SCIP: A SCANDINAVIAN PROJECT ON SYSTEMS ANALYSIS AND DESIGN

A personal account from a perspective of some 30 years later

Paul Lindgreen [1]

Prof. emeritus, Systems Analysis, Copenhagen Business School; P_lindgreen@hotmail.com

Abstract: Three distinct historic development phases are identified. These phases are based on a review of two years participation together with other Scandinavian colleagues in a very inspiring research group on the development of computer supported analysis and design of computer application systems. These include: 1) Making the computer itself a reliable device, 2) Creating the necessary software and recognizing sound programming principles, and 3) Trying to utilize the provided tools in organizations and business. The paper ends with some reflections on why the two first phases were successfully while the present third phase still is characterized by so many disasters.

Key words: Effective computer usage, co-research, early success versus present failures

1. BACKGROUND

In June 1969 during the NORDDATA conference that took place in Koncerthuset in Stockholm I presented a paper that partly argued for the need of a formal language enabling to specify the functions of computerized data processing systems, and which also discussed the possibility of constructing a compiler transforming the specifications to programs that could be executed on a computer. The most prominent argument for this intended tool, however, was not the program generation itself, but rather that the well-definedness of the possible requirement statements (supported by an underlying formalism) would force the "systemeers" – a term coined in

[1] See author's note at the end of this article.

Scandinavia at that time – really to consider what the data-processing needs were and to be explicit in the specifications thereof.

After the presentation, Professor Börje Langefors from Royal Institute of Technology, Stockholm, contacted me. He commented upon the intentions and ideas I had presented and asked if I would have an interest in joining a group – named SCIP – of researchers from a number of universities and business companies in Scandinavia. He explained that the letters in the word 'SCIP' meant to associate to 'SCandinavian', 'Information', 'Processing', and 'Project' as keywords for some of the main interests and aims behind the work of the group. They had recently and formally established SCIP with economic support from the national research boards in the Scandinavian countries and from an organization named NORDFORSK (Scandinavian Council for Applied Research). Thereby, funds were available during a couple of years for the participation in SCIP by representatives from interested research groups in Scandinavian universities and business companies who were working with theories, principles and tools for proper and effective development of computer application systems – in particular with the design of tools where computers were intended to support the development.

At that time, I was in the employ of the Danish company Regnecentralen, and as a young engineer with no academic background myself other than a master's degree in electronics from the Technical University of Copenhagen, I was, of course, very honored by Langefors' invitation. Knowing that it would be fully in the spirit of Regnecentralen [1] for an employee to take part in such a Scandinavian co-research project, I immediately accepted to join the SCIP group. At that time, I knew Langefors only by name and from reading a few selected parts of his comprehensive textbook: "Theoretical Analysis of Information Systems" from 1966 [2]. He had mentioned to me the names of some of the other SCIP-participants, but I must admit that at that time none of them were known to me, nor were their research projects. However, as I joined the SCIP group I experienced all its members to be inspiring and intellectually open-minded colleagues, and mutual friendships with some of them have lasted since then.

In the preamble of the proceedings [3] from a SCIP-organized conference in Århus, Denmark, that in practice seems to be the last officially recorded act of the SCIP group, Langefors writes about the background for establishing SCIP:

"The present SCIP work may perhaps in retrospect be regarded as having started in 1968 in Trondheim, Norway (SINTEF), when a group headed by Arne Sølvberg started to implement a computerized system for information system analysis and design. About a year later (1969), project CADIS was started at the Department for Information Processing at Stockholm University and the Royal Institute of Technology; and shortly after, at a

meeting in Stockholm with people from Denmark, Finland, Norway, and Sweden, it was decided that a Nordic cooperative research (and application) effort was desirable. Thus SCIP was formed with support from ..."

2. SCIP PROJECTS AND PARTICIPANTS

The only (at least publicly available) documentation about the Scandinavian research projects involved under the SCIP-umbrella appears in the above-mentioned proceedings from the Århus conference [3]. These proceedings are rather unusual as regard the *editorial form*, because each paper includes unusually and extraordinary detailed records of the associated discussions during the conference.

Langefors himself had an introductory paper in the proceedings, but this paper does not describe a proper SCIP project. Rather, it summarizes some aspects more or less common for all the projects. Additionally, it contains some personal comments about issues and concepts that he – at least at that time – regarded as important in system analysis and design. In addition, the discussions on these issues are recorded.

In the following, the "hereby official" SCIP projects appear listed in the order in which they appear in the proceedings with specification of the name of the project and/or a short summary of the aim(s) of the project, with names and affiliation of those of the involved or associated persons who at least once attended a SCIP arrangement:

1. SYSKON – a project concerning System Development
- Christian Andersen, Institute of Management, University of Århus
- Fritz Krogh – Jespersen, A/S Regnecentralen, Århus
- Anders Petersen, Århus Business School

2. Project NO (Apparently an acronym based on the first name of each of the two project members)

The purpose of project NO is to develop methods for the production of a decision basis for political decisions, the documentation of the decision basis and of the political decisions, and ensuring that the system owner makes political decisions.
- Ole Øhlenschlæger Madsen
- Niels Jørgen Relsted

both from Institute of Management, University of Århus

3. DATAMATICS and INFORMATICS

The task is to develop the abstract foundation of datamatics – named *systemology* and *systematics* – and its application to the development of information systems – named *informatics* (or *pragmatic cybernetics*).

– Poul Sveistrup, University of Copenhagen

4. *Exact Description of DAtamatic Problems (EDDAP)*

The purpose of EDDAP is to:

- Analyze and describe the conceptual basis for a formal language for specifying the systemic function of datamatic systems, i.e. the function as experienced by the users of the system
- Construct a declarative language based on the realized concept structure
- Design and program a compiler that from an EDDAP description can generate the program that determines the function of the described datamatic system.

(The contribution to the conference was in fact a working paper under the first of these three sub-projects.)

– Paul Lindgreen

– Flemming Sylvest Pedersen

both from A/S Regnecentralen Copenhagen

5. *DIFO – Design of Information systems, especially File Organization*

To develop methods for the design of a file organization that minimizes the sum of the following cost factors: design of files, programming of file manipulation programs, generation of files, maintenance of files, data processing, and required or reserved space.

– Pertti Järvinen, Departement of Computer Science, University of Tampere.

6. *CASCADE – The development of tools and methods for the analysis and design of information processing systems*

– Per Aanstad

– Geir Skylstad

– Arne Sølvberg

all from Computing Center, University of Trondheim

7. *CADIS – Computer-Aided Design of Information Systems*

The CADIS project proceeds along two strongly interrelated lines. One is to expand and develop the methodology of information systems analysis and design. The other is to develop computer-based tools (man-machine interactive programs) which may aid analysts and designers in their work.

– Stig Berild

– Janis Bubenko, Jr.

– Olle Källhammer
– Eva Lindencrona
all from Dep. of Information Processing, Royal Institute of Technology, Stockholm.

8. The specification of formal information systems for administrative control
– E. Torsten Lundquist, Departement of Development, KemaData AB, Stockholm.

9. The development of a computerized real-time documentation system applying computer graphics as a means for interaction.
– Hans E. Andersin
– Kristel Siro
– Markku Syrjänen
all from Institutionen för Informationsbehandlingslära, Tekniska Högskolan, Helsinki.

In addition to this list of projects and participants it should be mentioned that in a great part of SCIP's lifetime the extremely friendly and helpful employee from NORDFORSK, fil.cand. Ann-Kristin Wentzel, served as a perfect host for the SCIP-sessions on the various sites in Scandinavia where we joined.

3. SCIP IN PERSPECTIVE

In the relatively short time around 1970 when SCIP was active and the involved persons could meet two to three times a year, they would discuss and exchange knowledge and experience. In fact, they had a "co-research life" in that period, I believe, and we all sensed that we were at a turning point at the start of a new phase in *datamatics* – a term I coined in the sixties for the discipline of applying electronic computers in society. I do not recall how this shift happened explicitly – neither in SCIP, nor in broader context. But in retrospect the following three *phases of problem-focus* in the area of the electronic computer usage can be clearly distinguished:

1. In the 1960s, research concentrated on *developing reliable hardware* for computers and for data-storage devices in a scale that was enormous compared to the pioneering time in the forties. In Scandinavia, this period ended with the manifestation of computers such as BESK, DASK, and SARA and organizations that could provide access to use of them.

2. In the 1960s, we also saw a decade where the concept "software engineering" emerged. The major research effort was to construct and specify *adequate programming languages* such as FORTRAN, ALGOL, and COBOL to invent *useful compilation principles and techniques*, and actually to develop *compilers for practical use.* In this period, the basic principles for constructing *reliable and maintainable algorithms* became important with principles that later on became known under the term 'structured programming'. With the contemporary construction in Norway of the SIMULA-language [4] also the conceptual basic principles were laid for what later on became known as "the object-oriented approach" – the dogma that from the nineties and on has become an extremely successful paradigm in software engineering and, unfortunately, elsewhere in analysis and design remains a sad conceptual misunderstanding (see the concluding reflections).

3. It was at the end of the 1960s that SCIP was born and from that on had its relatively short life. At that time where the technological foundation for a widespread practical use of computers in society was established, *"the era of datamatics"* was born which we up to now in still increasing scale experience in all parts of society. However, SCIP clearly showed an indication of a change of focus from computers themselves and how to program them to *how to utilize* the available potential of data processing and data storage technology in organizations. That is, *how to do it in a sensible, rational, and resource-effective way.*

But, alas! Despite all the inventions and data-handling and transmission possibilities of datamatics that have appeared nearly continuously since then, and despite that some of them in practice have revolutionized the way we *potentially* can act and cooperate in society – we still are not able to organize projects aiming at utilizing the technical potential in a sensible way. Many development projects have become organizational and/or functional disasters. New projects still do so and most of them far exceed their budgets both as regard costs and as development time. Furthermore, nearly always when a developed system finally is released for use, a long period follows with costly activities to modify the system such that it finally will support effective business procedures instead of prohibiting them.

It is interesting that in strong contrast to the previous two phases this third one aiming at *utilizing* the datamatic potential in a sensible way – *this phase has lasted for more that 30 years.* As I see it, we are still far from reaching a practice where project teams and their employers *honestly* can say: "Our development was rational and resource-effective, because our approach was in accordance with well-recognized principles and based on *a consistent structure of relevant concepts* that support our understanding instead of distorting it".

However, we must admit that to reach such a state was never the official goal of SCIP. The task was not to search for a proper theoretical basis for "systemeering". Since the official intentions of NORDFORSK were to support *applied* research, the primary objective of SCIP was to provide a forum for exchanging research results and provide them for the public to help improving the practical development of datamatic systems. In accordance with this, most of the projects under the SCIP-umbrella were concerned with the construction and implementation of some kind of working approach and/or practical tools to support the work – most often computer-based ones. However, as I remember the extremely vivid, interpreting, and often very exited discussions during the SCIP meetings, they nearly *always* were theoretic and primarily concerned with basic concepts and the conceptual structures that were assumed to be behind the various system design issues in practice. This deviation from the official SCIP intentions can actually be seen in the above mentioned introductory paper of the proceedings from the SCIP conference in Århus,. There Langefors directly regrets the strong focus on (assumed) relevant concepts, which constantly popped up during the discussions of the conference papers.

On the other hand, this deviation from the official NORDFORSK goal was exactly what created the "academic seed" that would grow into a new very active international research forum. Arne Sølvberg and Börje Langefors took the initiative. They (and possibly supported by others as well) carried out a great deal of organizational work. This resulted in the establishment of a new technical committee under IFIP named TC8 (Information Systems) and in the forming of its first working group 8.1 (Design and Evaluation of Information Systems), a group I also later on joined and where I again met many of the former SCIP colleagues. Over the time, TC8 became one of the most active organs of IFIP. Nevertheless, it is interesting that whenever I listened to or took part in the always-stimulating discussions during the various WG 8.1 sessions, I usually experienced a SCIP déjà-vue.

4. SOME REFLECTIONS ON THE THREE HISTORIC PHASES OF PROBLEM FOCUS

When I presented the draft version of this paper in Trondheim, I stated:

> "Although I am not a historian, it is my opinion that any historic record is of little interest, unless we can learn something from it – something that in some respect may help us to behave better in the future".

Up to this point, the paper to my best knowledge is historically correct, although – as mentioned in the preamble – it was not possible to provide the full story of SCIP. The temporal position of SCIP within the three

phases of problem focus is also correct. Somebody may object that the description of the last (current) phase might be too negative, taking into account "all the great achievements" we have seen in our professional domain.

Well, future historians must decide how fair the judgment of this phase really is, because my historic point of realization is a quite different one – the obvious question every active computer professional in my opinion should ask and try to answer:

> Why is it that the epoch covering the first two phases was so relatively short, and still – despite the pioneering conditions – extremely successful compared to the current epoch of the third phase: the one characterized by an endless flow of implementation-"catastrophes" and a galore of short-lived buzzwords for fancy new "concepts" and "revolutionary" approaches?

It is far beyond the scope of the present HINC conference to aim at a full answer to that question. However, on the professional and historic background of having been active through all three phases, I have over the years tried to find some *key issues* behind the apparently lasting problems with utilizing the datamatic potential in organizations. On that background I allow myself to conclude with an interpretation of *some important differences* between the two epochs, and also – in the spirit I experienced both at Regnecentralen and in the SCIP project – to give a few hints of what in my opinion basically must be changed in "the datamatic society", if the current epoch shall end with success. (Several much more detailed comments on some of the issues can be found in [5] and [6]).

So what are the most crucial differences between the two epochs? In both phases of *the first epoch*, the focus was on making the computer *a reliable and effective tool.* The epoch was very successful in this respect, and the main reason was that the objects of the problems were *deterministic.* As the outcome of the epoch, the people behind the development projects provided society with:

- *A fast data-processing device* that after the invention of the transistor (and other kinds of electronic components) became *reliable in* practice
- *Useful programming principles and languages*
- *Reliable compilers and operating systems* that permitted a relatively effective and fail-safe utilization of the datamatic tools as such.

Most characteristic for the first epoch compared to the present one is that only *relatively few people* worldwide – often in university(-like) environments – were involved in the development of the necessary hardware and software. Of course, already at that time, thousands of electronic engineers and software pioneers were active in the development, but probably less than fifty *key-persons* worldwide really made the difference by fostering the right ideas and creating the insight that was the

basis for the success. Important is also that the development in practice mainly took place under conditions where the involved persons freely could:

- Exchange design and research results
- Discuss with each other across national and organizational domains and thereby filter out common *useful* concepts that supported the understanding of the datalogical problems and principles
- Provide techniques for each other that enabled the construction of effective and reliable computers and software for controlling them.

The present epoch, in contrast, is quite different at least in two respects. *First,* the number of people active in datamatic development has increased dramatically, and the whole area is now commercialized. The most prominent result is that the intense and insight-giving communication characteristic for the first epoch has given way to a narrow-sighted clinging to ideologies and dogmas.

Secondly, the problems have moved from the deterministic hardware/software domain into the basically *un-deterministic* realm of *organizations*. Here the datamatic systems were to interact with *human actors*. Well-known expectations are that computers should be able to serve the organization – typically by mediating the communication between people, by acting as an external memory for them, and even to take over some of their former business activities – for example to make useful estimates or to take proper decisions. Moreover, because computers compared to humans are deterministic and extremely faster, they are often supposed to be able perform their tasks in a much better and cheaper way than people can do.

In principle this may be correct, but often it is realized too late in the development process that it is not necessarily so in practice, and then severe problems occur. Ironically, it is the success with computers in the first epoch that is the reason: It has caused a severe blindness as regard two underlying, but generally unrecognized or ignored problems:

1. Exactly because computers are deterministic they lack the important human ability *to improvise* – to deviate from strict rules and to react in a sensible manner also in situations that were not foreseen by the programmers.
2. Even worse – the success in developing an effective data-processing tool caused the widespread misunderstanding that the hardware/software *concepts* useful for solving the problems of that epoch also were those relevant for understanding and solving the present problems with computers as actors in organizations.

From many years of trying "to understand the world", it is my experience that for a notion of something to justify as a *concept* it must be in accordance with the following definition:

A *concept* is a conception of a thing characterized in such a way that exactly thereby it becomes *useful* for understanding other things – in particular for understanding other concepts. However, a set of sensible concepts is not enough: One of the important things I learned from participating in SCIP was that even the most well renowned and generally recognized concepts could be disastrous to base your insight on, *if they do not properly reflect* the phenomena that are relevant in your domain of analysis. That is exactly the sad situation we now have experienced twice in the present epoch:

For many years, the dogma behind the development and implementation of datamatic systems as co-actors in organizations was to use the popular, guru-advocated, but utterly naive Structured-Analysis and Structured-Design approaches. When at the end of the eighties it finally became obvious – even for the most religious adherents – that the approaches were quite unsatisfactory, an alternative ideology had to be found. And what luck; it was there just to pick. In the software-engineering domain, a truly successful development codex had emerged.

Exactly because the so-called *object-oriented approach* has been – and still is – *extremely successful in software design*, we wrongly assume that then it must also be the proper paradigm in an organizational context. That is not so. The reason is that the OOA-paradigm is far from reflecting the concepts that are relevant for understanding and describing organizations. One can model organizations in many ways, but behind the understanding of any organization is a set of *fundamental un-avoidable concepts* of which the most important are:

> Actors, actions, and co-actions, interaction with the environment, operands, goals for the activities, communication between actors, information, and data representing the information.

However, brainwashed by the OOA-paradigm, the OOA-systemeers view an organization just as a complex structure of interacting "objects" of various types.

5. CONCLUSION

Therefore, what can we learn from the "story" of SCIP viewed in this subjective historic perspective? Well, Piet Hein once expressed it generally as "TTT"; that is, "Things Take Time". I could add, "Getting insight takes much more time".

REFERENCES

[1] Per V. Klüver, Technology Transfer, Modernisation, and Welfare State – Regnecentralen and the Making of Governmental Policy on Computer Technology in Denmark in the 1960s. Procedings of HiNC1.

[2] Börje Langefors: Theoretical Analysis of Information Systems. Studentlitteratur 1966.

[3] Janis Bubenko jr., Börje Langefors, Arne Sølvberg (Eds.): Computer Aided Information Systems Analysis and Design. Studentlitteratur 1971.

[4] Stein Krogdahl: The birth of Simula. Procedings of HiNC1.

[5] Paul Lindgreen: Systemanalyse: Analyse af organisatoriske systemer med henblik på fornuftig edb-anvendelse. Jurist- og Økonomforbundets Forlag, 1990.

[6] Paul Lindgreen: Towards a useful Information concept. In: Brinkkemper, Lindencrona, Sølvberg (Eds.): Information Systems Engineering. Springer 2000.

AUTHOR'S NOTE

I regard the period of participating in the SCIP activities as an extremely vivid and inspiring part of my research career, and as one during which I learned much from Scandinavian colleagues. It was a time when I became conscious of many of the fundamental principles and concepts underlying all kinds of practical use of computers. Despite this, I have been able to dig out only a minor part of the SCIP-history. Many facts about SCIP were apparently never recorded or safely stored, and much knowledge about participants, projects, and about the many discussions of concepts, approaches, issues, positions , etc. that may be relevant from a historic point of view, is now forgotten or is lost in the minds of those SCIP participants who have passed away.