



**HAL**  
open science

## Operations Analysis Computing at FFI, 1970–1995

Norodd Hagenson

► **To cite this version:**

Norodd Hagenson. Operations Analysis Computing at FFI, 1970–1995. 3rd History of Nordic Computing (HiNC), Oct 2010, Stockholm, Sweden. pp.35-42, 10.1007/978-3-642-23315-9\_5 . hal-01564641

**HAL Id: hal-01564641**

**<https://inria.hal.science/hal-01564641>**

Submitted on 19 Jul 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Operations Analysis Computing at FFI, 1970–1995

Norodd Hagenson

Forsvarets forskningsinstitutt, P. O. Box 25, N-2027 Kjeller, Norway  
norodd.hagenson@ffi.no

**Abstract.** By initially presenting FFI and its operations analysis (OA) activity, this paper illustrates the use of OA computing at FFI in three ways, in defense analysis, Simula, and various analyses of air defense. The paper focuses on methods, particularly the use of discrete event simulation. Most examples are from 1978–1995, and Simula’s role is highlighted.

**Keywords:** Defense, ground-based air defense, history, long-term planning, operations analysis, Simula, simulation

## 1 FFI and Operations Analysis

The Defense Research Institute (Forsvarets forskningsinstitutt, FFI) was established 1946 and is Norway’s major defense research and development organization [1] with a staff of 716 and an annual turnover of 754 MNOK (2009) [2]. FFI not only provides scientific and technological advice to the Norwegian Ministry of Defense and the armed forces, but also carries out research, development, and analysis in support of the structures of the armed forces. In addition, it undertakes weapons and equipment development as a basis for a competitive national defense industry.

FFI established its Systems Analysis Group in 1959 [1]. The main idea was to use operations analysis (OA) for defense planning purposes, but a further notion was that emerging computer power could play a role in our OA activity. The group later became the Division for Systems Analysis and eventually the Analysis Division. It has focused on long-term defense planning, using operations analysis that includes simulation, which is used to experiment with new systems in a future environment.

Various kinds of computing power were employed for OA since the group was establishment in 1959. Analyses have ranged from weapons testing through weapon system evaluation and tactical employment to analyses of defense service structures and the full defense structure. Service and defense wide analyses were performed for the first time in the 1970s. Such high-level analyses have had a major impact on defense planning and defense structuring in Norway. Delivering relevant advice on time was of paramount importance; analyzing the wrong problem or providing a good answer too late was of little use. User involvement was another key to success.

## 2 Defense Analyses

Having successfully used OA for specific problems such as weapon system development in the 1960s, FFI's Systems Analysis Group saw an opportunity to influence major and important decisions. The Royal Norwegian Air Force was going to acquire new combat aircraft. FFI's first air force analysis was carried out 1970–1974 [1, 3, 4] and addressed not only the question of aircraft selection posed by the Air Force, but also important issues which the customer did not ask, for example, what support was needed in terms of airbases and aircraft service capacity. A discrete event Simula-model, the Base Model [5], was used in this analysis, simulating combat aircraft sortie production.

The navy and army were also analyzed, as were the entire Norwegian armed forces from 1974 onwards [1, 3]. System dynamics (SD), the main tool in the latter project, models dynamic effects in systems with feedback and is often applied to, but not restricted to economic and social systems. SD really describes a system by coupled linear differential equations. In the armed forces' study, SD modeled their flow and attrition, and was later used at FFI in other contexts. Our experience of SD has been good. High-level studies of defense structures were repeated in the 1980s and 1990s, and they are now a continuous activity.

The cost of defense structure is an important example of continuous computing activity. While the first version of the main model, BUDSJ [3, 6], was FORTRAN based, today's version, KOSTMOD [3, 7], is a database. Involving the customer is one of the first commandments in OA. A couple of attempts to transfer data collection and cost analysis to defense headquarters were made during the 1980s. However, the coming and going of officers in key positions made the task difficult and it was never completely successful. These attempts clearly demonstrated the necessity of dealing with a community that has sufficient continuity to maintain knowledge, methods, and data. Thus, the estimation and analysis of defense costs have become permanent activities at FFI.

## 3 The Use of Simula at FFI

Simula is considered to be the world's first object-oriented programming language. It was developed by Kristen Nygaard and Ole-Johan Dahl at the Norwegian Computing Center in the 1960s [8]. Both had recognized the need for more powerful programming languages when they worked at FFI in the 1950s. Simula, based on Algol, had added important concepts such as classes and objects, subclasses, inheritance, and virtual procedures. System classes are used to implement connected lists and events, enabling discrete event simulation by quasi-parallel processes. The clean concepts in Simula will easily leave a lasting impression on programmers' way of thinking.

Simula was an important programming language at FFI. RBK (Regneanlegget Blindern-Kjeller) was established and obtained its Cyber-74 computer in 1972 [9]. A Simula compiler was included, but its quality was not satisfactory. FFI therefore developed a new Simula compiler for Cyber-74, starting in 1973 [10]. This compiler

was marketed and sold by CDC (Control Data Corporation, the maker of Cyber-74), but it was owned and maintained by FFI. The quality of this compiler was good and it was able to handle large programs, even featuring the separate compilation of program modules. The only major drawback was its inability to nest SIMSET- and SIMULATION-blocks while maintaining correct qualification for variables belonging to the outer SIMSET-block, when seen from the inner SIMULATION-block. However, this flaw was easy to circumvent with a few auxiliary functions. Nevertheless, the defect appeared to be common in other Simula implementations as well.

Simulation with Simula represented a radically new and different approach to computing. Formerly, a program was a sequence of statements often described by a flow chart, but by using Class Simulation, a program could now consist of several separate co-routines that would execute in some order that was not always directly predictable, allowing for complex interaction between processes.

During the 1980s, in-house minicomputers from Norsk Data, with Tandberg terminals, partially took over programming and computing. Simula became available on ND computers in the 1980s. Lund Simula was used on UNIX workstations at FFI until 2001.

The use of Simula at FFI gradually dwindled. In the 1980s and 1990s, other languages, such as Simscript and Modsim II, were also employed for simulation model development. Simula was still used for models that had initially been developed in Simula and were being reused and further developed. Simscript was employed for a US model. Modsim II replaced Simula for the new development of discrete event simulation models. Simula's role at FFI is further illustrated through the following examples.

## **4 Application of Simula for Analyses of Air Defense Systems**

Several projects have analyzed ground-based air defense, of which some serve as examples here. A number of important studies using Simula were omitted, however, such as air defense of maritime patrol boats.

A model is a catalyst that stimulates thinking. Modeling problems usually provide better understanding than modeling systems. A model is an abstraction that should include the key factors, but not more details than necessary.

### **4.1 The Air Defense Model**

The Air Defense Model (AD-model) [11], simulating ground-based air defense, was the longest-lasting Simula model at FFI; it was used between 1978 and 2001. During this period, it was reused, expanded, and adapted to new problems and new weapon systems several times. Initially, the model ran on Cyber-74 with the FFI Simula compiler, but the compiler needed an update as the AD-model grew in size.

The AD-model is a closed, discrete event Monte Carlo model that simulates several ground-based fire units engaging several airborne targets in a digital terrain. It

would typically execute hundred replications in a run. The model simulates fire units sitting in fixed positions engaging targets moving in 3D over a digital terrain. The airborne targets fly along predetermined tracks, while the ground-based air defense is modeled with considerably more detail. Some important features are reaction times, weapon ranges, and velocities. Officers were involved in establishing scenarios and input.

The Air Defense Model utilized almost every feature in Simula. Cancelling processes was the only feature that was used for a short time and then abandoned. An experience with a bug that caused a process to cancel itself taught us a lesson. It led to a strange situation where processes were in disarray and difficult to debug with the available tools. When cancelling a process, it may be necessary to keep close track of each procedure, including its state. A more elegant and manageable approach to the programming of process interaction was to reactivate a process when there was an event which might influence it, let the process analyze the situation as it was being perceived, and then allow the process itself decide what to do. A useful way of thinking is to assume that it is not necessary to know very much about how a program reached a certain state, and to just handle the present state regardless of its history.

Simula facilitated a structure where new weapon systems could easily be added, using the powerful concept of virtual procedures to model weapon system specific features. From the start, it used digital terrain initially based on in-house manually digitized iso height curves. It was soon apparent that a grid-based terrain database was preferable, and the change was implemented, initially using terrain from the Defense Mapping Agency and later from Norges Geografiske Oppmåling, now the Norwegian Mapping Authority (Statens Kartverk).

Virtual procedures were particularly useful and included generic classes for a fire unit, a missile, and a gun. Every class related to weapons was derived from such generic classes and used virtual procedures for specific properties, for example, 3D weapon range or the random drawing of a reaction time. The concept was flexible enough to allow the easy addition of new weapon systems, once their properties were specified. Even new concepts could be accommodated. Without the adaptability and growth potential of virtual procedures, the AD-model might have been short-lived, perhaps only surviving through its first project.

To a computer, a program is blind calculation. A human must always interpret the results and the meaningfulness of results must be checked. On one occasion, the digital terrain played us a trick. Through an error, a flight track had the incorrect height and passed through a mountain. The AD-model did not check for terrain collision, but faithfully represented the aircraft as obscured by terrain where its flight track passed through the mountain. The mistake was easily spotted and corrected, and gave us a good laugh.

The AD-model was born when Hollerith cards were still being used. Despite the jokes and comments about old-fashioned technology, it was actually faster, easier and more practical to edit and arrange cards than it was to use the primitive line-oriented editors, which were available at the time. Cyber-74 had a very useful program called Update, which could delete and add lines of code in a file, thus, only incremental changes needed to be accumulated. When we acquired the first page-oriented editors from Norsk Data, conditions changed, and Hollerith cards soon disappeared.

An attempt was made to compile the AD-model on an ND machine using an early version of their Simula compiler. The attempt failed due to the program's size and complexity, but it was eventually possible to compile and run the AD-model on an ND-500 machine by splitting it into modules.

About 1990, the AD-model was transferred to a UNIX environment, initially to an Apollo workstation and later to an HP workstation. The compiler was the Lund Simula compiler, which was of good quality.

## **4.2 Command and Control of the Air Stations' Low-Level Ground-Based Air Defense**

This project gave rise to the Air Defense Model (AD-model) [11]. At the time, Norway was planning to buy the Roland II. An important issue was how to coordinate several fire units with high firepower avoiding other fire units engaging the same target at the same time. The AD-model was used to simulate the effectiveness of various ground-based air defense systems in a given scenario, which contributed the principles of deployment. This project established a basic understanding of ground-based air defense and the effects involved [3].

Additionally, coordination between fire units was simulated by passing messages that might be delayed or backlogged. Thus, decisions were made based on a perceived situation rather than on absolutely correct information. This led to the concept of decentralized coordination, which was analyzed with the AD-model. Cost-effectiveness was also calculated, using the simple idea that the maximum value of coordination was confined to the cost of the missiles that could be saved. When this concept was presented to Hughes, they stated that this was the first time they had met a customer who knew what he wanted, why he wanted it, and how much he was willing to pay.

## **4.3 The Hawk Study**

Roland II turned out to be expensive. Some people wanted Hawk instead. No analysis had compared the two. In late 1982, FFI was asked to provide a study in about three months [11], and something unique and drastic happened. Other tasks were put aside and most of the Systems Analysis Group worked on the study. Developing two new systems in the AD-model was considered to be too risky in such a short time. In addition, there was the terrain issue, since our in-house digitized terrain did not cover all the air stations in question. It was, therefore, initially decided to carry out hand simulations using maps, templates, and calculators. Tactical situations involving six airbases were to be analyzed. A program being developed to aid the construction of flight tracks got to a flying start, and was put to use before it was fully developed.

However, three of us were allowed to make an attempt to modify the AD-model, by introducing new weapons and changing the digital terrain interface to a grid-based one. The programming was completed two weeks ahead of schedule. In the meantime, the hand simulations had taken more time than anticipated and had only covered

about one third of the planned work. In just one week, the AD-model completed the rest.

Based on FFI's analysis and the priorities made by the Chief of Defense, a modified Hawk system was chosen. Named NOAH, Norwegian Adapted Hawk, the system had new surveillance radar and, most significantly, a new fire control system from Kongsberg. This fire control system has subsequently been updated and adapted, becoming the backbone of Norway's ground-based air defense.

During the Hawk study, we obtained a small plotter that used colored pens on paper and plastic foils, for which a creative soul found a very good use. He visualized simulation outcomes such as bomb patterns, which aided our own understanding and analysis. The plots were also a useful means of communication. By the end of the study, the plotter had more or less been written-off, but remained usable, nevertheless, for several more years. It was a very useful investment.

#### **4.4 Triangulation of Jamming Strobes**

In the Hawk study, it was noted that jamming could degrade the ground-based air defense radars. The jamming signal (noise) from a jamming pod would show up as a strobe on the radarscope, denying range information, but still providing some directional information. By combining information from two or more radars, it seemed possible to locate the jammers by triangulation [11], which a new project examined. The main problem was the large number of ghosts, which occur when strobes cross in positions with no jammers. A few simple tests could eliminate some of the ghosts, but most of the eliminations came by tracking the PJs (Probable Jammer Location), using advanced statistical methods. By observing each PJ over a period, some PJs would increase their likelihood of being a target, while others would decrease and be eliminated. A dedicated FORTRAN model was used for the triangulation study. The developed triangulation method was implemented in NOAH's fire control system. Norway was actually first to automatically triangulate jamming strobes.

#### **4.5 NASAMS**

Leasing the Hawk missiles was expensive and the Nike system needed replacement. An idea surfaced at the Air Materiel Command in the mid-1980s: Could they use an air-to-air missile like AMRAAM placed in a simple container on a truck? After FFI analyzed some critical issues, employing a specially modified AD-model [11], NASAMS (Norwegian Advanced Surface to Air Missile System) was born. NOAH had been transformed into a very different air defense system while retaining Kongsberg's fire control system as its backbone.

A few years later, there was a question about a more thorough evaluation of NASAMS' effectiveness. Thus, in 1993 and 1994, the AD-model was modified and put to use again.

## **5 Vulnerability of Air Stations**

After the 1967 war between Israel, Egypt, Jordan, and Syria, the low-level threat to air stations remained in focus for twenty years. Ground-based air defense was only one element; the runway was vulnerable and hard to protect. This project focused on runway bombing and repair in an operational context [11], using a couple of dedicated simulation models, of which one was RunBom (for Runway Bombing), a Pascal model. Its main purpose was to find the optimal MOS (Minimum Operating Strip – a sufficiently large stretch of runway on which a combat aircraft could operate), by calculating the least possible number of craters that need to be repaired after a bomb attack on a runway. This problem had been addressed in a model (BABO) from STC (SHAPE Technical Centre), but the model was quite slow. RunBom utilized the fact that fighter-bombers could cut across the runway with series of bombs instead of the heavy bomber method of carpet bombing. It was thus sufficient to place a MOS through one of the cuts, a calculation that RunBom could do quite fast. Therefore, it was expanded to include runway repair as well, simulating repeated attacks in order to study the operational effect of runway bombing and repair over time, deriving requirements for runway repair capacity.

## **6 A Computer with Personality**

Some computers behave in such ways that make one feel they have personalities of their own. A ND500 machine from Norsk Data was our strangest computer. We named it Sisyfos (Norwegian spelling), for the Greek King Sisyphus who was condemned for eternity to roll a rock up a mountain, lose his grip almost at the top, watch the rock roll back down the mountain, then have to start all over again. This computer, Sisyfos, would sometimes unexpectedly stop executing a program at some random point. When we tried the program again, it would often run flawlessly, but we never knew for sure whether it would execute completely. Later I learnt that the ND 500 was supposed to have hardware faults that could, on occasion, be encountered during heavy computing, and thus could cause non-repeatable execution errors.

## **7 Final Discussion**

Simula pioneered powerful qualities and has had a major impact on all other object-oriented languages that followed. However, Simula lacked graphics, and it had no modern debugging tools. Simula's initial high price was undoubtedly a major obstacle for the distribution of the language at the beginning. The subsequent lack of development and support resulted in an unavoidable slow death.

Simulation combines a number of effects in such a way as to reveal complex interactions and relationships that would not be readily derived with less powerful tools. When starting an analysis, one often has some expectations about what the results will be. A model helps to quantify effects and adds them in a way that is hard



for a human to keep track of simultaneously and completely. A model often demonstrates its power by showing some results that initially seem strange and unexpected, but are logical on closer examination. At the end of an analysis, the results often seem obvious. The model may no longer be needed, but without its quality of being a catalyst that stimulates thinking, those obvious results would not have become apparent. A good model may thus render itself superfluous.

A key success factor for operations analysis is to maintain a community of skilled experts where knowledge is maintained and further developed. An organization's collective memory is very important and people are often more important than tools. Regarding both simulation and cost estimation, the tools and individuals involved may have changed, but the community lives on. Another key factor is to analyze the right problems; analyzing the wrong ones leads to irrelevance, no matter how good the tools are.

The use of hardware and associated software tools at FFI has been both strategic and pragmatic. Strategic choices, such as a shared mainframe and FFI's own Simula compiler, were particularly important in the early years. In addition, although FFI's own version of digital terrain from the late 1970s was short lived, it provided a needed tool at an initial stage. As more IT products appeared, market forces have played an increasingly important role in product survivability and the kinds of tools that users are steered towards.

Many studies and computing tools have not been mentioned in this paper, which can only provide some examples and a sense of the computing history of operations analysis at FFI.

## References

1. Njølstad, O., Wicken, O.: Kunnskap som våpen, Forsvarets forskningsinstitutt 1946–1975. Tano Aschehoug, Oslo (1997)
2. FFI Årsrapport 2009, Forsvarets forskningsinstitutt, Kjeller (2010)
3. Solstrand, R.H.: Langsiktig planlegging i Forsvaret – vitenskap i skjæringspunktet mellom politikk, byråkrati og kommandostyring, FFI-Rapport 2010/01924, Forsvarets forskningsinstitutt, Kjeller (2010)
4. Sandnes, H. O.: The 1970–1974 Combat Aircraft Analysis: Priority to Defensive Counter Air and Anti-Shipping Operations: How Optimizing Defense Resources Altered the Use of RNoAF Fighters. Tapir Academic Press, Trondheim (2010)
5. Vebjør, K.T.: Basemodellen – en modell av sortieproduksjon fra et basesystem under en begrenset konflikt. FFI Notat 74/TN-S-362, Forsvarets forskningsinstitutt, Kjeller (1974)
6. Nielsen, S.A., Magnus, E.: Kortfattet beskrivelse av budsjettmodellen "BUDSJ", FFI Notat 77/TN-S-477, Forsvarets forskningsinstitutt, Kjeller (1977)
7. Gullichsen, S.: Long term cost analysis and KOSTMOD 4.0, FFI Notat 2007/01169, Forsvarets forskningsinstitutt, Kjeller (2007)
8. Nygaard, K., Dahl, O.J.: The development of the Simula languages. In: Wexelblat, R.L. (ed.) History of programming languages, pp. 439–493. Academic Press, New York (1981)
9. IT-historien @ UiO.. <http://www.usit.uio.no/om/it-historien/forskning/rbk.html>
10. Egil Heistad, telephone interview by Norodd Hagenson, 1 June 2010

11. Hagenon, N.: Luftvern og sårbarhet av egne flystasjoner. In: Skogen, E. (ed.) Fra Forsvarets forskningsinstituttts historie, vol. 8, pp. 4–12. Forsvarets forskningsinstitutt, Kjeller (2004)