a series of discussions, which in May 1962 resulted in the first formal language proposal.

This early approach was, to a large extent, based on Nygaard's ideas of a mathematically formulated network-concept consisting of active stations serving a flow of passive customers. These stations consisted of a queue part and a service part, and the actions associated with the service part, were described by a sequence of formalised statements. The customers possessed no similar operating rules, but were instead described through a number of variables called characteristics. Such a customer was supposed to be generated by the service part of a given station, then transferred to the queue part of another station, and subsequently to the service part of that station. Here the customer was served, and then passed on to the next station in the network, and so on until it ultimately disappeared by not being transferred any further. The actions taken by the stations were regarded as instantaneous, occuring at discrete points in time, and accordingly this class of systems was called discrete event networks. [9]

As I mentioned earlier, the intention was to build the language around a fairly general mathematical structure. Accordingly, a salient point at this stage was whether they should construct their own, or rather adapt their concepts to an already existing one ? Nygaard and Dahl realised early on that if SIMULA was to become a real programming language, and not just another academic "paper language", they would have to join forces with one of the dominant programming languages. In the early sixties ALGOL-60 [10] was the leading programming language in Europe. The elegant and powerful concepts of this language appealed to Nygaard and Dahl, and made it in their opinion the perfect match for SIMULA.[11] The crucial decision of linking SIMULA to ALGOL-60 was made during the spring of 1962, and should later prove to be both an enormous strength and a serious obstacle. I will return to these technical matters later on, for the time being let us just ascertain that ALGOL-60 came to constitute a decisive technical stronghold, and as such it must be regarded as an important element in the making of the SIMULA-network.[12]

• The Computer Question.

Let us now, for a moment, turn our attention to NCC, and the difficult situation in the early sixties. By 1961 it was clear that the DEUCE engagement had a decisive negative impact on NCC's financial situation. Notable deficiencies had been recorded both in 1959 and 1960,[13] and the prospects for 1961 did not indicate any immediate improvement to this situation. One important explanation for this unfortunate development is the simple fact that NCC never quite managed to master the DEUCE computer. Technical problems combined with a general lack of experience and proficiency related to this specific machine, proved to have a severe impact on NCC's ability to execute it's customer commissions.[14] It must be added, however, that DEUCE was in many ways an extraordinarily tricky and complicated machine which differed radically from the computers familiar to NCC's staff of programmers, for example the Ferranti MERCURY at NDRE. This notion is justified by the fact that the Central Bureau of Statistics, which mainly used DEUCE as a regular punch-card machine, experienced no similar problems.[15] The DEUCE experiences had first of all demonstrated the necessity of a highly qualified and professional staff, and the introduction of OR early in 1960 had proved to be a step in the right direction. Furthermore it had also revealed NCC's profound need for a new computer which would improve the institute's ability to fulfil it's mandate.

In April 1961 NCC received an informal proposal from Dansk Regnecentral in Copenhagen regarding a possible future co-operation.[16] At the time, Regnecentralen's director Nils Ivar Beck, had ideas for a large network of Scandinavian computing centres called Scandinavian Electronic System. According to the Danish proposition, NCC could, within a few years, become part of this network and would in the short run benefit from such a co-operation in several ways. In other words, the Danish proposition seemed to offer the ideal solution to NCC's most immediate problems, and after a few preliminary meetings during 1961, informal relations between the two computing centers were thus established. From Nygaard's point of view it was therefore tacitly understood that when he and NCC's director, Bjørn Ørjansen was given the task of drawing up a report on NCC's immediate computer needs, they should conclude by recommending a GIER from Dansk Regnecentral.[17] GIER was a recognised medium size computer, but in Nygaard and Ørjansen's opinion not the ideal solution neither for NCC, nor for Norwegian Computer Science on the whole. What they wanted was a real Mainframe like English Electric's KDF-9. However, this computer was, at the time, far beyond NCC's financial reach, and consequently they settled for the original GIER alternative. Based on the conclusions in the Nygaard & Ørjansen report, NTNF decided in February 1962 that NCC should order a GIER from Denmark, and granted 2 mill. NOK. ($280.112) for this purpose.[18]

This was roughly the situation when another important actor, Sperry Rand Univac, entered the stage towards the end of May 1962. In connection with the marketing of their brand-new computers UNIVAC III and UNIVAC 1107, the company arranged an Executive Tour to the United States for some

160 prospective European customers, and Kristen Nygaard was invited to participate on behalf of NCC.[19]

Now, at this point in our story let us briefly recapitulate the status on the SIMULA development. As mentioned earlier, the SIMULA concept had, by May 1962, reached a state of semi-matureness, and Nygaard and Dahl felt that they now had a presentable language concept at hand. The preliminary groundwork was done, and it was now time to seek out and enroll the financial sources. As reported by Nygaard and Dahl in 1981,[20] there was no initial enthusiasm for SIMULA in NCC's environment. That is, apart from the valuable support given by the board of NCC. The main objection was that there would be no use for a programming language like SIMULA, and if by any chance there was, such a language certainly existed already. Furthermore it was asserted that their ideas were not good enough, and that they in general lacked the competence needed to embark upon such an extensive project, which for these reasons would never be completed. Finally it was maintained that this kind of work should not be performed in small countries like Norway. From these statements it should be evident that gathering financial support within the NTNF system, would indeed be a difficult and protracted mission.

Fortunately though, other options existed, and in an attempt to reverse Univac's sales mission Kristen Nygaard decided to introduce SIMULA to the Americans. As soon as the Executive Tour arrived in New York, he contacted the Univac Europe representative James W. Nickitas and presented him with SIMULA and another NCC software project on linear programming, called the LP-package.[21] Nickitas found Nygaard's ideas interesting, and agreed to set up a meeting with a few influential representatives from Univac's software division. Present at this meeting was Univac's director of systems programming Robert Bemer, also known as the father of ASCII [22], and previously a key person at IBM. Bemer had been a sworn ALGOL-60 fan right from the beginning, and at one point while still at IBM, he had in fact tried to supersede FORTRAN by ALGOL-60. As he listened to Nygaard explaining his ideas for an ALGOL-based simulation language, he became more and more convinced that SIMULA's sophisticated simulation facilities would significantly benefit ALGOL-60 in it's struggle with FORTRAN. [23]

By the end of the meeting he announced that he was to chair a session at the IFIP-62 [24] World Congress in Munich, and that he very much wanted a presentation of SIMULA at this occasion. Nygaard immediately accepted this offer as he realised the significance of such an opportunity.[25] The presentation of SIMULA at the IFIP-conference first of all implied an important step toward consolidation of the language concepts. Furthermore, it offered a suitable setting for the introduction of SIMULA, and simultaneously provided Nygaard and Dahl with an important entry to the distinguished community of computer professionals. Even though Kristen Nygaard had managed to draw professional attention to SIMULA, the vital question of financial support was still unsettled. At this point, however, Univac started to move. In connection with the marketing of UNIVAC 1107 the company needed an 1107 demonstration site in Europe as soon as possible. In this regard, Kristen Nygaard must have made quite an impression on the Americans, and really succeeded in convincing them of NCC's professional qualities, because upon his return to Norway, Nickitas approached him with an informal proposal announcing that Univac was interested in establishing this demonstration site at NCC. There might also be another explanation for Univac's way of reasoning in this case which is closely linked to the question of how many potential partners the company actually had to choose from. There is reason to belive that countries which clearly had computer development aspirations for themselves, like France and the UK, would be rather reluctant when it came to the question of spending large amounts of money on technology import, instead of encouraging similar domestic development projects. In light of this, Univac's freedom of choice might actually have been limited to a few smaller countries like Norway. Anyway the 1107 installation should serve promotional purposes for Univac, and NCC would get a 50% discount on an eventual purchase. In return for this generous offer however, Univac wanted NCC to provide them with SIMULA and the LP-package.[26]

• A New and Powerful Ally

When Univac at this stage made their appearance, the conditions for the SIMULA development changed considerably. As previously explained, the prospects of attaining grants within the NTNF system were not at all promising. Now all of a sudden a new situation had arisen. Through their initiative Univac had altered the terms by linking the development of SIMULA to NCC's computer acquisition. It is worth noticing that Univac at the time knew that NCC had already ordered a GIER computer from Denmark.[27] In this regard their initiative must be understood as an attempt to subvert the GIER order by offering NCC a more favourable contract. However their strategy was far more subtle than this. In addition to the favourable economic conditions, it also implied inside collaboration. By claiming implementations of SIMULA and the LP-package in return for their computer offer, they clearly aimed at enlisting Kristen Nygaard as their ally and inside man in Norway.

From Nygaard's point of view Univac's initiative must really have had an appealing sound to it. First of all because UNIVAC 1107 would provide a far better environment for the development of a SIMULA compiler than GIER. Another aspect which must also have attracted Nygaard was Univac's conspicuous marketing position and world wide distribution network. Associated with a powerful computer like the 1107, SIMULA would be launched onto the world market as standard software in a scale that NCC would never have accomplished.[28] However, the most important aspect was probably the prospects of NCC taking on the developing costs related to the SIMULA project. In a broader context a number of other elements might also have imposed their influence on Nygaard's way of reasoning. Here however, I will settle for maybe the most momentous one, which also, to some extent, was interconnected with the SIMULA effort. From the preceding description it should be apparent that Kristen Nygaard was quite an ambitious man, and that his professional aspirations were largely linked to NCC's position as research institute in computer science. In this respect he must have judged the possibilities of attaining a computer like the 1107, in the early sixties among the ultimate solutions in high-tech computer technology, as a Godgiven opportunity to fortify Norwegian computer science, and undoubtedly his own professional prestige simultaneously. In any case, it was obvious that the American company would constitute a formidable stronghold for the SIMULA development, and that Kristen Nygaard and Univac had common interests in this case, even though they pursued different goals.

Back in Norway, Nygaard's mission was, accordingly, to canvass opinion for UNIVAC 1107 at NCC and within the NTNF system in general. Since the research council had already ordered the GIER computer from Denmark, the effort had to be aimed at converting their interests in this case. Towards midsummer of 1962, Luthar Harr, director of Univac Europe, Stig Wallstam, director of Univac Scandinavia, and James Nickitas came to Norway to announce their formal proposal. In connection with this, members of the board of NCC together with a few influential people within the research council was summoned to a meeting with the Univac representatives. At this meeting the Americans came up with an offer which conveyed that NCC could acquire a UNIVAC 1107 at 50% discount, figuring up to approximately 7,1 mill. NOK. ($991.400).[29]

During the discussions however, the Americans got the notion that the computer configuration in question was too large and thus too expensive. So in an attempt to make it more appealing to the Norwegians they subsequently decided to extend their offer. In the following discussions they revealed that they would be willing to offer NCC a software contract on SIMULA and the LP-package. This unexpected offer, seemingly an improvised attempt to entice NCC and Kristen Nygaard in particular, was actually an ace that they had been hiding up their sleeves all along in an attempt to get SIMULA for free. At the time, head of Univac Systems Programming, Robert Bemer, had a yearly budget of about $ 8 mill. out of which he could spend up to 5% on whatever project he found worth doing. [30] Since the SIMULA project, in his opinion, was very much worth doing, he subsequently notified Univac General Sale that he wanted a software contract on SIMULA, and authorised the necessary funding out of the 5% discretionary money that he had available. Anyway the American initiative implied a closer link between SIMULA's destiny and the outcome of Univac's sales mission, and for Kristen Nygaard this must undoubtedly have been a vital spur for further engagement.

The American offer had, a strict time-limit attached to it, and the deadline for acceptance was fixed to August 1. 1962. Later, this deadline was later postponed until October 1., and delivery stipulated to March 1963. In July NTNF's working committee decided to take the 1107 question under deliberation, and Kristen Nygaard was given the task of drawing up a report on the subject. In this report, Nygaard not surprisingly concluded that an eventual purchase of a UNIVAC 1107 could be justified. He argued that the needs for computing power in research, public agencies, private commerce and various industries was rapidly increasing, and that a computer like the 1107 would indeed cater for present as well as future needs. Another asset was of, course, the fact that Univac's offer was approximately 2.5 mill NOK. ($350.140) less than the competitors' (IBM and English Electric) quotations.[31] In September the working committee handled the report, and after serious rounds of discussions they finally concluded by recommending that NTNF should acquire a UNIVAC 1107, and that the GIER order should thus be cancelled.[32] In August 1963 the 1107 arrived in Oslo. After a few weeks of assembling and testing, the computer was finally in operative condition, approximately four months after schedule.[33]

• The Department for Special Projects

The SIMULA-project now finally seemed to be underway. The vital financial question was at last settled, and the technical premises fairly well clarified. However, despite of these promising conditions it should still take close to a year before the SIMULA development really took off. The reason for this seemingly unexpected delay is partly of technical, partly of political origin, and clearly shows how science and research is largely dependent on the external premises embedded in this kind of activity.

Ever since Univac's offer was first known to NTNF in June 1962, they had emphasised the fact that an engagement involving such a heavy investment would necessarily imply that the bulk of NCC's available resources had to be directed strictly towards the business side of the institute's activity.[34] Since NCC operated largely on a commissionary basis, this meant that activities constituting sources of income had to be given top priority. In this respect it can be asserted that the UNIVAC 1107 came to represent a double-edged sword, at least as far as basic research activities were concerned. It is somewhat difficult to establish exactly how this situation might have affected the SIMULA development, since this project, as we know, was financed by Univac. It is evident, however, that the 1107 engagement restricted research latitude in general, and gave way for a professional profile which, at least to a certain extent, resembled the situation before 1960. For Kristen Nygaard, who in December 1962 was appointed NCC's director of research, this outlook must have been most disquieting, and he obviously could not accept that research should be pushed into "a small corner".[35]

And so in an attempt to compromise between these diverging interests, and at the same time establish a more suitable, efficient and dynamic organisation, the board of directors decided to restructure the entire institute by dividing it into a number of independent departments. According to the board's resolution of December 11. 1962, these new departments was meant to engage in practical commissions as well as applied research on specific target areas within NCC's mandate.[36] The SIMULA project, however, represented a slight problem with regard to this new organisational structure. Since software development had not previously been an integrated part of NCC's activity, and moreover required highly specific professional expertise, it must have been somewhat difficult to ascertain under which department it actually belonged. It was, so to speak, a disturbing element inflicted on NCC as a result of Kristen Nygaard's entrepreneurial activity in connection with the UNIVAC deal.

The dimension of conflict between interests also embodies another important aspect which must have contributed to the board's decision to reorganise the institute. When NTNF decided to go for UNIVAC this implied, as we have seen, that one had to attach greater importance to economic reasoning. This responsibility rested first and foremost with NCC's director, Bjørn Ørjansen, but it also applied to the rest of the staff. As we know, Kristen Nygaard did not quite seem share this opinion, and accordingly he did what he could to prevent research from being curtailed of recourses. For NCC's director, Nygaard's activity created a most difficult administrative situation, and for various reasons which shall not be commented on here, untenable social conditions within NCC developed.[37]

It might therefore have been a matter of necessity when the board subsequently decided to establish a Department for Special Projects and put Nygaard in charge as director of research. In this way they could keep him occupied, and prevent him from interfering with administrative matters, and simultaneously provide a suitable forum for software development.[38] The Department for Special Projects came to constitute a precedent for software development at NCC, which undoubtedly have been of major significance to NCC's later SIMULA engagements. SIMULA had captured another important stronghold.

• The ALGOL Connection and the Development of SIMULA I

After this rather sweeping detour let us now return to the technical matters. When Kristen Nygaard and Ole-Johan Dahl started out, during the spring of 1962, they had a rather vague set of ideas for a programming language which should meet a broad set of specifications. If we compare these initial ideas with the actual outcome of their scientific endeavour, the SIMULA I compiler, we find that there is a rather distinct difference between these two positions. In the following, I will make an effort to point out a few reasons for this change of goals.

As we now know, the early approach to SIMULA was based on an idea of a mathematically formulated network concept associated with ALGOL-60. [39] In general, Nygaard and Dahl's idea was to implement SIMULA as a simulation procedure package along with a preprosessor to ALGOL-60.[40] The preprosessor idea implied that a given SIMULA-program first had to be translated to ALGOL and then in turn, compiled into an executable program. In other words, this meant that a SIMULA program had to operate strictly within the framework of ALGOL-60, and as we shall see, this proved to be a serious obstacle, especially when simulation aspects were involved. However, at this early stage their reasoning was mainly preoccupied with the idea that customers in a simulation model could be depicted as ALGOL blocks, and furthermore, characterised using local variables. At that time, this idea looked rather promising since ALGOL's recursive block mechanism did cater for multiple occurrences of user defined data structures. [41]

By the spring of 1963 however, Dahl's work on the storage management scheme, for one thing, made it quite evident that ALGOL's block structure and strict dynamic single stack regime were in-compatible with an adequate implementation of SIMULA's sophisticated simulation facilities. In short, the problem facing Nygaard and Dahl at this stage was that ALGOL-60's procedure calls and storage allocation mechanisms operated strictly according to a stack principle, whereas objects (customers) in a simulation model rather tended to behave according to the queue principle.[42] In light of this, they subsequently realised that they would not achieve their design objectives unless they found a way to get around ALGOL's rigorous stack regime.

During the summer and autumn of 1963, while Kristen Nygaard was preoccupied fighting off problems on the political arena, Ole-Johan Dahl commenced work on a new storage allocation scheme based on a two-dimensional free area list. [43] With this new scheme at hand, they found that they were no longer tied by the restrictions imposed by ALGOL-60, and having this new freedom of choice they eventually decided to drop the preprosessor idea completely. Instead they decided to implement SIMULA through a modification and extension of Univac's ALGOL-60 compiler. This change of strategy opened a whole new set of perspectives on SIMULA, and accordingly, they were compelled to start over again. This time by deriving the basic concepts trough a variety of thorough case studies, ranging from job shops via airport departure systems to the dispersion of epidemics in a population.

With regard to the original network concept, they eventually discovered that this could just as well be regarded the other way around i.e. active customers making use of passive stations, which in turn lead to the realisation that an in-between, or dual point of view could profitably be adopted. From this perspective the customers was regarded as active in moving from station to station, but passive in their interaction with the service parts of the various stations. As a result of this vital detection, the joint activity within the system itself now became the one general principle applying to wide classes of systems. In light of this new understanding they found that the simple network concept seemed too narrow and thus inappropriate, and for these reasons it was subsequently abandoned. Instead

Nygaard and Dahl introduced the far more powerful process concept which came to constitute the basic, unifying feature of the SIMULA I language. In short, a process can be understood as a generalised ALGOL procedure with quasi-paralell properties. [44] This decisive breakthrough, in February 1964, implied that the simple notion of a system being described by a general mathematical structure had been replaced by a far more applicable comprehension. A system was now understood as consisting of a series of interacting processes operating in quasi-paralell as ALGOL stacks within the main program execution. [45]

By March 1964 the design phase had finally come to an end. It was now time to translate the paper version of SIMULA into an operating compiler. The implementation effort was solely conducted by Ole-Johan Dahl. On specific ALGOL related items he had, however, some assistance from the two American software engineers Ken Jones and Joeseph Speroni, the latter being the responsible for UNIVAC's ALGOL-60 compiler developed at CASE University in Cleveland. [46] The implementation effort proceeded throughout the year, and in December 1964, the first SIMULA I compiler was ready for acceptance by Univac.

• Renegotiating SIMULA

When Nygaard and Dahl first commenced work on SIMULA during the spring of 1962, their idea was to develop a simple simulation procedure package along with preprosessor on top of ALGOL-60. This seemingly straightforward approach turned out to be a dead end, when they ultimately ran up against the rigorous stack regime of ALGOL-60. As we have seen, the solution to this problem was found in Ole-Johan Dahl's storage management scheme from November 1963. From then on their ambitions took on a far greater amplitude, and in wake of Dahl's path-breaking new allocation scheme a turning point for the entire SIMULA effort can thus be traced. From the end of 1963, it is evident that they now strived toward general, unifying concepts in order to realise a real high-level programming language. In this respect, SIMULA I must be perceived as an intermediate position en route to what was later to become SIMULA 67.

In a previous section of this article, I have stated that ALGOL-60 came to constitute a vital technical stronghold with regard to the SIMULA I effort. In light of the events previously discussed, such a statement might seem a bit odd, since the connection to ALGOL was apparently weakened by the technical progress made during the autumn of 1963. The fact of the matter is however, that after this change of course, the ALGOL connection became more prominent than ever before. There are many reasons for this which unfortunately cannot be adequately examined here. I will however emphasise one aspect in particular, which was connected to their design objectives, and that is the question of program security and consistency, also denoted as orthogonality. In ALGOL-60, this was a deeply ingrained principle which in turn also came to have a decisive impact on SIMULA. In 1963 Nygaard and Dahl stated that the language should be problem oriented and not computer oriented, even if this implied an increase in the amount of work to be done by the computer. [47] Two years later, in 1965, they had come to the conclusion that the success of SIMULA would, regardless of their insistence, depend upon it's compile and run time efficiency, and for this reason the computer orientation had become a more prominent aspect. [48] However, this compromise must be considered a mere adjustment to the language, since the basic system description capabilities rooted in the orthogonal principle, in the end, resulted in simple, logical implementations which on average performed fairly well, and besides were very easy to maintain.

• The SIMULA 67 Common Base Language

During 1965 and 1966 Nygaard and Dahl spent a lot of time introducing and teaching SIMULA, and the use of the language rapidly spread to Sweden, Germany, the Soviet Union, and a number of other countries. Apart from the UNIVAC version, SIMULA also became available on Burroughs B5500 computers during 1968, and later on, the Russian URAL-16 computer. As we know however, Nygaard and Dahl's ambitions were of a greater amplitude than this. They knew that they now had a powerful and fairly generalised language concept at hand, which would make an excellent platform for a general purpose programming language. Furthermore, through utilisation of the language, they subsequently realised that a number of short-commings existed, for example with regard to the inspect statement for remote process attribute accessing, and an obvious lack of serviceable tools for expressing common properties between related processes in the system. Moreover it was obvious that the sophisticated simulation facilities embedded in SIMULA I was too heavy a burden to carry for a programming language with general purpose ambitions, and finally they had become aware of certain serious deficiencies in the UNIVAC ALGOL-60 compiler itself. [49]

In the autumn of 1965 the Norwegian Institute of Technology in Trondheim contacted NCC and expressed it's interest in implementing a new and improved ALGOL-60 compiler for the 1107, designed especially with SIMULA in mind. From Nygaard and Dahl's point of view this sounded like a promising suggestion, and for some time during 1966 professional relations with a computing team headed by Knut Skog in Trondheim was maintained.

As I have previously discussed, Nygaard and Dahl had since late 1963/ early 1964, been constantly in search of general unifying concepts. As this pursuit proceeded throughout the summer and autumn of 1966 they became more and more preoccupied with the opportunities embedded in Tony Hoare's record class construct, first presented in ALGOL bulletin no. 21, 1965. After having carefully examined Hoare's record proposal they eventually came to the conclusion that, even though it obviously had a number of very useful properties, it failed to fully meet their requirements. What they were really looking for was some kind of generalised process concept with record class properties.[50]

The answer to their problem suddenly appeared in December 1966, when the idea of prefixing was introduced. A process, later called an object, could now be regarded as consisting of two layers: A prefix layer containing references to it's predecessor and successor along with a number of other properties, and a main layer containing the attributes of the object in question. In addition to this important new feature, they also introduced the class concept, which can roughly be described as a highly refined version of SIMULA I's activity concept. This powerful new concept made it possible to establish class and subclass hierarchies of concatenated objects. As an example of this we can imagine the class Car, which can be apprehended as a generalisation of the subclasses Bus and Truck. In other words, the basic concept in human language of speaking in general and more special terms had been adopted as a way of expressing reality in the context of a programming language. Having these general, unifying tools at hand, Nygaard and Dahl immediately decided, to commence design of a new, general high-level programming language in terms of which an improved SIMULA I could be expressed. [51]

As I have pointed out earlier, their motivation for embarking upon yet another extensive programming language project can roughly be regarded as a combination of high ambitions, and a certain degree of dissatisfaction with their existing software product, the SIMULA I compiler. In addition to these constituting reasons, so to speak, I belive there is yet another reason, which gradually must have imposed it's influence on their strive toward new concepts. As a result of a proposal made by Ole-Johan Dahl, at the time the Norwegian representative to IFIP Technical Committee 2 (on programming languages), it had been decided in the autumn of 1965, that an IFIP Working Conference on simulation languages should be held in Oslo in May 1967.

Seen in association with this, the SIMULA 67 effort might additionally be regarded as an important step in extending the SIMULA network, since acknowledgement from a professional forum like IFIP TC-2, undoubtedly would constitute a professional stronghold of major importance. In any case, when the conceptual breakthrough finally came by the turn of the year, they were determined to present a new and revised version of SIMULA at this IFIP Working Conference. [52]

Prior to the conference Nygaard and Dahl had been working around the clock to finish their Class and Subclass Declarations paper. Despite the short time available to them they managed to incorporate all the important new aspects, and thus this paper became in a sense the first formal definition of the new language. The important virtual concept was, however, not included in the original paper since it had emerged at a very late stage, after the papers were submitted and therefore had to be presented in a separate paper. The conference's response to SIMULA 67 was positive, and Nygaard and Dahl now had a distinct feeling that the project finally was on the right track.

Two weeks later, in June 1967, another important conference was organised. The purpose of this conference was mainly twofold, firstly to define a standard for the exchange of SIMULA programs between various implementations, called the Common Base Standard, and secondly to initiate implementation projects for the Control Data 3000 (upper and lower) series and the UNIVAC 1100 series. Once again Nygaard and Dahl came up with a number of new proposals. One of the things that they wanted to incorporate was a unification of the related notions Type and Class. The new proposal underwent serious discussions, but after having considered all the implications and difficulties involved the pragmatic approach prevailed, and implementors subsequently rejected it. This, however, did not signify that the idea as such was dead, and like the Sleeping Beauty it would eventually come to life again. (see last section of this article)

Items related to string handling and I/O had not been discussed in any of Nygaard and Dahl's many proposals to the conference. However, the implementors unanimously stressed the need of having these things incorporated and defined as part of the Common Base Definition. In order to secure high standardisation and portability, it was therefore decided to furnish SIMULA with these facilities. The responsibility for design and development was given to Bjørn Myhrhaug, an important actor in the SIMULA environment, and a close colleague of Nygaard and Dahl. The results of his work was accepted at the first meeting of the SIMULA Standards Group (SSG) in February 1968, after which SIMULA was formally frozen. SIMULA was now finally about to become an existing general purpose programming language.

• From Intellectual Construct to Physical Artifact – the SIMULA 67 Compilers

As I have underlined earlier, Nygaard and Dahl had great ambitions for SIMULA 67. They believed at the time that a large number of programming languages developed during the 1960s would disappear, and that only a dozen or so would make it through the next decade. FORTRAN and COBOL would survive due to large and strong user communities world-wide, and the heavy investments in training and software development related to these languages. On a lower scale of usage they envisioned that some ten other languages would remain in use by 1980, and it was their ambition that SIMULA 67 should be in this group.[53] In order to accomplish this goal it was quite clear that SIMULA had to be available on the important Mainframes which around 1970 meant the UNIVAC 1100 series, and the IBM 360 series of computers.

Since NCC already had an 1107 in-house, and since the institute felt a special obligation toward this particular system because of SIMULA I and it's user community, it was decided that Sigurd Kubosch and Ron Kerr should commence work on a new UNIVAC implementation.[54] Their first prototype was completed during 1968. It had many serious deficiencies and performed rather poorly. One year later, in the summer of 1969 a new, but restricted test version was released to Univac, St. Paul for evaluation purposes, and in October 1970 the first pre-release for external users was made. For various reasons, that I shall not comment on here, the Research Council decided, in September 1969, to sell the 1107, and hand over the profitable computing commissions, which that year alone amounted to approximately 1,2 mill. NOK ($167.832), [55] to Computas, a company owned the large and influential ship classification society Det Norske Veritas. [56] This dramatic event which could have threatened the entire project, both in terms of computer availability per se but also financially, proved in the long run to have quite a positive effect on the compiler performance. The new target machine, UNIVAC 1108, which, in many ways, represented state of the art around 1970 provided the user with one of the most modern and efficient operating systems available (EXEC 8), but perhaps more important, the new machine offered the user a lot more valuable memory space than the ancestor, UNIVAC 1107. The UNIVAC SIMULA compiler was released in 1971, and in part due to the new environment under which it was developed, it became one of the fastest SIMULA compilers ever made, comparable in speed to the NU-ALGOL compiler from the Norwegian Institute of Technology. [57]

Univac was rather mixed in it's reactions against the new SIMULA compiler. First of all, SIMULA I had been a useful but not very important part of their software repertoire, and they felt no market demand for a new and improved version. Secondly, they had spent a substantial amount of money on the SIMULA I development, and saw no reason why they should share SIMULA with other manufacturers. Nevertheless, in 1968 a long series of discussions between NCC and Univac concerning a possible transfer of rights to the compiler took place, but no contracts were ever signed. [58]

In the case of the IBM 360 compiler the situation was somewhat different. When Nygaard and his team in 1967/68 expressed their interest in developing SIMULA 67 compilers for IBM and UNIVAC the conditions for doing this was stated very clearly. Firstly, such projects could only be undertaken if sufficient external financing were at hand, and secondly, that one, in the course of 5 years, would be able to cover the total expenses by selling the compilers on a strictly commercial basis. [59] According to these preconditions Kristen Nygaard and the private consultant Harald Omdahl, former director of the Joint Computing Center of the four largest banks i Norway, started the tedious and difficult mission of putting together a consortium of firms which would be interested in investing money in the compiler development projects. The hunt for investors went on with little or no success until Nygaard eventually meet with representatives for the Swedish Research Institute for National Defence (Swedish abbreviation: FOA) in the summer of 1969. FOA had, for many years, been wanting to make more active use of SIMULA in connection with their research activities, and as they had recently purchased an IBM 360/675 computer they were naturally very interested in Nygaard's proposals regarding the development of a SIMULA 67 compiler for this particular system.[60] On the basis of this the Swedes subsequently agreed to support the IBM 360 project by funding the participation of the two Swedish software engineers, Lars Enderin and Stefan Arnborg. Apart from this, IBM also made an important contribution by giving NCC a total of 240 hours of computing time to be used in connection with the development and testing of the compiler.[61] After nearly three years of extensive work the IBM compiler was released to the public in May 1972. Bjørn Myhrhaug, co-author of SIMULA 67, had been in charge of the project. In addition to the two Swedes I've just mentioned, the team had further consisted of Graham Birtwistle who were responsible for the syntax analysis, Francis Stevenson who took care of the code generation part, Paul Wynn on the CMS modification, and last but not least Karel Babcicky, who were responsible for semantics processing, and who, during the second part of the 1970s, were responsible for the entire SIMULA activity at the Norwegian Computing Center. [62]

Even though the costs related to the two implementation projects has been estimated at approximately 15 man years each [63], and both were conducted at NCC under auspices of Bjørn Myhrhaug, they were very much different. The IBM team worked within a well supported and carefully planned project, whereas the UNIVAC team was much more loosely organised, worked their way with less external support, and over a longer period of time. However, both projects produced high quality compilers which have been used in excess world-wide for more than 20 years, and contributed significantly to the dissemination of the SIMULA network. [64]

In the previous sections I have tried to give a general outline of NCC's own compiler development projects. These two compilers, however, were not the first SIMULA compilers available. Prior to the Common Base Conference, Kristen Nygaard had managed to get Control Data Corporation (CDC), through important Norwegian customers, interested in SIMULA implementations for their 3000 (upper and lower) and 6000 series of computers. [65] In May 1967 contracts were signed, and in Paris a team directed by Jacques Newey commenced work on a SIMULA 67 implementation for the CDC 6000. This compiler was later refined by SHAPE [66] Technical Center in the Netherlands, and a new version for the CDC Cyber 70 series was developed in 1973/ 74 by NDRE. In Norway, a team from the University of Oslo, headed by Per Ofstad, carried out the CDC 3300 (lower series) implementation, while another Norwegian team from NDRE and the University of Oslo's Joint Computer Installation at Kjeller (KCIN), headed by Svein A. Øvergaard, handled the CDC 3600 (upper series) implementation. The two projects were organised as a joint enterprise under the auspices of Per Martin Kjeldsaas from KCIN, and both were completed during the spring of 1969. Some financial support was provided by Control Data Europe, which in return obtained marketing and distribution rights, whereas the maintenance responsibility for the respective compilers remained with the University and KCIN. [67]

In addition to these projects, compilers for CII 10070 and IRIS 80 were implemented by the French company Compagnie Internationale pour l'Informatique (CII). The two identical systems were released in 1972, and was, according to an exploitation contract, given free of charge to CII's customers. [68]

Another important SIMULA development took place in Sweden in the first half of the 1970s, this time the target machine was Digital Equipment's DEC-10, or PDP-10 as it was called at the time. [69] Around 1970 the Swedish Research Institute of National Defence (FOA) in Stockholm had decided to establish a laboratory for advanced military studies in operations research. In connection with this they intended to purchase a new computer, and Jacob Palme, the most prominent SIMULA actor in Sweden at the time, was appointed chairman of a committee which purpose was to draw up an evaluation report on the subject. In accordance with FOA's former policy, one of the committee's most central demands was that the new computer had to have a SIMULA 67 compiler. In case the computer in question failed to meet this requirement they simply added 1,5 mill. SKR (approximately $209.790) to the total costs, in order for such a compiler to be developed separately. This way they were sure to have a SIMULA compiler without having to commit themselves to specific computer systems that already had SIMULA in their software library. After having carefully examined four different computers, they eventually decided on a machine from Digital Equipment Corp, the DECsystem-10. Since this computer had no SIMULA compiler available, such an implementation was ordered from the Swedish software house ENEA-Data. The first test version of the compiler was available to programmers at the QZ data center in Stockholm, or via communication networks from September 1974, and the first public release took place in January 1975.

The DECsystem-10 SIMULA was in many ways more comprehensive then it's predecessors. It contained, among other things, on-line debugging facilities which allowed setting and re-setting of break points during program execution. [70] Besides, the compiler was especially designed for interactive use, and would soon set a new standard for the development of SIMULA compilers.

Apart from FOA, Digital Equipment Corp. also contributed to the project under the condition that the compiler should be distributed free of charge. This condition was accepted, with the result that the DECsystem-10 compiler came to have a major impact on the dissemination of SIMULA, especially in the United States. In August 1975, eight months after it was released, the compiler had been distributed to 28 installations, 22 of which were located in the United States and Canada. [71]

• Towards a Global Network

Following our guide line, the actor-network approach, we find that the diffusion of the SIMULA network can be roughly said to have taken three different bearings. Most prominent and important are of course the many SIMULA implementations I have just discussed. Secondly the publishing of supporting literature, marketing and individual perception, and thirdly the impact of the supporting organisations.

In the early seventies the main supplier of supporting literature was NCC and it's associates. The bulk of this early literature consisted of technical reports and literature devoted to the introduction of the language. [72] There is one such publication I would like to mention in particular, called SIMULA 67 Implementation Guide. [73] As the name implies, this was a document containing all relevant technical information required to undertake an implementation of SIMULA 67. This report was considered a commercial secret until it was released to the public in 1971, and sold as a part of NCC's consultancy contracts with external SIMULA 67 implementors. I would also like to mention two other books witch, in a significant way, has contributed to the extension of the SIMULA network, namely the textbook "SIMULA Begin" from 1973, by Graham Birtwistle and, the technical book "Structured Programming" by Dahl, Dijkstra, and Hoare from 1972.[74]

Marketing is yet another aspect which became more prominent toward the end of our period. In the early days promotion was mainly taken care of by the developers themselves whenever they attended a conference, or gave various introduction courses to other research scientists, either at NCC or abroad. But as the SIMULA network expanded, and the amounts of money invested in the project steadily grew, this situation had to be taken care of in a more professional manner. As mentioned in connection with NCC's own compiler development projects, the institute felt no immediate responsibility for funding the accomplishment of these projects, especially not the IBM project. The general opinion was that the average life time for a programming language like SIMULA was approximately five years. Further, if one took into consideration the costs of developing a compiler, not to mention the great commercial risks involved, NCC was naturally very reserved in this matter. All the same, when NCC, eventually agreed to take responsibility for the two compiler development projects, their conditions for doing this was that sufficient external financing could be arranged, and that the institute as such would not be put to expense. Based on this mutual understanding the two projects were started (UNIVAC in 1967 and IBM in 1969), and Nygaard began his hunt for investors. As we know, however, his effort proved unsuccessful, and when this eventually became clear to the parties involved, both projects had developed to a point beyond no return, and NCC had to carry the financial burden alone.

Due to these circumstances NCC found itself, in the early 70s, in a very difficult economic situation. The institute operated, as I have said before, largely on a commissionary basis, and could not rely on government funding in this case. From the management and board of directors' point of view, it was therefore necessary for the SIMULA activity to be self-supported, and for the compiler investments to show profits. [75] In connection with this, marketing became much more important, and the English technical consultant Robin Hills of R. Hills (Consultants) Ltd. was enrolled to execute this line of work. In 1973 he formally proposed the establishment of an Association of SIMULA Users, and in the following years he arranged a series of SIMULA courses and introductory seminars world-wide. [76]

NCC has often been criticised for it's pricing policy regarding the two SIMULA compilers. This critique has in some cases been justifiable, like for instance when Donald Knuth in 1973 was prevented from introducing SIMULA at Stanford University, partly because of NCC's unwillingness to reduce prices, and give it away free of charge to universities.[77] When discussing these matters I think one should also bear in mind the fact that NCC was neither a conventional software house, nor a fully financed governmental institution. In addition to the SIMULA operations, NCC was also committed to a number of other research activities which deserved equal attention and financial support. Nevertheless, by 1973 the SIMULA network had spread to such an extent that NCC had lost track of it, and despite the fact that this dissemination could have gained even more momentum if NCC had lowered it's prices, the language was estimated to be in regular use at more than 250 sites, mainly in Europe and Australia. [78]

Apart from NCC itself, other organisations and interest groups have also contributed the diffusion of the SIMULA network. I have already mentioned the SIMULA Standards Group founded after the Common Base Conference in 1967. A similar interest group called SIMULA Development Group was later formed in 1975. [79] The SIMULA Standards Group consisted of one representative from each of the SIMULA implementation teams, plus two representatives from the Norwegian Computing Center. An important part of their task was to guard the SIMULA 67 standard against the undesirable flourishing of dialects experienced in so many other languages. In this respect their effort must be regarded as quite successful since the SIMULA 67 remained very much on track throughout the period and indeed up to the very present. On the other hand, however, it seems equally clear that SSG, in some respects, appeared much to conservative when facing new proposals, like for instance in the controversies over the missing CASE-statement. [80]

However, the most significant of these organisations have undoubtedly been the Association of SIMULA Users (ASU), founded in Oslo in September 1973. This organisation, through annual conferences, work shops and newsletters, provided an international "forum of SIMULA Users for discussions and exchange of ideas on SIMULA applications". Further more the ASU should encourage the use of SIMULA, and "make recommendations to the SIMULA Standards group and it's members about developments of the SIMULA language, and support all efforts to keep SIMULA well defined and truly machine independent".[81] At the first ASU conference in Oslo the number of founding members were 43, one year later, in August 1974, this number had increased to a total of 172, and the SIMULA network now extended to more than 23 different countries around the world. [82]

• Beyond SIMULA

In the last section of this article I will take a short glimpse beyond SIMULA to see how this particular language has affected two other, more recent language development projects, and how SIMULA has contributed to the conception of the paradigm we know as object oriented programming.

After having been deeply involved in a major research project for Norwegian trade unions in the early 70s, Nygaard, in 1973, returned to software engineering. Starting from the SIMULA platform, he now wanted to derive a pure system description language in terms of which continuous real-world systems could be better comprehended. The new language was called DELTA [83], and one of it's most essential concepts was that objects existed in a true, physical parallel mode, and not in quasi parallel as was the case in SIMULA. The language was also intended as an experiment with alternative notations and concepts, aiming at achieving clearer and more logical system descriptions.[84] As opposed to SIMULA, the DELTA language was never intended to be implemented, and in 1975 the project was thus brought to a conclusion.

In the second half of the 1970s Kristen Nygaard embarked upon yet another language development project called BETA, this time in cooperation with research scientists from the University of Århus and the University of Ålborg in Denmark. The generating idea behind this project was the very same that was turned down by the Common Base Conference ten years earlier. Very roughly, one can say that the BETA project was an attempt to generalise and refine the notion of classes, records, types and procedures into one basic construct upon which a new programming language could eventually be designed and implemented.[85] The effort proved successful, and the BETA team [86] came up with a new construct called Super Pattern. Based on this highly general concept, a DELTA compiler was later developed at the University of Århus.

Since SIMULA was constructed in the 1960s, other programming languages have followed in it's path, and as the number of heirs and practitioners steadily grew a new paradigm known as object oriented programming (OOP) began to proliferate. The best way to illustrate the fundamentals of OOP is probably to compare it with the more conventional linear design found in languages like ALGOL, FORTRAN or COBOL. Programs written in these languages can be compared to an incredibly swift puppeteer who is running around jerking the strings on lifeless puppets, which in this case represents the data structures, and thus bringing them to life. SIMULA and other OOP languages, on the other hand, tries to give the spark of life to each and every puppet, and instead of playing the puppeteer operating each puppet in a sequential manner, they play all the puppets simultaneously. [87]

As I said, building on the basic principles in SIMULA, other OOP languages eventually started to appear. One of the most prominent of these is Smalltalk, developed by the American Alan Kay at Zerox PARC in the early 1970s.[88] Two other, fairly influential languages are Eiffel and Objective-C. But the language which undoubtedly has contributed the most to the dissemination of OOP is C++ by the Dane Bjarne Stroustrup. An important explanation for this can be found in the close relation to C.[89] C and C++ is today considered de facto standard in software engineering, and because of this the OOP paradigm have naturally gained considerable momentum in recent years as the number of practitioners coming from C has literally exploded.

• Conclusions

In the preceding sections I have made an effort to describe the development of SIMULA as the construction of a network. Applying a suitably modified actor-network approach, I have tried to show that the history of SIMULA cannot be adequately comprehended unless social and political, as well as technical aspects is taken into consideration. Within the broader scope of this article I have therefore deliberately stressed the circumstances related to the establishment, and early dissemination of the SIMULA network over the years 1962 to 1967.

Kristen Nygaard and Ole-Johan Dahl is often referred to as the fathers of SIMULA. This fatherhood, however, was not as direct as that of Athena from Zeus's head. The point I am trying to make, is that the development process, spanning from the conception of the initial ideas to SIMULA as a technical artifact, safely concealed within the opaque framing of a computer, has a lot more to it that just the intellectual aspects. As I have repeatedly tried to illustrate, their concern was at least twofold. Apart from their constant strive towards ultimate technical solutions, they also had to exert themselves in order to secure the necessary strongholds to substantiate their project. This demanding, but highly important mission implied, as we have seen, a determined enlisting of allies. As a result of their networking activity the number of elements tied to SIMULA gradually increased, and by the time the SIMULA I compiler was completed the network included important actors like Univac, NCC, ALGOL, and several others.

Now, is it possible to measure the strength and quality of this network ? Well, the simple fact that SIMULA I, despite it's imperfection and strong competitors, [90] actually became quite successful should be a hint as to the solidity and power of the network. If we shift our focus to SIMULA 67, we will observe an even stronger network. In addition to technical leading-edge solutions, the network had been considerably fortifyed by, among other things, the reputation of SIMULA I, and as we approach the end of the period in question, we find that the SIMULA network had expanded even further. In light of what I have just discussed, it should be evident that SIMULA's superior technical solutions alone was no guarantee for success. Without an encompassing network of actors, powerful enough to withstand, and overcome challenges imposed by other competing networks, SIMULA would probably not have gained the status it has today.

As a final remark I would like to add that, despite any effort, SIMULA never became as wide-spread and commonly used as languages like FORTRAN, COBOL, C or Pascal. For an historian, this of course leads to the inevitable question; Why ? The answer to this, and a number of other important questions related to the SIMULA effort in the 1970s and 80s, remains to be revealed.[91]

- **References**

[1] A previous version of this article was first presented by the author at the 18th. SIMULA Users' Conference in Noordwijkerhout, the Netherlands, 10.–12. June 1992. The article is based on empirical research related to my dissertation thesis in history at the University of Trondheim. A more comprehensive study on this subject will be presented in my forthcoming thesis on the history of the SIMULA development. I would like to thank my supervisors professor Per Maurseth at the Department of History, and professor Håkon With Andersen at the Centre for Technology and Society for their professional guidance and valuable advice along the way. I would also like to thank colleagues and fellow students at the University of Trondheim for their interest and most useful

comments on preliminary drafts. Furthermore I am indebted to Kristen Nygaard and Ole-Johan Dahl for their thorough and detailed description of "The Development of the SIMULA Languages" in *History of Programming Languages* ,Wexelblatt (ed), Academic Press: New York 1981. Further I would like to thank Simula a.s. in Oslo for providing me with PC-SIMULA, this has given me an outstanding opportunity to learn some of the basic features of SIMULA, and hopefully contributed to a better understanding of this remarkable language. And last, but not least, I would like to express my sincere gratitude to Mrs. Eileen Schreiner who generously have offered her time and effort on "debugging" preliminary drafts of this article. Not being a native English speaker, your help has been most valuable and much appreciated. Finally I would like to point out that any errors in writing or eventual misjudgements in this article of course are solely my own responsibility.

[2] Latour, B.: "The Prince for Machines as Well as for Machinations". In *Technology and Social Process*, B. Elliot (ed), Edinburgh: Edinburgh University Press 1988. pp. 20-44. See also Latour, B.: *Science in Action - How to Follow Scientists and Engineers Trough Society.* Milton Keynes: Open University Press 1987.

[3] The phrase "seamless web" means that technology and society constitutes an entangled, coherent system which cannot be understood unless all relevant aspects of the system itself is taken into consideration. Hughes, T.P.: "The Seamless Web: Technology, Science, Etcetera, Etcetera". In *Social Studies of Science*, SAGE Publications Ltd: London, Beverly Hills and New Dehli. Vol 16 1986, pp. 281-292. For further introduction to the constructivist approach see also: *The Social Construction of Technological Systems*, Wiebe, Hughes and Pinch (ed). Cambridge, Mass., MIT-Press, 1987.

[4] The NCC Archive: "Den fremtidige virksomhet ved Norsk Regnesentral", May 3. 1961.

[5] Interview with Kristen Nygaard, November 28. 1991.

[6] A comprehensive description of the programming language development can be found in *History of Programming Languages*, Wexelblatt (ed), Academic Press: New York 1981. This volume comprises a total of 13 major programming languages developed during the fifties and sixties, including languages like FORTRAN, COBOL, ALGOL, LISP, and SIMULA to mention but a few.

[7] Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), op.cit., p 440.

[8] Nygaard K: "A Status Report on SIMULA - A Language for the Description of Discrete-event Networks". In *Proceedings of the Third International Conference on Operational Research*, pp. 825-831. London: English Universities Press, 1963.

[9] Ibid.

[10] ALGOL-60 represents a formidable milestone in the history of programming languages. During the 60s and 70s this language was the prime source of most theoretical work related to compiler design and development.

[11] Interview with Ole Johan Dahl, November 7. 1991.

[12] In a pure Latourian mode of understanding ALGOL-60 would have been characterised as a non-human actor, and as such treated on equal terms with human actors. In this article I have chosen not to use this term since it can be somewhat difficult to handle especially when it comes to the question of intention. In this respect the word can be rather contradictory since an object of non-human nature obviously cannot act intentionally. It is, however, possible to escape this dilemma for example by regarding non-human actors as elements simultaneously constituting boundaries and opportunities for a given artifact.

[13] NCC's Financial Accounts 1959, 1960.

[14] The NTNF Archives: Note from the Board of NCC, November 10. 1960.

[15] Interview with Sverre Spurkland, November 27. 1991

[16] The NCC Archive: "Den fremtidige virksomhet ved Norsk Regnesentral", May 3. 1961.

[17] Interview with Kristen Nygaard, November 28. 1991.

[18] The NTNF Archives: "Forslag om anskaffelse av en elektronisk regnemaskin type UNIVAC-1107 til Norsk Regnesentral", innstilling fra rådets arbeidsutvalg. Fall 1962. Please note that all US$ figures in this article is based on US-Norwegian exchange rates listed in *Historical Statistics 1978* p. 513

[19] The NTNF Archives: Note from NTNF's director Robert Major. July 19. 1962.

[20] Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., p. 442.

[21] This project was based on a new and promising algorithm for linear programming, denoted parametric decent, developed by Sverre Spurkland at NCC. Even though the project was based on a very powerful concept, and later on was to receive substantial financial support from Univac, it was never completed.

[22] ASCII (American Standard Code for Information Interchange) is a standard computer text format in which each character is represented by seven bits.

[23] Telephone interview with Robert Bemer, Tuesday April 14. 1992.

[24] IFIP as an abbreviation for: International Federation for Information Processing.

[25] Nygaard K: "SIMULA, An Extension of ALGOL to the Description of Discrete Event Networks". In *Proceedings of the IFIP Congress, 1962*. Amsterdam: North-Holland Publ..

[26] Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., p. 455.

[27] Ibid., p. 454.

[28] Interview with Kristen Nygaard, November 28. 1991

[29] The NTNF Archives: Sperry Rand Univac's offer to NCC regarding UNIVAC 1107, June 26. 1962, and "Agreement made 1st. day of June by and between Sperry Rand Corporation (Univac) and the Royal Norwegian Council for scientific and industrial research". June 1. 1963

[30] Telephone interview with Robert Bemer, April 14. 1992, and letter from Robert Bemer to the author, July 29. 1992.

[31] The NTNF Archives: "Forslag om anskaffelse av en elektronisk regnemaskin type UNIVAC-1107 til Norsk Regnesentral", innstilling fra rådets arbeidsutvalg. Fall 1962.

[32] NTNF's working committee handled the UNIVAC case at meetings held on August 21. and September 12. 1963. The committee members were generally in favour of the conclusions in Nygaard's report, that is with a minor exception for NDRE's director Finn Lied. He pointed out that it would be an unfortunate development if NCC were to expand it's staff by recruiting expertise from other similar institutions. He was also concerned that the acquisition of a large and expensive computer like the 1107 would jeopardise NDRE's chances of getting a similar computer in the near future. However, despite his concern he did not choose to dissent in this case. As for the cancelling of the GIER order this proved to be rather more difficult than at first assumed. However, after a few rounds of negotiations during the spring of 1963, the problem was solved and NTNF's total losses amounted to merely 8000.- NOK ($1117). The contract between NTNF and Univac was signed on October 24. 1962. (Source: The NTNF Archives: Minutes from the working committee's meetings August 21., September 12. 1962, and February 19. 1963)

[33] When UNIVAC 1107 was ordered in October 1962 the software systems to accompany the computer was not yet ready. However, Univac ensured NTNF, through a penal clause in the contract, that these systems would be available in time of delivery. The original delivery date was set to March 1963, but for various reasons later to be postponed. When the computer eventually arrived, in August 1963, it became apparent that the software systems provided did not meet the standard promised by Univac. This situation did not improve considerably, and by March 1964 NCC had accumulated a claim on Univac amounting to approximately 1.8 mill. NOK ($250.000). However, progress was being made, and in June 1964 NCC finally considered the software situation to be satisfactory. By then NCC's standing claim on Univac's had become substantial, and important people within Univac's organisation found themselves in a most awkward position. So, in order to maintain on friendly terms with Univac, and to avoid unnecessary "head-rolling" within that company, NCC agreed to accept an upgrade of the hardware configuration instead of ready payment. In short, this meant that NCC got a very powerful computer configuration on exceedingly favourable terms. (Source: Note from NCC's director Leif K. Olaussen: "Oversikt over forholdet mellom Norsk Regnesentral og Univac angående levering av den elektroniske regnemaskinen UNIVAC 1107 og tilhørende programmeringssystemer", January 15. 1965, and interview with Robert Major, November 29. 1991)

[34] The NTNF Archives: Letter from NTNF's director Robert Major to NCC's director Bjørn Ørjansen, July 21. 1962

[35] The NTNF Archives: Note from Kristen Nygaard to the board of NCC, January 22. 1964

[36] The NTNF Archives: Resolution from the board of NCC: "Organisasjonsplan for Norsk Regnesentral", December 11. 1962, and interview with Karl Holberg, March 5. 1992.

[37] The conflict reached a climax in March 1963 when three employees was fired for, among other things, conspiracy against Ørjansen and Nygaard. Since the politics surrounding these events was rather complex I will not engage in further discussions here. As a result of this crisis however, Bjørn Ørjansen decided to resign from his position as director of NCC and was released by Leif K. Olaussen in December 1963. For Kristen Nygaard the heavy strain caused by the working conditions in wake of the crisis lead to a sick leave during the autumn of 1963. This difficult situation must also have contributed to restrain the SIMULA development.

[38] The NTNF Archives: Resolution from the board of NCC: "Organisasjonsplan for Norsk Regnesentral", December 11. 1962, and interviews with: Bjørn Ørjansen November 25., Sverre Spurkland November 27., Robert Major November 29. 1991, and Karl Holberg March 5. 1992. In the resolution it is stated that the Department for Special Projects should focus on important problems within NCC's mandate, being of such character that they could not be adequately treated by any of the other departments. As examples to this kind of activity the board puts forward; software development, research in new areas of computer application, special projects related to the development of NCC's professional profile, public relations and consulting services for important customers. As we can see, it can easily be asserted that this was a "playground" especially designed to suit Kristen Nygaard.

[39] It should be added, that even though Nygaard and Dahl at this stage asserted that SIMULA should be ALGOL-based, they opened for a later version based on FORTRAN, using the same basic concepts.

[40] Dahl O.-J. and Nygaard K: "Preliminary presentation of the SIMULA Language (as of May 18th. 1963) and some examples of network descriptions". NCC: May 1963.

[41] Nygaard K. and Dahl O.-J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., pp. 447-448, and interview with Ole-Johan Dahl November 7. 1991.

[42] The term "stack" is intended to conjure up visions of objects encountered in daily life, such as a stack of dirty dishes in your home kitchen. In computer science, a stack of objects has the property that the last object placed on the stack will be the first object removed. This property constitutes what is commonly referred to as a LIFO-structure, or in plain English, Last In, First Out. While the stack can be thought of as having only one end, a "queue" on the other hand is viewed as having two ends. This gives the queue a first in, first out behaviour (FIFO-structure) , which makes it appropriate for modelling and simulation of complex real-world systems.

[43] Dahl O.-J: "The SIMULA Storage Allocation Scheme". NCC Document no. 162, November 1963, and interview with Ole-Johan Dahl, November 7. 1991.

[44] Interviews with Ole-Johan Dahl, November 7. 1991 and March 4. 1992, and Kristen Nygaard, November 28. 1991

[45] Quasi-parallel process execution implied that a main program could switch control from one process to another during a program run, according to special sequencing statements. In SIMULA this sequencing was determined by the Hold-statement, in addition to local reference variables in each inactive process, identifying where the control should resume operation next time the process was activated. (Source: Dahl O.-J: "The SIMULA Data Structures", NCC Document, February/ March 1964.)

[46] Letter from Ole-Johan Dahl to the author, April 10. 1992

[47] Nygaard K: "A Status Report on SIMULA - A Language for the Description of Discrete-event Networks". In *Proceedings of the Third International Conference on Operational Research*, pp. 825-831. London: English Universities Press, 1963.

[48] Dahl O.-J., and Nygaard K: *SIMULA - A language for programming and description of discrete event systems. Introduction and user's manual.* NCC Publication no. 11 1965.

[49] Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., pp 459-461.

[50] Interview with Ole-Johan Dahl, March 4. 1992.

[51] Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., p 459-461.

[52] Dahl O.-J., and Nygaard K: "Class and subclass declarations". NCC Document, March 1967.

[53] Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., p. 471

[54] Interview Drude Berntsen, November 6. 1991.

55 NTNF's annual review 1969, p. 69

56 The running of the UNIVAC 1107 was, in the second half of the 1960s, one of NCC's most important sources of income, and a cornerstone in the institute's research activities. When the Research Council's decision was known this naturally gave rise to speculations concerning NCC's future, and rightfully so since one had strong reasons to belive that central forces within the Research Council wanted to shut down the entire institute, and transfer it's operations to other NTNF institutes. In a reaction to this the NCC employees organised themselves and launched a counter attack on NTNF. The conflict reached it's climax in January/ February 1970, after which NCC was reorganised and the situation returned to normal. Source: Interview with Sverre Spurkland, November 27. 1991.

57 Interview with Sigurd Kubosch, March 5. 1992, and SIMULA Newsletter: Vol 1, No. 1, May 1973

58 Interview with Drude Berntsen, November 6. 1991, and "Draft outline for a SIMULA-67 agreement between UNIVAC and the Norwegian Computing Center", September 1969.

59 Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., p. 470

60 Interview with Jacob Palme, August 19. 1992, and Jacob Palme: "Are Simulation Languages Convenient for Military Simulations ?". Research Institute of National Defence, Operations Research Center – Sweden, April 1968.

61 Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., p. 475

62 Karel Babcicky: "The Stone Age SIMULA". Talk presented at; "SIMULA 1967–1992", Special Workshop at TOOLS'92, Dortmund – Germany, March 31. 1992

63 Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., p. 475.

64 Karel Babcicky: "The Stone Age SIMULA". Talk presented at; "SIMULA 1967–1992", Special Workshop at TOOLS'92, Dortmund – Germany, March 31. 1992

65 NCC files: "Agreement on implementation of the SIMULA 67 language between Control Data A/S Norway and the Norwegian Computing Center". Oslo May 23. 1967.

66 SHAPE is an abbreviation for; Supreme Headquarters Allied Powers Europe.

67 Nygaard K. and Dahl O.J: "The Development of the SIMULA Languages", Wexelblatt (ed), Op.cit., pp. 473 - 474.

68 Drude Berntsen: "Welcoming Speech", in *Proceedings of the I. SIMULA Users' Conference*. Oslo, September 24. - 25. 1973.

69 The following account, describing the development of the Swedish DEC-10 SIMULA compiler is based upon an interview with Jacob Palme, August 19. 1992

70 *SIMULA Newsletter*, Vol. 2, No. 3, August 1974

71 *DEC System-10 SIMULA Gazette*, Vol. 1, No. 2, August 1975

72 "A SIMULA Bibliography", compiled by G. Birtwistle, B. Liblong and B. Wyvill. This bibliography comprises some 550 individual titles ranging from comprehensive books to articles, papers and technical reports. University of Chalgary, May 27. 1982.

73 Dahl O.-J. and Myhrhaug B: *SIMULA-67 Implementation Guide*, NCC Publication No. S-9, June 1969.

74 Birtwistle G.M., Dahl O.-J., Myhrhaug B., and Nygaard K: *SIMULA Begin*. Studentlitteratur, Lund, Sweden and Auerbach Publ.., Philadelphia, Pennsylvania, 1973. And Dahl O.-J., Dijkstra E.W., and Hoare C.A.R.: *Structured Programming*. New York: Academic Press, 1972.

75 Interview with Drude Berntsen: November 6. 1991.

76 *SIMULA Newsletter*, 1973 - 1974.

77 Kristen Nygaard: "Constructing SIMULA – Cooperation and War". Talk presented at; "SIMULA 1967–1992", Special Workshop at TOOLS'92, Dortmund – Germany, March 31. 1992, and interview with Kristen Nygaard: November 28. 1991.

78 Drude Berntsen: "Welcoming Speech", in *Proceedings of the I. SIMULA Users' Conference*. Oslo, September 24. - 25. 1973. Among the largest users known in 1973 we find; The Norwegian Defence Research Establishment, the Swedish Research Institute of National Defence, the Finnish State Computing Center and Broken Hill Proprietary Co. Ltd. in Australia. In addition these major institutions, a number of European universities were also known to use the language for educational purposes.

79 Babcicky K: "SIMULA-67", in *Encyclopedia of Computer Science and Technology*, Belzer, Holzman and Kent (ed). Marcel Dekker Inc. New York and Basel, 1980.

80 Interview with Sigurd Kubosch, March 5. 1992

81 *SIMULA Newsletter*: Vol. 1, No. 3, November 1973, p.16.

82 *SIMULA Newsletter:* Vol 2, No. 3, August 1974, p. 3. As of August 1974 the following countries were represented in ASU; Australia, Austria, Belgium, Czechoslovakia, Denmark, Finland, France, Germany, Hong Kong, Hungary, Israel, Italy, New Zealand, the Netherlands, Norway, Poland, South Africa, Spain, Switzerland, the United Kingdom and the United States. In addition to this one must also take into account the considerable SIMULA activity in the former Soviet Union, which in many ways constituted a network of it's own, separated from the organisational framework of the ASU.

83 DELTA was an acronym for the project name: Development of Language Tools for Administration and Research, and in addition to Nygaard himself, the team consisted of Petter Håndlykken and Erik Holbæk-Hanssen.

84 Kristen Nygaard: "Statusreport on the DELTA Project", in *SIMULA Newsletter* Vol. 2, No. 3, August 1974, and interview with Kristen Nygaard: November 28. 1991.

85 Birger Møller-Pedersen: "How did SIMULA make us Think of BETA". Talk presented at; "SIMULA 1967–1992", Special Workshop at TOOLS'92, Dortmund – Germany, March 31. 1992, and interview with Kristen Nygaard: November 28. 1991.

86 The BETA team consisted of Kristen Nygaard, Bent Brun Kristensen and Ole L. Madsen. The team was later joined by Birger Møller-Pedersen.

[87] The analogy is taken from Alan Kay, cited in "Alan Kay's Magical Mystery Tour – the Search for the Ultimate Personal Computer", in *TWA Ambassador*, January 1984.

[88] For a brief introduction to Smalltalk see; *Byte*, August 1981.

[89] "Historien om OOP". Conversation between Kristen Nygaard and Bjarne Stroustrup in *PC World Denmark*, 9/1991.

[90] In this article I might have created the impression that SIMULA-I was the first simulation programming language available. This, however, is not true. FORTRAN based simulation languages such as GPSS and SIMSCRIPT were already available prior to SIMULA-I, and in terms of usage SIMULA-I never gained the same popularity as these two languages.

[91] For Kristen Nygaard and Ole-Johan Dahl the SIMULA effort must be regarded as an important platform for their later work in informatics and computer science. As far as Kristen Nygaard is concerned he was in the early 70s engaged in research for Norwegian trade unions on planning, control, and data processing. He has also been preoccupied with social impact of computer technology, and the development of the system description language DELTA and the programming language BETA. Kristen Nygaard is today professor at the Norwegian Research Council for Science and the Humanities in Oslo. Ole-Johan Dahl left NCC in 1968, when he was appointed Norway's first professor in informatics and computer science at the University of Oslo. In the early seventies he was deeply involved in the development of the Structured Programming approach together with people like E.W. Dijkstra and C.A.R. Hoare. In the second part of the 1970s his research was, among other things, focused on the areas program architecture, specification tools, and verification techniques.