Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte
Band: 40 (1982)

Artikel: NEMESIS: a joint effort of computer users in the Netherlands
Autor: Kruisman, Gerard
DOI: https://doi.org/10.5169/seals-30897

NEMESIS: a Joint Effort of Computer Users in the Netherlands

NEMESIS: Résultats d'un effort concerté des ingénieurs aux Pays-Bas

NEMESIS: eine gemeinsame Leistung der niederländischen Computeranwender

Gerard KRUISMAN
Consult. Eng.
R + K Consulting Engineers
Rijswijk, the Netherlands

Gerard Kruisman, born 1936, obtained his civil engineering degree at the University of technology, Delft, Netherlands. He became a consulting engineer, specializing in new fields in engineering. He is an advocate of multi-disciplinary cooperation within the profession, and has helped to create many organisational structures, especially in the application of CAD techniques.

SUMMARY
As a result of close cooperation between engineering users of the GENESYS system new ideas on the use of engineering systems have been developed into a concept for a new Netherlands Modular Engineering System (NEMESIS). The paper deals with the basic experiences gained with the use of the Genesys system and the policy criteria for developing a new system based on the advanced technology available today. An inventory of user requirements is provided as well as the results of a survey on CAD/CAM interests in the Dutch industry. Organizational and technological aspects of the NEMESIS development project are presented.

RESUME
A partir des expériences réalisées par de nombreux utilisateurs du système GENESYS, les bases d'un nouveau système informatique modulaire, destiné aux applications techniques, ont été élaborées en Hollande (Netherlands Modular Engineering System). L'article présente les principaux critères qui doivent guider les développements s'appuyant sur une technologie moderne. Un inventaire des besoins des utilisateurs est établi ainsi que l'intérêt manifesté par l'industrie lors d'une enquête sur la CAO. Des suggestions sur l'organisation et les aspects techniques du développement du projet NEMESIS sont formulées en guise de conclusion.

ZUSAMMENFASSUNG
Dank enger Zusammenarbeit zwischen den Anwendern des GENESYS-Systems wurden neue Gedanken in Bezug auf die Anwendung technischer Systeme ausgearbeitet, die zum Entwurf eines neuen niederländischen technischen Modulsystems (Netherlands Modular Engineering System: NEMESIS) geführt haben. Der Text behandelt wichtige Erfahrungen mit dem Gebrauch des GENESYS-Systems und Kriterien für die Entwicklung eines neuen Systems auf der Grundlage der heute verfügbaren modernen Technologie. Eine Liste der Anforderungen der Anwender wird gegeben, sowie eine Übersicht über die Resultate einer Umfrage über die CAD/CAM-Interessen bei der niederländischen Industrie. Organisatorische und technologische Aspekte des NEMESIS-Entwicklungsprojekts werden erwähnt.

CONTENTS

INTRODUCTION

During the IABSE Colloquium on 'Interface between computing and design in structural engineering', Bergamo, Italy, 1978, the author presented a paper entitled 'The unknown triangle', in which paper relationships between a human being and his surrounding environment are described as subjective experiences. Through education, training, cooperation or other ways of exchange of experience between human beings, subjective experiences may become intersubjective experiences. Natural languages, sciences, political trends, religions etc. have at least some parts in them that bind groups of people together in such a way that the relationships between members of such group may be considered to frame an intersubjective reality.
The same applies to engineers. They have their technical background - civil engineering, mechanical engineering or otherwise. Within each discipline there are specialists like structural engineers who have built up their intersubjective reality focused on design and realization of the structures that provide strenth and stability to our built-up environment.
During recent years computer based tools have become available that enable engineers to store and process data they use for the design and construction of these structures. Data can relate to the design process, such as geometrical descriptions of the structure, structural materials, deformations and strength behaviour etc., to the production preparation, such as technical specifications, component lists, planning and routing schedules etc., and to the production itself. In the latter case one can think of NC machines, optimized material usage, material and product storage management etc.
Computer based tools have in some case become rather specialized. Examples are automated draughting systems and programmable handlers like industrial robots. Although still primitive in comparison to human abilities, fascinating examples are available of robots, equipped with vision and tactile sensors, which have a 'learning' capacity.
Designers of computer based tools for production engineering look at present production procedures, analyse and streamline working methods and information handling process, trying to create tools that fit into and integrate the production process. In this paper the author will confine himself to the design process and report on attempts that are being made in the Netherlands to develop a 'generally applicable engineering system'. The closing remark[1] provided at the end of the above mentioned paper contains a stimulus for the development of such a system but it also encompasses discouragement, disappointment and ultimate failure if the subjectivity of the experience of human individuals is not taken into account as a major fact of life.

[1] From the foregoing (the paper) it may be clear that to attempt to define a general interface between design and computing is fruitless. Each designer and program developer will define his own interface. Through continuous discussions and cooperation these individual interfaces will be unified and at least match, at least a little.

It is therefore not without reason that the new engineering system has been named NEMESIS (Netherlands Modular Engineering System) after the Greek goddess of retribution for human recklessness.

AN ENGINEERING SYSTEM

Take a micro CPU chip in one hand and a detailed drawing of some structural component in the other.
Although the micro chip in the hands of a designer can produce the drawing, there is quite a job to perform and specialized experiences to provide in order to be able to elaborate the chip into an engineering tool that enables the designer to carry out the necessary design calculations and to produce the drawing.

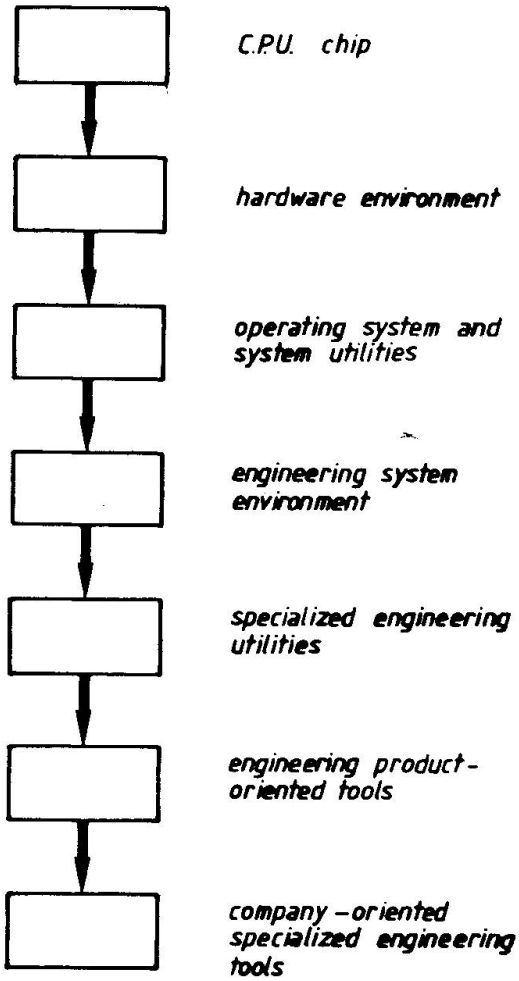

Fig.1 Chain of specialized knowledge required to make a micro chip useful to an engineer in his company-environment

The idea of bringing together a group of computer programs and extracting from them common functions like data structuring and handling functions, mathematical functions, drawing or plotting functions, I/O functions based on a so-called problem-oriented language (POL), etc. is not new.
The integrated Civil Engineering System (ICES) and the General Engineering System (GENESYS) are - amongst others - developments of the sixties and they are still in use after almost twenty years.
Once a user has learnt the POL methodology, he will be familiar with all the engineering application programs contained within the system.
Programmers who have learnt to use the engineering system functions (in the case of ICES contained in the ICETRAN language and in the case of GENESYS in the GENTRAN language) will be able to effectively produce userfriendly programs for the engineering end user.
Although these engineering systems are still in use and have enthusiastic supporters, be it within the worldwide ICES User Groups (IUG) or on a smaller scale the Association GENESYS Netherlands (VGN), the existing engineering systems cannot easily be adjusted to the new developments in hard- and system software.
The systems are based on hardware concepts from the sixties. System components cannot easily be replaced and the system architecture is of a 'closed' type.
To connect such a system with an external utility such as a DBMS (Data Base Management System) or with an automated draughting system is quite a costly job.
This is not a characteristic of only these engineering systems. There have become available very powerful (and costly) draughting system which fail to have possibilities to connect external systems[2] . Other disadvantages of existing engineering systems are the lack of effective programming aids. Although the dedicated programming languages like Icetran and Gentran enable the programmer to write much more user-friendly and effective programs with the aid of built in dedicated I/O functions (data-entry and report functions, command interpreters and definition functions), the writing of these programs requires more effect than direct programming in e.g. Fortran. Managers of engineering user environments cannot easily be persuaded to adopt an engineering system that costs a yearly licence fee, imposes a computer overhead, requires extra education of end users and programmers, leads to more expensive computer programs which cannot use specific features of the in-house computer systems and where the cost-effectiveness of the problem solving process cannot be proven by the system.
Notwithstanding these disadvantages it is a fact that user environments can become enthusiastic about the possibilities such systems offer to improve the quality of the problem solving process.
Quality improvement results from the high level user interface which the engineering problem solving programs (application programs) within an engineering system offer to the designer. Through this improved communication technique the designer can 'play around' with his problem. He can easily vary structural parameters and investigate in this way the sensitivity of his structure to deviations of actual values from assumed values during design, or vary the geometry of the design and prove that the geometry he supposes to be an optimal solution, really meets the requirements.
Experience results from 'trial and error', trespassing borders marking commonly accepted impossibilities.
Just watch a playing child trying to impose its imagination on the environment, leading to confrontation and thus to experience. Artists perform experiments, using all kinds of new techniques in order to find new ways to express themselves.
Why not provide such facilities to the designer, so he can play around with the procreations of his mind? Isn't that the intrinsic value of Computer Aided Design?
Engineering systems shall provide an effective CAD toolkit to the designer to enable him to gain experience on the products of his mind. He cannot afford a structure to collapse in the real world, but in his subjective world models should have an ability to show consequences of ultimate design.

[2] see also FACE-report 2 'The automation of draughting work' [17]

EXPERIENCES WITH THE GENESYS SYSTEM

In 1972 the GENESYS system was presented officially to the engineering society at Loughborough, UK, and in 1972 the Department of Waterworks in the Netherlands (Rijkswaterstaat) signed a licence-contract with GENESYS Ltd. Within the Association of Computer Users in Engineering (CIAD) a project group 'Integrated systems' showed much interest in the GENESYS system and in the developments of engineering application subsystems that were performed by the Rijkswaterstaat. The system became used by consulting engineering firms and others, but the expected widespread use by engineering environments did not show up. Main bottlenecks were the relatively high licence fee for the main system, lack of specialist subsystems based on Dutch codes of practice, thresholds to learn and use new working methods and the fact that a governmental body is not equipped to promote externally the use of such a system. The GENESYS system design is such that about 95% of the Fortran source code is computer-brand independent and the remaining 5% of the code contains the interface to a particular computer and its operating system. The support of many different implementations appeared to be a limiting factor.
In 1975 discussions started between Rijkswaterstaat and the project group on the establishment of an independent GENESYS Centre in the Netherlands, resulting in the formal charter of the Association GENESYS Netherlands in the beginning of 1977. The Association groups together users and potential users of the GENESYS system. Shortly afterwards the Foundation GENESYS Netherlands was created by CIAD, Rijkswaterstaat and the newborn Association. It was the intention (and in the meantime this has taken place) that after some years both CIAD and Rijkswaterstaat would withdraw from the Board of the Foundation and the vacant seats within the Board would be filled by representatives of the Association.
It is the main purpose of the Foundation to exploit the GENESYS system in a professional way, whereas the Association is a forum for the users where they can exchange experiences, work together on developments and express their requirements in a structured way towards the Foundation. After about five years this organisational structure has been settled and it is functioning quite successfully.
The Association applies the same success formula for its functioning as CIAD, which is not surprising because there is a strong connection to CIAD. Most of the members of the Association are members of CIAD too.
During the five years of its existence representatives of the 15 member organisations have worked hard on a number of subjects related to the improvement of the effective use of the GENESYS system. A total number of about 50 reports have been produced.
Subjects of reports are
- organisational procedures and annual reporting (11)
- evaluation of existing (British) subsystems (3)
- development reports on new (Dutch) subsystems (4)
- bottleneck investigations on the use of the GENESYS system (3)
- new GENESYS implementations (1)
- inquiry on user requirements for and development of draughting facilities attached to the GENESYS system (9)
- development of facilities for interfacing existing Fortran programs to the GENESYS system (4)
- development of quality classification standards for subsystems to be made generally available through the organisation (4)
- investigation on improvements of the existing GENESYS system (2)
- investigation into and preparation of the development of a new engineering system NEMESIS (12)

Which means that much effort has been given to the consolidation of the Association in order to obtain a firm basis for extensive exchange of information, quality improvement and the possibility to effectively deal with new arising needs

of the users. Especially the initiative to develop ideas on the use of computers by engineers in near future, taking into account new technologies both in hardware and in software, attracted new members to the Association, who were not GENESYS users themselves. It is felt within the Association that the original objectives of the developers of the GENESYS system still hold.
- uniform main system with general functions suitable for use within engineering application programs
- high degree of portability of main system and subsystems, with the computer-dependent operating system interface concentrated in the main system
- standardized and unambiguous user interface and documentation
- high degree of user friendliness,

In marketing the GENESYS system in the Netherlands the Dutch GENESYS organisation (Association + Foundation) with a number of thresholds which present potential user environments to use the system.

Important thresholds are
- in case of existing computer using engineering environments (non-GENESYS-users)
  . effort to be made by the user environment to become familiar with the specific GENESYS working method.
    This Threshold is in agreement with [15] where the development of a level 4 I/O standard is not recommended as long as the previous levels are not generally accepted. The I/O user interface of the existing GENESYS system meets - at least to a high degree - the requirements for level 4.
  . resistance from existing users who are familiar with existing Fortran programs. As a result of the often cumbersome I/O rules (which differ per program) experienced users become experts. It is their skill to use such programs. User friendly programs make their skill redundant.
    The general question is 'Why should we change over to a new system with all related problems, both financial and organisational, only to get available programs we already use in Fortran'.
    Only where GENESYS subsystems provide problem-solving tools which are required at that time experienced user environments may become interested to look at the subsystems and thus at the system.

- in case of existing GENESYS using engineering environments
  (thresholds to put more effort in extending the use of the system over the organisation, e.g. involving more departments or writing new subsystems)
  . up to recently the GENESYS system was available only on mini and main frame computers. Especially in the latter case the volume of required direct memory can be disadvantageous to multi-user environments. Users sometimes have to wait rather long before the 'large-GENESYS-program' can run on the computer.
  . writing subsystems requires programmers trained in the use of the Gentran programming language. Since there is little experience available, programmers have to become experienced mainly through 'trial and error', which is not the most efficient (although effective) way.
  . although the use of user-oriented command structure almost obliges, but at least seduces, the programmer to design a functional model, the GENESYS system does not provide programming aids to define a structured process or a process-oriented data structure. Some programming aids are available (e.g. the TRACE statement), but testing is still as uneasy as in normal Fortran.
  . as the GENESYS system is almost a 'closed' system, other software systems or application programs can hardly be connected to the GENESYS system. The GENESYS system does not provide facilities for external data transfer or working within a network.
  . as the GENESYS system is developed in the late sixties the system architecture is based on hardware and software concepts known at that time. Modern concepts available cannot be used within the system and its subsystem (e.g.

menu-techniques, digitizers, text processing, draughting techniques).
as the GENESYS system is one monolith structure, it cannot easily be tailored to the needs of individual user environments (e.g. solving a small problem using a small subsystem requires almost the same computer resources as solving a large problem using a large subsystem).

The general feeling within the Association is, however, that the original GENESYS-'idea' was and still is excellent. The resulting GENESYS product - developed about 14 years ago - is still usable for small engineering environments which start using computers. As a micro-version of the GENESYS system is available now, hardware costs will not be a limiting factor for these environments anymore.
As soon as those engineering environments start to make their requirements more advanced and want to integrate modern techniques they become trapped by the system.

An often heard objection from opponents of the GENESYS system is the computer-runtime overhead required by the system.
Advocates of the use of the system reply with the statement that the required man-hour time to solve a problem is much less. However, no data were available.
One of the projects of the Association focused on this problem.
At the University of Technology at Eindhoven in the Netherlands a test was performed using three groups of students; none of them had any knowledge of technical application programs and most of them even did not have any experience in the use of computers.
Each of the students had to compute stresses in a truss structure under several loading conditions.
One group had to solve the problem with help of the GENESYS-Vlasko program, another group with the ICES-Strudl program and the third group with the STAEX program. The latter program being a normal Fortran program with a general cumbersome input structure.
The following figures resulted from this test.
Required man-hours (fortran-program taken 100)

	STAEX	ICES-STRUDL	GEN.-VLASKO
acquaintance with user manual	31	27	10
input preparation	34	49	26
input correction	25	13	22
output interpretation	10	5	7
	100	94	65

From these figures it may be seen what influence an easy input preparation - through easy understanding of the underlying model - has on the overall required man-hours.
A similar conclusion can be drawn from the figures related to the average amount taken 100)

	STAEX	ICES-STRUDL	GEN.-VLASKO
Average of required times of assistance	100	108	77

The figures on required computer time to solve the problem (including error runs) were not very consistent. Leaving out data which could not be explained (the students had to collect the data) the following figures resulted from the test (Fortran program = 100)

	STAEX	ICES-STRUDL	GEN.-VLASKO
Average required computer time/run	100	219	363

Due to the uncertainty of the test data, these figures should not be taken too literally, but a system overhead of about a factor 2 for the GENESYS system has been found in other circumstances too.
If the costs of computing for solving a problem, using normal Fortran programs, are taken at about ¼ of the man-hour costs, which assumption does not seem to be unreasonable at this time, following figures would result from changing over to the use of an equivalent GENESYS subsystem (man-hours costs of Fortran-program = 100)

	Fortran program	GENESYS-subsysteem
man-hour costs	100	67 (factor 2/3)
computer costs	25	73 (factor 3)
Total costs	125	142

With a tendency of increasing man-hour costs and decreasing costs of computing, the cost figure will favour the use of engineering systems in future.
In some cases the costs of personal (micro) computers are accounted for in an overhead percentage on the man-hour costs. In those cases the use of an engineering system (especially an easy to use interface in conjunction with an easy to understand user manual or other means of information transfer) directly pays out. Easy use and understanding means, however, a larger environment where the rules and standards are generated and generally accepted.

TOWARDS A NEW ENGINEERING SYSTEM

At the start of the Association a survey was held among the GENESYS users on existing bottlenecks in the use of the system. Main bottlenecks have been described in the previous chapter. As a result of the survey the organisation GENESYS Netherlands contemplated its policy. Three policy lines were defined to meet the stated user requirements.

policy line 1 : Improve facilities of the GENESYS system through additional developments.
Major developments were
- interface package to Fortran programs ('Fortran-hook')
- draughting package (Calcomp-Gino-F simulation)
- micro GENESYS on 8-bit micro computer under CP/M

policy line 2 : Discuss with British contract partner renewal of the existing GENESYS system
Major developments were
- investigation of future market potentials of the GENESYS system
- investigation of the feasibility of merging the GENESYS system with a more recently written complementary system (BOS-system).

policy line 3 : Definition of a new, engineering system to be developed in the Netherlands
Basic starting points (for a new system which has been given the name NEMESIS are
- continuation of present working methods and attained standardization of computer use
- continuation of investments in existing GENESYS subsystems
- anticipation on the general use of micro computers
- system to be based on an 'open-growth' architecture, which means the system must have the ability to be tailored to the user requirements and existing hardware in a particular user environment together with the ability to grow simultaneously with growing needs of this environment
- knowledge on system software design and development to be increased and intensified in Dutch engineering software industry through system development

- continuous standardization of computer use to be based on continuing inventory of user requirements
- existing international contacts to be maintained and intensified.

The policy statements under this line are more of a long-term character and aim at innovation of Dutch industry from an informatics point of view catching up with developments in the industrialized countries.
A project group was installed with the task to define preliminary specifications for the new system and to investigate the feasibility of development.
The project group produced in three years a total number of 9 technical reports with the following subjects

- collection of building stones for the approach of the task to be performed
- inventory of similar, already existing systems or intended developments as far as known
- inventory of user requirements, based on the building stones collected
- general description of a concept for NEMESIS
- related work documents produced by the project group and other relevant documents
- investigation on the user friendliness of a GENESYS subsystem compared with other similar application programs
- the technology within NEMESIS
- inventory of on-going research projects at universities and research centres in the Netherlands of which results can be useful to the NEMESIS development
- NEMESIS development strategy
- NEMESIS project organisation

As the NEMESIS system is intended to become infrastructural software for CAD/CAM applications a step-wise development is foreseen, each successive step resulting in an improved system relative to the previous step. A start should be made with the development of a relatively simple system, followed by several steps in order to reach a fully grown-up system and going over in an 'innovation permanente', where new technologies from the market are continuously incorporated in the system.
The development of NEMESIS is foreseen as a catch-up manoeuvre of Dutch industry with CAD/CAM developments in industry in other countries.
It is recognized by the project group that the propagation of objectives of system development and spread of knowledge to be within the system and the attainment of enthusiastic involvement of large numbers of potential users is of the same order of importance as the incorporation of new and advanced technology in the system.
Especially the abstract character of the system makes it difficult to persuade potential users of the future applications software to become attached to the system and to understand the necessity of such system development, and they have difficulty in estimating their future interest. Only with help of the existing GENESYS system and subsystems one can show on a piece of (micro) hardware what is meant. There is an enormous gap between the NEMESIS development environment and non-computer oriented managers of engineering design and construction environments. 'Unspoiled' managers believe computers can be used by engineers in the same, simple way as they can drive their cars and providing the tools to make that attitude possible in future is exactly what the developers of NEMESIS are aiming at.
The gap between results from the complexity of human activities and flexibility of human behaviour in comparison with the 'stupidity' or low-level artificial intelligence of today's computers.

At several locations in the Netherlands CAD/CAM developments in industry are on their way. However, only large companies can afford to investigate possibilities of implementati CAD/CAM tools in their production processes.
Small and medium-sized companies do not have resources to follow developments

in the market 'to play' with possibilities in order to find out what is needed and what is useful. They have no time and money to analyse their production processes and to select and implement adequate computer based tools in those processes. In the diagram a qualitative approach of the indicated CAD/CAM problematics is shown.

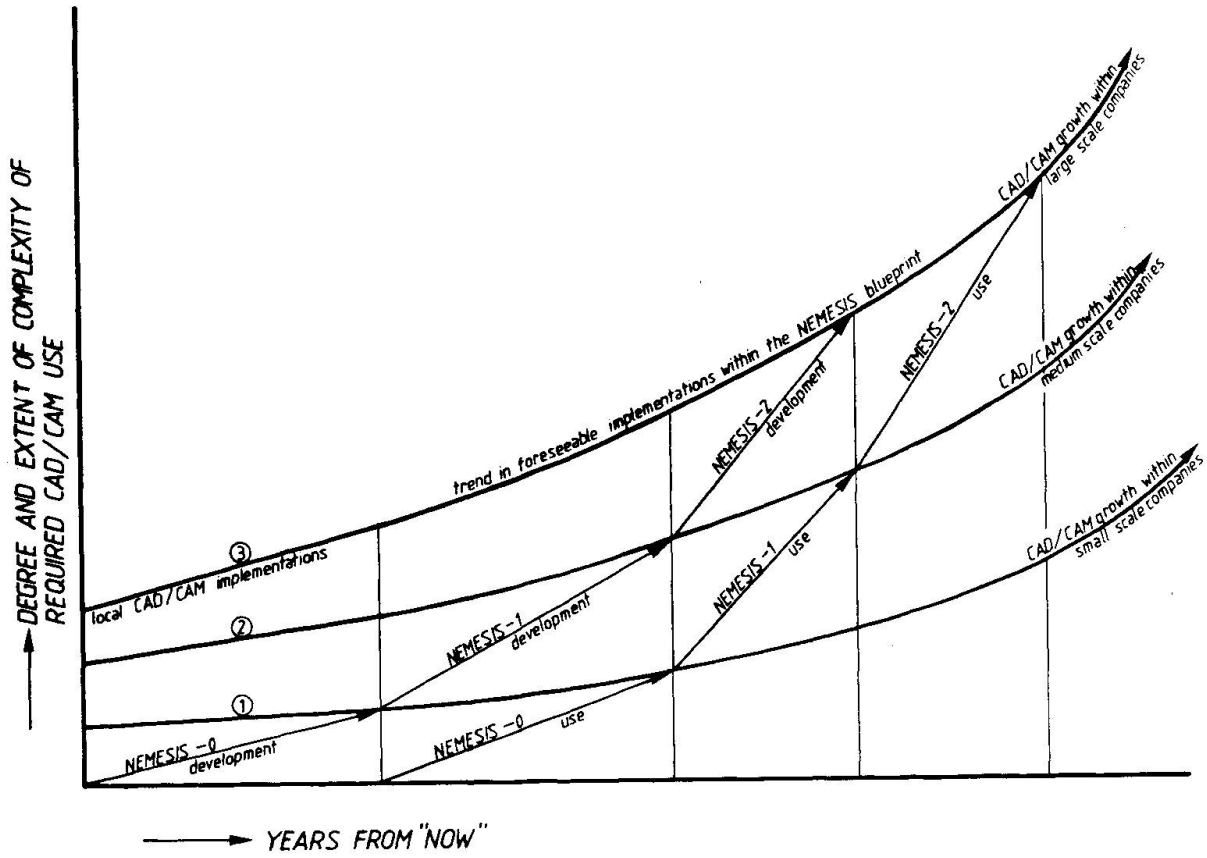

Fig.2 NEMESIS as a CAD/CAM catch-up manoeuvre

(1) large scale companies
(2) medium scale companies
(3) small scale companies

From a survey on the interest in CAD/CAM among Dutch industries carried out by a joint steering group of CIAD, Royal Institute of Engineers and the Netherlands Association on Informatics (INCAD-project) following figures resulted. The survey was done with a questionnaire sent out to 5309 engineering companies, representing about 16% of the total number of engineering companies in the Netherlands (estimated total number 31.285). The response was rather high. 15% of the questionnaires was returned, which already points at a relatively high interest of the companies on the subject. The interest of the respondents appeared to be differentiated according to

- interest in receiving adequate information 22%
- interest in consultancy (both providing and receiving) 29%
- interest in cooperative efforts 16%
- interest in governmental, financial facilities 22%

Expectations on CAD/CAM becoming of current interest (cumulative)

Expectations on CAD/CAM becoming of current interest (cumulative)

	CAD	CAM
- at this time	14%	8%
- within 2 years	23%	12%
- within 5 years	37%	21%
- within 10 years	51%	31%

From these figures it may be concluded that the potential market for CAD/CAM infrastructure software can be estimated at a minimum of about 0.51 x 0.15 x 31285 = 2400 companies. If about 10% of these companies can be stimulated to use the NEMESIS system it means that about 240 companies will have to carry the financial burden of the system development and maintenance.

The magnitude of engineering companies, expressed in number of employees, and response percentages are provided in the following table.

		Percentage of total number	Response	Interest in CAD	Interest in CAM
0 - 20	employees	84.5%	12%	28.2%	15.8%
20 - 50	employees	9.3%	24%	8.6%	5.5%
50 - 100	employees	3.5%	29%	4.1%	3.8%
100 - 500	employees	2.3%	55%	7.0%	3.9%
500	employees	0.4%	100%	3.0%	2.2%
		100 %		50.9%	31.2%

Related to the above mentioned group of 240 potential user companies the distribution would become

		number	percentage
0 - 20	employees	132	55
20 - 50	employees	41	17
50 - 100	employees	19	8
100 - 500	employees	34	14
500	employees	14	6
		240	100

More than half of the companies potentially interested in the system appear to have less than 20 employees.

A preliminary estimate of development costs is Dfl.10.10^6, which would mean a contribution for development costs of about ƒ 42.000,= per company together with yearly costs for maintenance and renewal estimated at 25% of the development costs resulting in about ƒ 10.500,= per year.
It must be borne in mind that these costs only relate to the NEMESIS infrastructure software so these costs have to be increased by costs of company dependent application software. The developers are convinced that the development costs for the infrastructure software are too high for small and medium-sized firms. As a result the Dutch government has been asked to bear these costs.

ENGINEERING USER ENVIRONMENTS

Computer programs are written for users. In the case of engineering programs the users are engineers. Engineers who want to solve technical problems with the help of computers. The program is a means of communication between the engineer and his tool, the computer. Next to the skill of problem solving, the engineer is expected to be a skilled operator of the program on the computer. He has to

'translate' his technical problem into the mathematical model the program is based on. In general the problem solving engineer using a program-tool is called 'engineering end user'. In order to do so, the program shall be presented to the engineer in such a way that he can easily understand the program. Which means, the program must fit into his intersubjective reality.
In some case the engineering end user is the program writer too. In that case he has entered the world of computer specialists. He has learned some programming language in which he can model the problem he wants to solve.
Because modelling a problem into a mathematical model, programming and testing it, will require quite an effort, it has become common practice to make the problem solving process more abstract, cq. the problem to be solved is generalized in such a way that similar problems fit in as well.
The more the problem-solving process is generalized, the more effort the engineering end user will have to put in at the moment he wants to solve his particular problem.
There grows a stress field between the costs of programming and the costs of problem solving. In administrative environments almost equal problems (e.g. salary computations) are solved over and over again with a program that is very near to the problem. In engineering environments specialist problems are not often repeated, problems have a tendency to be quite different each time the engineer wants to use the program. In order to spread the costs of program development, programs are therefore designed as general as possible.
The art of programming has become a skill on its own. Problem solving has been stated as an information process, which can be analysed, structured, systemized, programmed etc. The problem solving process of the programmer is program development. The result is a problem solving process for the engineering end user. The programmer, from the own nature of his work is familiar with the use of computers and thus searches for programming tools. For sake of systematics he is called 'the application programming user'.
The application programming user could make the problem solving tools he needs, but like it was with the engineering end user, new skills are required and this new expert user of the computer system is called 'the system developing user'.
In the context of the development of an engineering system, this user of a computer system is the last category taken into account. Otherwise we would end up with an integrated circuit designer, solving the design problem of an electronic component for a new generation of computers.
Aside the technical hierarchy there is the manager of the technical problem solving process. He is interested in the results of the process, which is not only the quality of the answer, but also the performance of the problem solving process. The effectiveness and efficiency of the process is (part of) his management task. In case of providing computer services as a separate commercial activity the performance of the problem solving process even becomes a major item. The manager could want to monitor and measure the performance of the system with help of the system and so he becomes a user himself, 'the system performance user'.

Up till now individual user types have been dealt with. One can think of several users of the same type forming together a user environment of that type.
However, in almost any case, there is a mixture of user types. Almost any user environment will have users of different user types.
In the case of the NEMESIS preparations, four different user environments have been distinguished as a more or less defined mixture of the individual user types mentioned.
This has been done because the system as a whole has to serve the user environment within a company. The system shall need adequate interfaces to the different user types and have an appropriate structure to tailor the system to the need of the user environment. The following user environments have been defined :

- engineering environment, where the engineering end use of the system predominates.(at least 2/3 of the system use by engineering end users)
- research and development environment, where the development of application programming together with engineering end use predominates (at least 2/3 of the system use by both user types, but not more than 1/2 in one of the groups)
- information processing environment, where the centre of gravity of activities lies in application programming together with system development. (at least 2/3 of the system use by both user types, but not more than 1/2 in one of the groups)
- service environment, where the performance management of the system predominates (at least 2/3 of the system use deals with performance processing. In case of a service centre the internal use is measured only).

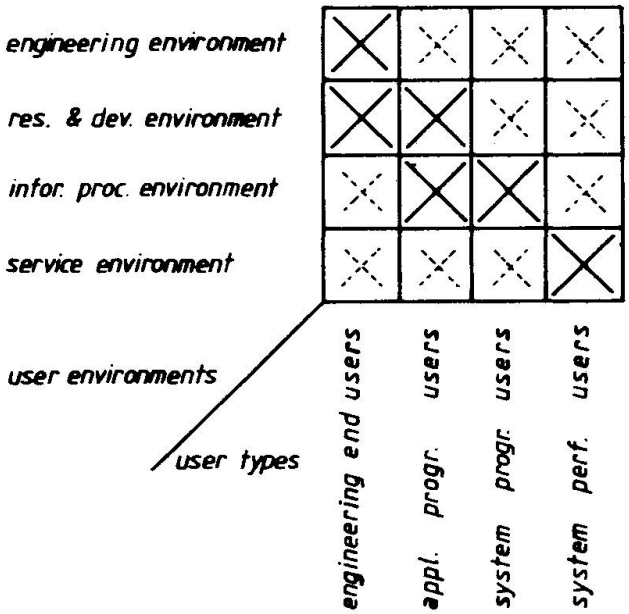

Fig.3 Matrix of user environments and user types

Environments can be subdivided in case a more detailed differentiation is required. One can think of the engineering end user environment split up in

- engineering management
- engineering design and analysis
- engineering administration (specs. cost estimates, planning etc.)
- engineering drawing
- engineering construction

etc.

Such a subdivision has a meaning only if the system provides special facilities to tailor the system to each sub environment.
Of course the definitions of user environments do not cover all possibilities but a preliminary inventory of user environments within the Association proved that all members could be typified unambiguously by one of the defined user environments.
In 1980 the number of members of the Association was 12,

- engineering environment 6 members
- R&D environment 4 members
- information processing environment 1 member
- service environment 1 member

USER REQUIREMENTS

An engineering system as is the case for any program shall be based on user requirements. User requirements, however, are quite difficult to collect, because a user cannot develop needs if he is not aware of bottlenecks in the existing situation and possibilities to improve.
Satisfied users have no needs and thus no requirements. Satisfaction is an integral acceptance of a situation without desires to change the situation. Ignorance of possibilities to improve often leads to satisfaction. The reverse approach is a general marketing strategy. Stress the benefits of new products in comparison with existing products and users will become dissatisfied with their present situation. The same applies to ideas. From the scala of available possibilities users develop desires and based on these desires more dedicated possibilities can be developed. It is an interaction between questions and answers between users and developers.
To this and the NEMESIS project group worked out a set of preliminary requirements and had these judged by existing user environments.
From dissatisfactions resulting from the use of GENESYS facilities in comparison with new facilities available on new technological products, more general requirements have been deduced :
The NEMESIS system shall constitute a system frame work that

1. is an effective aid to the various users
2. is adaptable to present and evolving needs and computer facilities of individual user environments
3. guarantees continuity of working methods, conventions adopted and investments made
4. provides a common basis for cooperation and exchange of information

From these main requirements a top down structure of more detailed requirements has been developed.

1. Effective aids to users

(A prelimninary subdivision under this heading has been made on basis of the user type specification from chapter 4).

1.1 Effective aids to engineering end users

1.1.1 The system must provide an easy-to-use interface between the user and the engineering problem solving aids.

1.1.2 The system must provide a systematically ordered range of day-to-day engineering problem solving aids.

1.1.3 The system must provide advanced engineering problem solving aids.

1.1.4 The system must provide convenient aids to store, handle and represent engineering data.

1.2 Effective aids to application programmers

1.2.1 The system must provide an easy-to-use interface between the user and the application programming aids.

1.2.2 The system must provide a range of aids for structuring and programming application programs.

1.2.3 The system must provide libraries of well-tested and documented modules to be used in application programs.

1.2.4 The system must allow R&D environments to implement advanced facilities at a preliminary stage of development.

1.2.5 The system must provide convenient aids to become trained in and have support on the use of the system and its programming aids.

1.3 Effective aids to system programmers.

1.3.1 The system must provide system programming aids.

1.3.2 The system must provide convenient aids to store, handle and represent the system and related documentation.

1.3.3 The system must provide convenient aids to become trained in the system and the system programming aids.

1.3.4 The system must provide convenient implementation aids.

1.3.5 The system must provide means to interface the system to other systems.

1.4 Effective aids to system performance users.

1.4.1 The system must provide an easy-to-use interface between the user and the the system performance aids.

1.4.2 The system must provide a systematically ordered range of management aids to manage a cost-effective use of the system.

1.4.3 The system must provide convenient aids to store, handle and represent data related to management aids.

1.4.4 The system must provide convenient aids to become trained in the use of the system and the management aids.

•

2. Adaptability to present and evolving needs and computing facilities of individual user environments

(A preliminary subdivision under this heading has been made on basis of hardware, software and working methods)

2.1 Adaptability to present and evolving hardware configurations.

2.1.1 The system must be portable over a wide range of computer configurations, ranging from relatively small to very large computer configurations.

2.1.2 The system must be adaptable to new hardware developments.

2.1.3 The interface between the user and the system must be brand-independent.

2.1.4 The system must provide upward compatibility in the interface between user and system, dependent only on the magnitude of the computer configuration.

2.1.5 The system must provide convenient and adequate aids to time an application program on a particular machine.

2.2 Adaptability to present and evolving software systems

2.2.1 The system must be adaptable to a wide range of operating systems, ranging from relatively simple operating systems to very complex systems.

2.2.2 The system must be adaptable to new developments in operating systems.

2.2.3 The system must be connectable to other software systems.

2.2.4 The system must be extendable by individual user environments.

2.2.5 The system must have an architecture that allows replacement of individual facilities.

2.3. Adaptability to present and evolving working methods.

2.3.1 The system must provide a smooth change-over from present individual working methods to system oriented practice at the individual pace of the user environment.

2.3.2 The system must provide adaptability to evolving working methods.

3. Continuity on working methods and investments

(A preliminary subdivision under this heading has been made on basis of maintenance, education, improvement, compatibility)

3.1 Maintenance.

3.1.1 The system must provide convenient and systematic means to maintain the system.

3.1.2 The system must provide convenient and systematic means to become acquainted with the system.

3.2 Education

3.2.1 The system must provide continuity for existing and future investments in engineering education.

3.3 Improvement.

3.3.1 The system must provide convenient and systematic means to improve the system.

3.3.2 The system must provide appropriate means to administer the sequence and status of improvement.

3.3.3 The system must provide the ability to recover earlier versions.

3.4 Compatibility

3.4.1 The system must provide upward compatibility in successive improvements in order to provide continuity to investments in education, data structures, program and skill.

3.4.2 The system must provide means to use existing GENESYS 2.6 subsystems.

4. Common basis for cooperation

(A preliminary subdivision under this heading has been made on the basis of standardization, exchange of experience and marketing).

4.1 Standardization

4.1.1 The system must incorporate present general standards and directed towards standards under development.

4.1.2 The system must correspond with existing GENESYS input-output conventions.

4.2 The system must be an adequate means to stimulate cooperation between users from different user environment.

4.2.2 The system must be an adequate means to be used by scientists and research workers in order to stimulate throughput of research know-how to practising engineers.

4.2.3 The system must be an adequate means to stimulate the exchange of subsystems, basic software modules, problem oriented modules and data structures between user environments.

4.2.4 The system must be an adequate means to be used by engineering education in order to prepare students for engineering practice.

4.3 Marketing

4.3.1 The system must be adequate means to be used by service organisations.

Based on these requirements an inquiry has been made among existing GENESYS user environments within the Association. The respondents consisted of 6 engineering environments and 2 R&D environments according the definitions provided in the previous chapter. Results are contained in [4] .
The priority sequences of the engineering environments and R&D environments are given in following schemes, where priority 1 means highest priority.

requirement	eng.env.	R&D env.
effective aids	4	3
adaptability	2	1
continuity	3	4
common basis	1	2

effective aids	eng. env.	R&D env.
eng. end users	1	2
appl.prog. users	2	1
syst. prog. users	3	3
system prog. users	4	4

adaptability	eng. env.	R&D env.
hardware env.	1	1
software env.	2	2
working methods	3	2

continuity	eng. env.	R&D env.
maintenance	2	2
education	4	1
improvement	3	2
compatibility	1	3

common basis	eng. env.	R&D env.
standardization	1	3
exp. exchange	2	1
marketing	3	2

From these priority schemes interesting conclusions can be drawn on the attitudes of the distinguished user environments. Engineering environments seek for unification, whereas R&D environments want to spread their knowledge. The relatively low appreciation by engineering environments for adaptability to working methods and education may result from the general unaquaintance with the process aspect of working methods and training facilities.
Next to the above mentioned requirement hierarchy a collection has been made of existing features which could be built into the system. Of course this would become a rather incoherent collection if no systematization was done.
Within the system five main functions have been distinguished and each main function is dubdivided into ten general subfunctions together forming what has been called a 'cabinet of functions'.
In each box required features can be placed, together giving a rather extended collection which is very useful in the design of specifications for an engineering

system to be newly developed. This 'checklist' has been provided in [3] .
The inquiry, mentioned before contained judgement of these features too.
The defined main functions are :

- run process management
  Through this function the user must be able to define and control the hardware he wants to work on and to monitor the system performance.
- training management
  Through this function the user must become aware of the capabilities of the system and become trained in the use of these capabilities.
- data management
  Through this function the user must be able to store, handle and represent data. For the engineering end user these data will apply to engineering data, for the application and system programmer to program data (statements, structures and program independent figures) and for the permance user to system measurement data.
- process management
  Through this function the user must be able to transform data sets into other data sets. (problem data in- and output) He must be able to monitor system provided processes of data transformation.
- documentation management
  Through this function the user must be provided by the system with up-to-date documentation (= explanation) on the system.
  Documentation data shall be coupled as much as possible within the software.
  Documentation in the form of paperwork, which cannot be generated by the system, must be kept to a minimum and be as imperturable as possible to modifications in the software in order to prevent incompatibility between software and documentation.

The defined subfunctions are :

- specification	(of what has to be done or stored)
- error control	(checking of specifications and relations to other specifications)
- monitoring	(of what is going on or stored)
- intervention	(in what is going on)
- modification	(of what has been specified or stored)
- deletion	(of what has been specified or stored)
- measuring	(of what is going on)
- protection	(of what has been specified or stored)
- security	(on what is going on or has been stored)
- representation	(of what is going on or has been stored)

In the scheme below priorities are shown as they have been indicated by the user environments

main function	eng. env.	R&D env.
run process man.	2	4
training man.	4	5
data man.	1	1
process man.	2	2
doc. man.	3	3

In the following schemes columns and rows above average interests are shown.

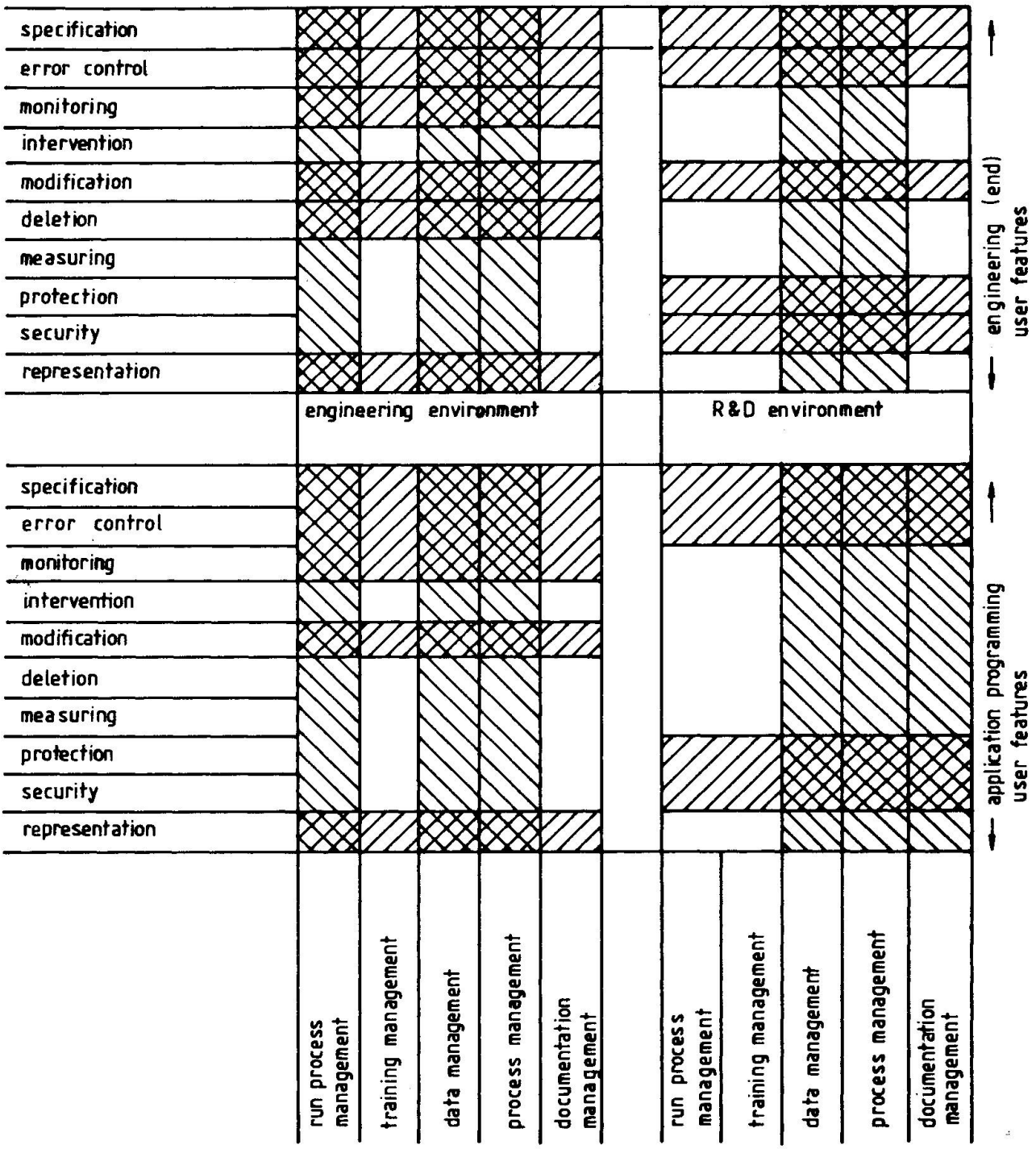

Fig.4 Main and sub-function making up a 'cabinet of functions'. Above average interests of users are shown.

NEMESIS is defined as a systematic frame work of mutually tuned technical facilities. As a result of specification of the individual facilities as extendable and in their components replaceable modules the system gets an 'open-growth' character.

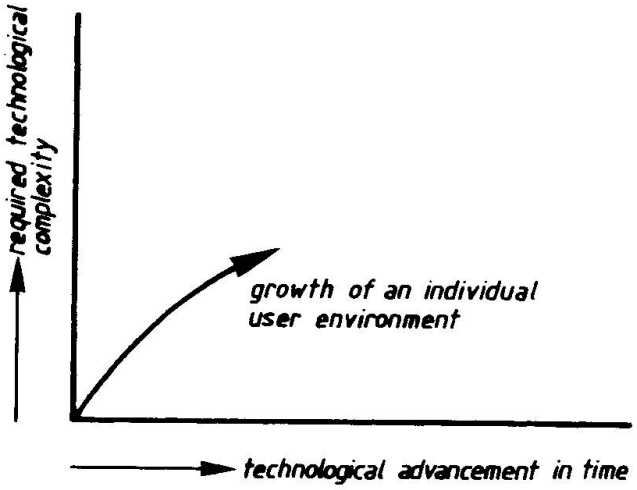

Fig.5 NEMESIS as an 'open-growth' system shall follow the growth of the user environment.

The NEMESIS system shall consist of

- a coherent frame work of specifications and standards for hardware components, their connection and monitoring to be used within the system
- a coherent frame work of conventions and standard for the architecture, integration and use of software components to be used within the system.
- an increasing number of hard- and software components, fitting in the above mentioned frame works and supporting the use of the standardized engineering tools within the system.

The novelty of NEMESIS will not be found in new components to be invented but in an integration of existing hard- and software components within a system of newly defined conventions and supporting software.
Main elements of the NEMESIS system will become

- micro-technology based on single chip processing components connected by standardized bus structures and providing standardized connections to support functional hardware components
- workstation concept, based on decentralization of intelligence.
  The NEMESIS workstation shall consist of all the hard- and software components required at that moment by a user for solving his problem.
  Basic hard- and software components of the workstation are

o a central monitor
o system management
o communication management
o application management
o operating system interface

o user interface
o network interface

The following scheme provides a preliminary architecture of the workstation.

- network-concept, based on integration of differentiated system use.
  Differentiation can take place internally and externally.
  Internal differentiation takes place as soon as internal functions are assigned to specialized system components with processing capacity independent of the central processor. One can think of specialized system components for data management, 'number crunching', graphics etc.
  External differentiation takes place as soon as external user interface functions are assigned to specialized workstations. One can think of workstations dedicated to the user by the user types defined in the previous chapter.

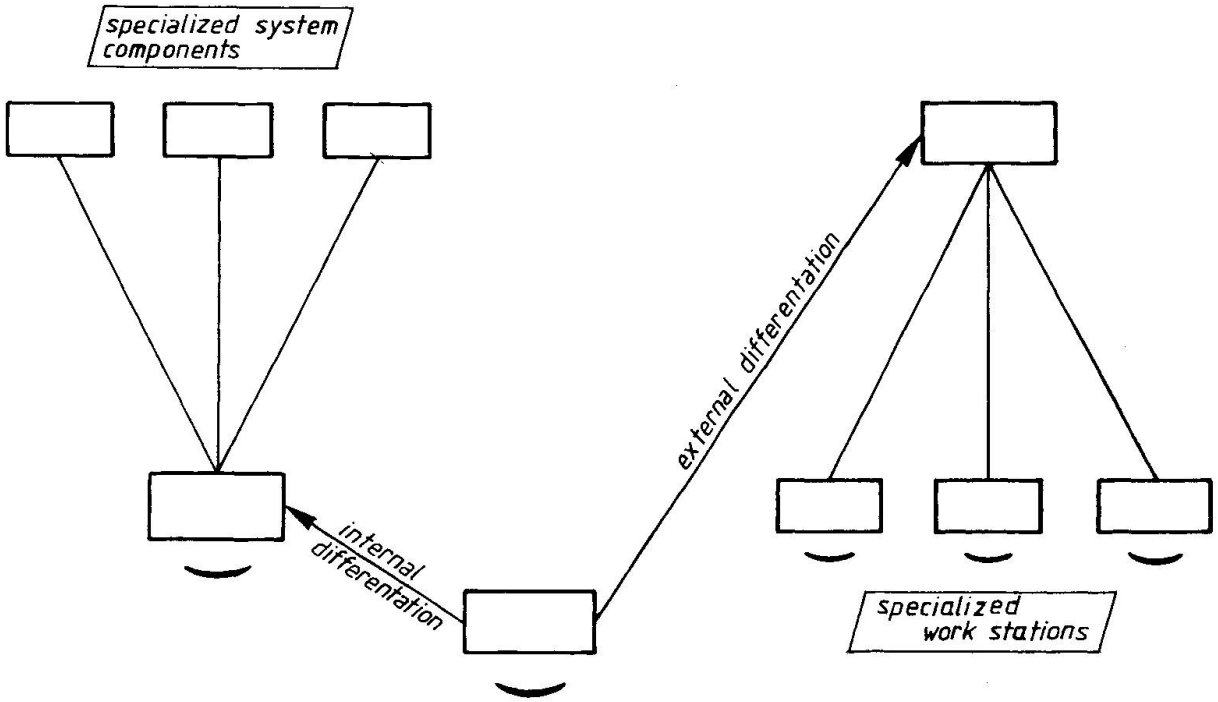

Fig.6 Internal and external differentiation of NEMESIS in future development steps

Networks tailored to the specific needs of individual user environments result from the required adaptability of the system.
The network shall provided possibilities to interface other systems, such as existing data base systems, measurement systems, industrial process monitoring system, etc.

- portable operating system concept.
  Main frame and mini computer manufacturers in the past designed their own hardware components and wrote their own operating system. Portability of application software was difficult because of the many different operating systems.
  Successful attempts to standardize programming languages have been made, but hardware manufacturers tried very hard to avoid portability by adding powerful programming features in order to seduce programmers to use non-standard language features.

Their main incentive was to bind the client to their products and provide continuity to their business.
The specialisation of micro hardware component industry changed the computer manufacturing market remarkably. Many small companies assembled micro computers choosing from available hardware components.
As the number of different micro-processors is small, system software, manufacturers can develop standard operating systems fitting to each of the micro processors. Well known operating sytems are CP/M on 8-bit processors and UNIX on 16-bit processors.
In order to avoid the many interfaces to individual operating systems as is the case with the GENESYS system and because NEMESIS is expected to be based on micro technology one of the available de facto standard, portable operating system will be chosen to base NEMESIS upon.

- modular architecture.
The modular architecture of NEMESIS shall make it possible to build highly specialized and user environment oriented systems with a minimum of environment dependent software. Companies derive the right to exist and survive due to the fact that they are specific and differ from others. As computer systems are tools to be used within specific manufacturing processes, the systems themselves must become specific by adding special and company-oriented or application software blocks. A schematic view on the modularity of NEMESIS is shown below (abbreviations according user-type definitions).

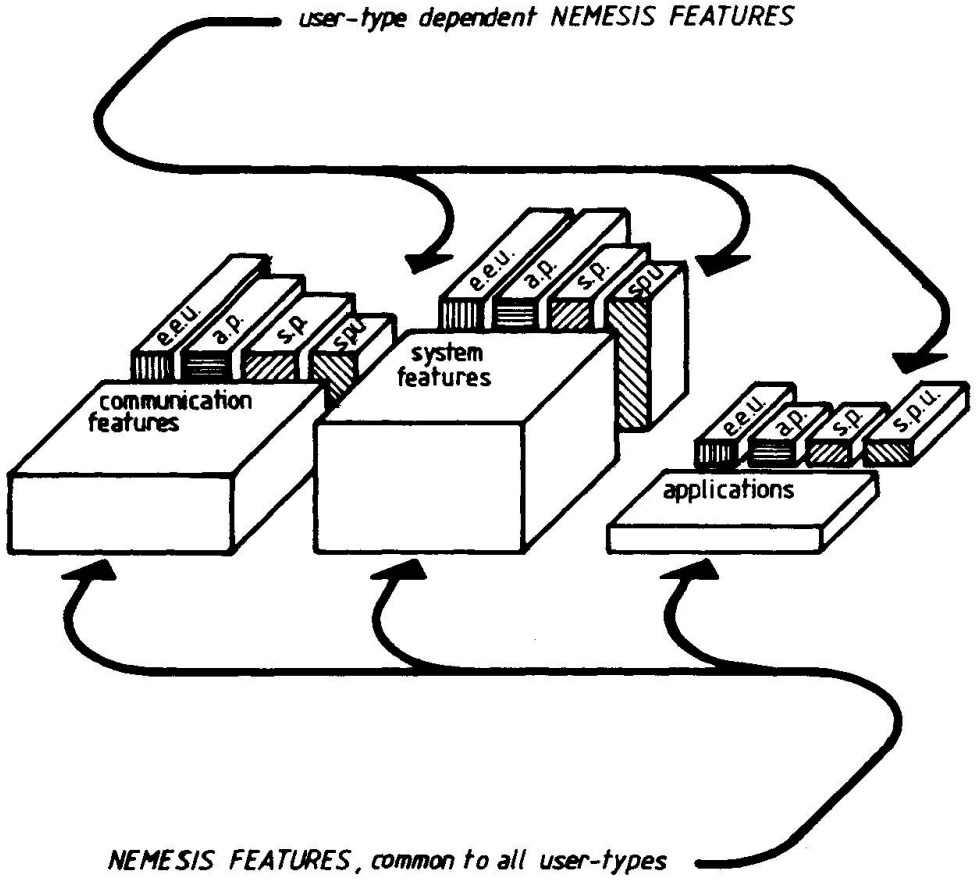

Fig.7 NEMESIS as a box of modular tricks

- standardized data concept.
  As NEMESIS is intended be used as a means of communication, a standardized framework for exchange of data is a prerequisite. Exchange of data will take place at many levels.
  - o between system (data base) and application programs
  - o between application programs
  - o between application program and user
  - o between users within one technical sector
  - o between technical sectors.
- standardized user interface concept.
  Many I/O devices have become available to the user to communicate with the computer. As the choice for a particular I/O device is often dependent on the problem solving process to be performed and depends on the taste of the user too, the user shall be as flexible as possible in this choice.
  Which means that the devices shall not have only a standard plug and a standard communication protocol. Each device must provide a standard data set in order to make them mutually interchangeable.

NEMESIS DEVELOPMENT PROJECT

As mentioned before, the NEMESIS development is planned to take place in several successive steps, each step ending in an improved product, where user-experience from the previously developed system is brought together with results of research into a system that aims at a higher application level.
In the figure below the development cycle of step 1 (i = 1,2,3) has been shown. For the start-up of step 1 there is a separate preparation step.

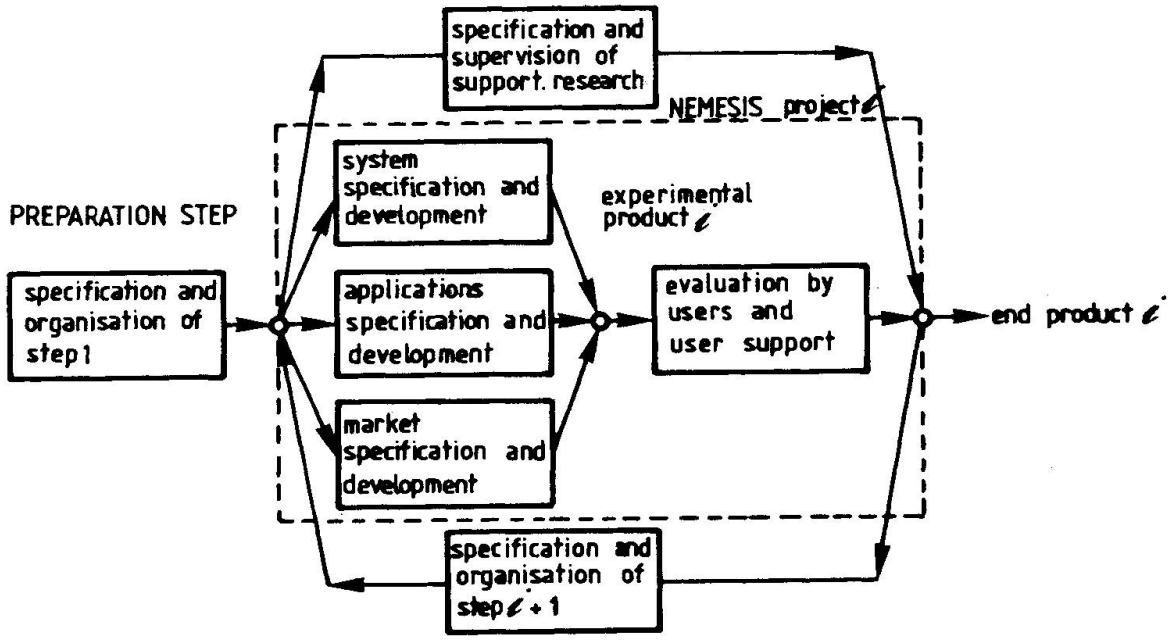

Fig.8 NEMESIS development as a project cycle with steps i (i = 1,2,3)

Gravity centres of successive NEMESIS development steps are

NEMESIS 0 : development and elaboration in workable concepts of standards and conventions, announcement of development objectives among potential users in the market, implementation of a confined NEMESIS system in generally available 8 or 16 bit micro hardware

NEMESIS 1 : implementation of advanced hard- and system software technology, extension of the NEMESIS system, internal and external differentiation of system components

NEMESIS 2 : implementation of communication concepts and differentiation of hardware and software concept, integration of differentiated user environments.

Each development step will be subdivided into a number of development phases, each phase ending in a milestone where the decision is taken to go into a next development phase

milestone 0	o	decisions to execute the development step
phase 1	↓	concept phase
milestone 1	o	approval of architecture and strategy
phase 2	↓	specification phase
milestone 2	o	approval of component specification
phase 3	↓	development preparation phase
milestone 3	o	approval of design and market strategy
phase 4	↓	detailed design phase
milestone 4	o	approval detailed specification
phase 5	↓	development system components
milestone 5	o	preliminary approval system components
phase 6	↓	development experimental system
milestone 6	o	preliminary approval experimental system
phase 7	↓	evaluation and user support
milestone 7	o	final approval on system for marketing
phase 8	↓	maintenance phase
milestone 8	o	decision to take system out of the market

A preliminary development organisation scheme is shown. It is the intention of the development organisation to let as many as possible development companies (who are also potential users of the final product) take part in development work.
This decentralized approach has been chosen in order to foster user affections at an early stage (create an intersubjective user environment) although this approach aggravates the tasks of projectmanagement.

SUMMARY AND CONCLUSIONS

The initiative to the NEMESIS system development is a result of close cooperation of computer users in engineering who started and extended their cooperative efforts on basis of the use of the GENESYS system.

The NEMESIS system development will be realized in several steps, each step ending in a technologically improved system at a higher infrastructure level.

The NEMESIS system development is a means to catch up with CAD/CAM developments in other industrialized countries, and shall be an incentive for many other developments (especially in the application area).

The NEMESIS system is intended to be a means of integration of many already available hardware and software components based on standards and conventions to

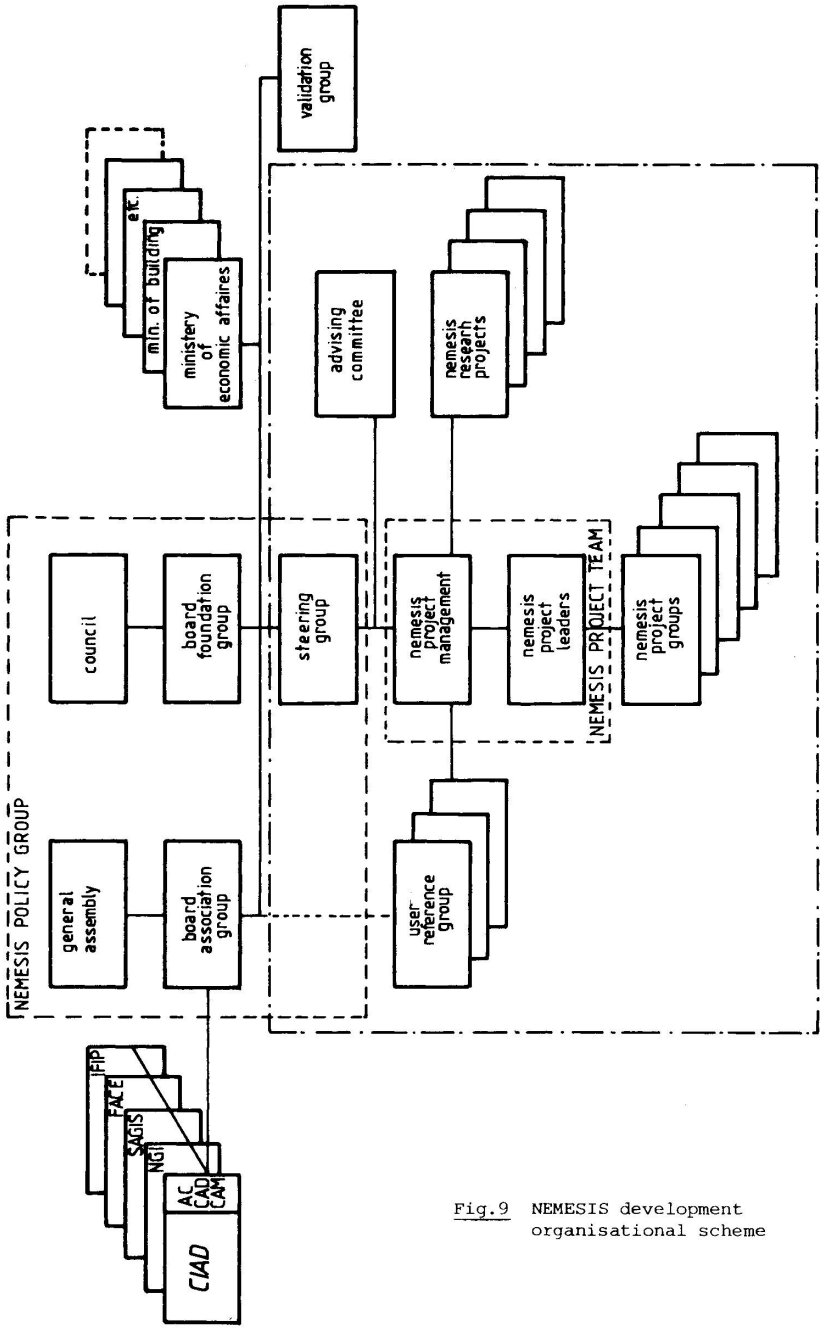

Fig.9 NEMESIS development organisational scheme

be newly developed by larger user groups. As such the NEMESIS system may act as a concentrator for users who are confronted with a dazzling stream of new hardware and software products on the market, which they cannot evaluate each on its own.

From a survey on CAD/CAM interests among engineering companies in the Netherlands, it appeared that more than 50% of the companies that returned the questionnaire (which was 16% of the random sample) expects CAD/CAM techniques to be implemented in their organisation within 10 years.
From these results the conclusion may be drawn that there is enough potential interest for the NEMESIS system as a CAD/CAM infrastructure system to justify its development.

ACKNOWLEDGEMENT

The author wants to stress the fact that the NEMESIS ideas and further elaboration contained in this paper are the results of close cooperation within the project group NEMESIS of the Association and many others. The members of the project group are Messrs. J.H.A.E. Amkreutz, S.I.E. Blok, N.T. van Harpen, W.Nijenhuis and F.P. Tolman, and the author.
The Boards of the Association and Foundation, the staff of CIAD, the CAD-steeringgroup and representatives of the Ministries of Economic Affairs, Housing and Public Works and Waterworks have contributed to the project too. Each from their own point of view. The author expresses his gratitude to all participants in the project for the opportunity to present the NEMESIS ideas at an international forum.

REFERENCES

1. Reports VGN2 and 2.1 : Inventory of bottlenecks in the use of the GENESYS system, 1978 (Dutch)
2. Report VGN-4.3 : Preliminary quality classification criteria for GENESYS, 1981 (Dutch)
3. Report VGN-10 : Qualitative approach towards a new, user environment oriented engineering system, 1979 (English)
4. Report VGN-10.1 : Inquiry on user requirements for a new, user environment oriented engineering system (NEMESIS) 1981, (English)
5. Report VGN-10.2 and 10.3: Concept of NEMESIS, a new engineering system, volume 1 and 2 , 1981 (English)
6. Report VGN-10.4 : Investigation of the user-friendliness of a GENESYS subsystem in relation to other similar programs, 1981 (English)
7. Report VGN-10.5 : Inventory of research projects and development groups which could support the NEMESIS system development, 1982, (Dutch)
8. Report VGN-10.6 : The technology of NEMESIS, 1982 (Dutch)
9. Report VGN-10.7 : NEMESIS development strategy, 1982 (Dutch)
10. Report VGN-10.8 : NEMESIS project organisation, 1982 (Dutch)
11. NEMESIS nota 1 : Policy report 1 "NEEM NOU NEMESIS..." from the oards of VGN and SGN, 1980 (Dutch)
12. NEMESIS nota 2 : Policy report 2 "NU NAAR NEMESIS..." from the Boards of VGN and SGN, 1982 (Dutch)
13. Samenspel, an information brochure for Dutch industry on CAD, 1981, published by the CAD steering group (Dutch)
14. Results of an inventory on CAD/CAM interests in Dutch industry, 1981, CAD-steering group, unpublished. (Dutch)
15. Feasibility study of common I/O conventions for the building industry, EEG predevelopment study, RIB, Germany 1982 (English)
16. The performance specification of a workstation for the building industry EEG predevelopment study, CICA, Great Britian, 1982 (English)
17. The automation of draughting work, FACE report nr. 2 1982 (English)