State-of-the-Art

ARCADE – An Architecture for CAD in Electronics *

J. Vlietstra
AT&T and Philips Telecommunications B.V., P.O. Box 1168, 1200 BD Hilversum, The Netherlands

Ever since the introduction of computers in development and engineering departments, design activities have been supported by computer methods in a growing number of design functions, in the interest of reducing development time and cost, and increasing the quality of the products. Computer Aided Design (CAD) – a generic term for these computerized tools – is recognized as a method for drastically reducing the development time of industrial products.

CAD methods are currently being used during each of the various phases in the development activity. The introduction of these CAD tools for the various stages in the development cycle, however, was to a large extent effected in isolation, i.e. not as a "subset" of an overall concept. Lack of such a concept, with its inherent disciplines and compatibility aspects, has given rise to sub-optimal efficiency and effectivity. To remedy this situation, a generalized architecture for CAD has been developed. This architecture drastically reduced the large variety of input methods, combines all product data in one well-structured product file, supports all necessary design functions, assures the proper generation of technical documents, and above all, facilitates maximum efficiency by integrating the processes for various stages in the development cycle.

The essentials of ARCADE will be described in this article, together with how the system is designed.

The "ARCADE" way and method of working in an organization in which development, engineering, manufacturing, logistic and documentation departments are concerned with technical automation issues will, as such, be discussed.

Benefits will, of course, also be elucidated.

Keywords: CAD, computer aided manufacture, design, synthesis, analysis, detailing, layout, documentation, electronics, engineering, data processing, graphics processing, GKS, CORE, interface, integration, integrated factory, engineering office automation.

North-Holland
Computers in Industry 5 (1984) 3–20

1. The Evolution of CAD

Computer Aided Design (CAD) activities were introduced in design and engineering offices soon after the introduction of the first industrial computers in 1951 [22]. The engineering environments which most benefitted by the advent of computer technology were the avionic and the electronic engineering environments. It became possible for these two fields of application to replace the tedious and difficult hand-calculation methods with procedures and algorithms that could be solved using the first electronic computers, providing more accurate results than ever before.

The avionic industry was the very first to benefit from CAD techniques by utilizing programming methods to calculate the loads and stresses in the essential parts of an airplane. The finite element analysis techniques [10] used in such calculations thus became the backbone of many design

Jakob Vlietstra joined the Research Laboratories of the N.V. Philips' Gloeilampenfabrieken in 1960 after a nine year career as Deck Officer in the Merchant Navy. During the period 1960-1965, he studied Numerical Analysis, Programming, and Process Analysis at the Technical Universities of Eindhoven and Delft, and at the Amsterdam Mathematical Centre. In addition, from 1961-1969, the author was responsible for software development for numerically controlled machine tools at Philips, and was a member of a small research team that realized a project wherein the design and manufacture of television tubes were carried out with the aid of computerized processes. From 1969-1973, Jakob Vlietstra investigated the fundamentals of Computer Aided Design. In 1973 he became System Architect and Concern Coordinator for Computer Aided Design in Electronic Engineering applications. He was appointed Assistant to the Director of the Product and Factory Engineering Department of Philips Telecommunication Industry in 1981 and, recently he joined the newly created company: AT&T and Philips Telecommunications in which he manages the CAD and CAM departments. Jakob Vlietstra chaired IFIP's Technical Committee on Computer Applications in Technology from 1977-1983 and he received the IFIP Silver Core Award in 1977.

0166-3615/84/$03.00 © 1984 Elsevier Science Publishers B.V. (North-Holland)

methods which would never have been applied without the fast calculation abilities of electronic computers. In the very same industry the analysis of structures was soon to be followed by the application of computer methods to model shapes and bodies of aircrafts [4]. The combination of this surface modelling technique with the structural analysis methods provided the basis for the fantastic speed with which the aircraft and missile industry leaped forward.

Not so much later, the first CAD techniques were modestly applied in the engineering discipline that had itself given birth to the means which were used by the avionic industry – electronics. The first methods were developed to calculate the behaviour of analogue circuitry. The advent of digital techniques soon led to the first simulation programs which could analyse the correct design of this type of product.

Similarly to the development in the aircraft industry, other applications were rapidly introduced in the electronics industry. The publication of Lee's algorithm [9] opened the way to replace the tedious and time-consuming layout activities by computer-supported aids.

The two examples cited above were put into practice in the very first 10 years of the existence of computers in industrial environments. In the decade that followed, the methods were refined, the algorithms improved, other application areas were developed, and the constant improvements in the speed and memory capacity of computers made the processing faster and more efficient.

The development of computer graphics [17] added capabilities to the CAD methods to such an extent that CAD activities were no longer confined to laboratory type of work, but were even being introduced in drafting offices.

And with the introduction of CAD into the drafting world, together with the subsequent success of the so-called "turnkey" suppliers of CAD-equipment, a small drama started.

The orientation fixed on the possibilities of replacing conventional drafting methods with more (and sometimes very) advanced techniques created a culture which dominated (and in the opinion of the author still dominates) CAD practices far too much. It is not to be disputed that the introduction of computer graphics into drafting practice yielded very striking results. But what happened was a degradation of the very essentials of the complete design process itself. Hence, all actions were directed towards drafting and detailing, and far too few CAD experts focused on the other phases in the design process. Thus we here itemize six important phases in the design process:

- the specification phase
- the design synthesis phase
- the design verification activities
- the detailing actions
- the test and production-preparation work
- the composition of the product documentation.

Each phase is equally important and regrettably the very essential phases of product specification and product documentation have hardly been considered as CAD activities where at least as much can be earned in time and capacity, as is being earned in the detailing phase. [1] It may indeed be argued whether advanced and computer interpretable specification techniques are defining the most important aspects of all that which is called CAD.

The advent of the VLSI era will show that without considering CAD for the VLSI design process in its totality, the complexity of such circuits will never be mastered.

With this introduction the author has tried to describe the important aspects of CAD. Each of these will be referred to in the following sections and short explanations will be provided where necessary.

2. Deficiencies in the Current CAD Approaches

In the Introduction, six important phases in CAD were defined. Especially in the design verification phase, the detailing phase, and the activities that lead to data which can be used in the factory and the test departments, a large variety of CAD application programs have been created. In the electronics industry the number has grown to such an extent that an inventory and an appropriate classification of all the programs have virtually become impossible. In the area of simulators which simulate the electrical (or analogue) behaviour of a circuit alone, hundreds of programs exist.

Most programs have been designed and developed by the larger corporations. A fair number

[1] All drafting work, including the layout process in electronic engineering is classified as "detailing." This is done to maintain the same notation over a large number of engineering disciplines.

have been developed at universities and a number of those university CAD programs have been introduced into many industrial organizations. The most successful in this group of CAD programs are (per design phase) outlined below:

- in design verification
  ECAP [3], SPICE [13], ASTAP [7] (electrical simulation),
  TEGAS [18], HILO, CASSANDRE [2], LASAR [14] (logic simulation),
- in design detailing
  REDAC, APPLICON, CV, SCI-CARDS for PCB layout [21] CALMA, GAELIC [12] for LSI layout,
- in test preparation work
  LASAR for test pattern generation.

The problem, however, is that each of the more widely-known programs, as well as many less well-known programs and programs, for example, developed in-house for specific purposes, exhibits unique features not present in the other programs. In quite a number of design departments it is also noticeable that the younger engineers introduce programs they learned to use in their university environment. In the author's organization this and the lack of applying proper data processing techniques created a situation in CAD that called for a drastic revision of the system.

Both aspects will be discussed in the following subsections.

2.1. The Gap between CAD Users and CAD Programs

CAD programs are usually developed by persons who have had their education in "Computer Science" or "Informatics." In general, a wide gap exists between the typical community of electrical and/or electronic engineers and those who are responsible for the creation of CAD programs. Despite the fact that the electronic engineering community (the users) has developed its own symbols and notations, as well as specific standards and norms, within many organizations when new CAD programs for the electronic engineering community were developed, new notations were introduced. This was and still is commonplace. In many cases the standards laid down in the charts and tables of the IEC have practically nowhere been taken into consideration. This actually means that on both the data entry side and the report or output generating side of the CAD programs, the contact with the real world has not been accommodated – not even conceptually!

This all results in the current tragic situation of tens (if not hundreds) of different input languages all serving the very same purpose: to describe the functions and interconnections of an electronic network.

At the same time it must also be observed that our Standardization Institutes are too slow to react properly to the changes in technology. The "Tower of Bable" effect has been very visible ever since the introduction of computer languages and it was to be expected that a similar situation would evolve in problem-oriented languages. It was only in the specific area of Numerically Controlled (NC) Machine Tools that an early standardization effort prevented the emergence of a multitude of NC languages.

As a consequence, it is impossible today to acquire a program where the input language can be called "standard." And if an organization exploits more than one CAD program, more than one input language (the exact number being the number of CAD programs) must also be introduced.

On the output side the situation is slightly better when it concerns the production of drawings. The influence of International Standards has forced the manufacturers of CAD programs to adhere to those standards, albeit that these standards are not always interpreted in the same way. When it comes to producing documents or results for which standard forms do not exist, or of which it is not widely known that standard forms could be applied, one observes the same chaotic situation.

Thus, many companies have been forced to write specific input pre-processors and output post-processors to work from a standard data-entry format and to work to the internally existing output forms respectively.

2.2. Deficiencies in CAD Program Structures

As mentioned earlier, CAD programs have been developed for specific CAD functions in isolation and none of these programs is designed to receive or forward information respectively from and to other CAD programs used for different design activities. In addition the way the information is stored in almost each of the existing CAD programs has a properly designed data pool, in which all relevant design data and the results generated by the programs are stored in a well-structured way.

This can be illustrated by the fact that data stored in a program which can perform the placement of components on a Printed Circuit Board (PCB) and has to be transferred to programs which need such placement data, i.e. for routing purposes or test purposes, must

- either be transferred manually, or
- must be transferred by means of a special purpose transfer program.

The total design process, as indicated in the Introduction, exhibits many phases and it has been explained that each of these phases knows a large number of special purpose CAD programs to perform certain given tasks. In the event that it is difficult or even impossible to have access to results that have been generated in a given program and stored in a non-structured way, the possibility of incorporating such a CAD program in the total design process is negligible.

2.3. Deficiencies in the Software Engineering of CAD

One of the principles of software engineering, which is hardly ever applied in CAD itself is the "engineering" part, that is the approach of using *standard* software tools in the development of software [15]. That lack of appropriate software engineering in the development of CAD techniques and programs is most noticeably demonstrated through the total absence of the application of standard data processing techniques. This observation also holds for the fact that in many cases good programming practices are missing as well.

Yet, many data processing techniques have to be applied, and close investigations of the very nature of a large number of CAD programs have indicated that as much as 60 percent of the total effort in constructing CAD programs is spent on the (renewed) development of certain techniques which were definitely already available. This actually means that only 40 percent of the work in developing CAD programs was used in the construction of algorithms. The heart of the intelligence of CAD is embedded in such algorithms and it is deplorable that too much energy and valuable resources have been spent, and are still being spent, in reinventing the wheel!

Some very useful examples of generally available data processing techniques and tools that could have been used by the designers of CAD-programs are:

1. Language Processing Capabilities
2. Graphic Processing Techniques
3. Data Base Management Techniques
4. Document Formatting and Composition Tools.

These standard software engineering vehicles are discussed in what follows.

2.3.1. Language Processing Capabilities

A reasonable number of so-called *problem oriented language processing* programs are available. With such programs it is possible to easily define the syntax of the language (which is to be the user interface to the CAD program) and to construct a number of routines in which the semantics (the meaning) of the input language is defined and elaborated. Despite this, in almost all cases, CAD programs exhibit language processing features that were developed for the given program only.

The proper use of a generalized language processor shows two major advantages:

- The definition of the input language is rigorous and the development of the processor itself, at least an order of magnitude faster.
- In addition, changes, i.e. on the request of the user, can be made easily and a new input processor is generated almost immediately.

The disadvantage is often argued to be the fact that the input processor, created in this way, is slower. But, it should be borne in mind that the input process is only a minor portion of the total CAD job to be carried out by the CAD program. The most important function of almost every CAD program is carried out by its algorithms which will aid the designer in solving his or her design problems.

2.3.2. Graphic Processing Techniques

The same arguments as mentioned in the previous subsection are also valid for CAD Graphic Processing, as graphic features are also frequently developed from scratch. But here also, a large number of generalized graphic facilities ar available and can be applied. Especially the developments in the U.S.A. [5] and the German Federal Republic [6,8] which have led to the standardization of a graphic processing system by the International Standardization Organization (ISO) should not and cannot be ignored by all who are involved in the construction of modules in CAD which have to cater for the entry of graphics data and the output of documents that contain graphic figures or symbols.

2.3.3. Data Base Management Techniques

Here also, instead of using available data base management techniques and/or adapting the available systems to make them suitable for the use in CAD, many CAD programmers developed their own data base management and filing techniques. Especially in the specific area of generating data processing methods, a tremendous amount of redundant work has been and is still being carried out. The arguments that were used again focused on the issue of performance. Currently available data base management systems which were developed for the specific "business" applications do lack a number of features necessary to exploit them successfully in CAD. First of all, large amounts of data that have to be searched for and retrieved as "bulk" information must be catered for. It appeared, however, to be easier and more efficient to adapt commercially available data base systems than to construct entirely new ones.

2.3.4. Document Formatting and Composition Tools

The result of any CAD process has to be documented. Despite this, documentation has been an extremely neglected area of interest for the CAD author. The advent of word- and text-processing techniques has sidetracked many CAD programmers into writing their own text-processor. And, it will not come as a surprise to the author that until the issue of document layout, creation or document-formatting is raised as an important issue for CAD, will attention be focused on that specific subject. Without the need for really doing so. Almost every manufacturer of advanced computing equipment can provide their users with good text editors and reasonably good document composition and formatting tools. Besides that, a fair number of reasonable to excellent document formatters are available and although the price of acquiring such systems may be high, the development of an own formatting system will be many times higher.

2.4. A Basis for Further Activity

Each of the observations made in the preceding subsections were important considerations for constructing a CAD architecture in which an attempt has been made to avoid both the deficiencies in the preceding and to avoid duplicative efforts in constructing novel data processing techniques.

3. An Architecture for Computer Aided Design: ARCADE

In the following a description will be given of the "ARCADE" concept and its practical realization.

It must be stated at the beginning that ARCADE does not in itself contain so-called CAD programs or CAD tools. Thus, when referenced, the layout- or the simulation programs proper are meant. The concept of ARCADE is based on solving the deficiencies described in Section 2. ARCADE, however, provides a general solution for employing each and every CAD program which is deemed worthy to be used in a given industrial environment.

3.1. System Considerations

Every information processing activity can be visualized as shown in the following schematic representation (Fig. 1), a stream in which input data is subjected to a process that converts the input data into data or information that must be outputted. This fundamental action, where the process can be seen as a "transformer" of the input data, can have important features such as:

- Storing and retrieving information in a special data archive; this data is not the output data but information that may be stored temporarily to be accessed and used later.
- In so-called "human-machine interactive" environments the process can be interrupted and controlled by the human.

In CAD programs this information processing principle applies also. Design data is entered in a computer; algorithms perform certain functions and transformations on this data; information is stored in a file for later or suspended use; the designer may add information or control the transformation process ineractively and the process is concluded by generating an output stream which can be printed or displayed ("hard" and "soft" copies).

In what follows, the generalized approach to the ARCADE input, output and processing aspects is discussed.

3.1.1. ARCADE Input and Information Storage

Every CAD process starts with the preparation of the design data that will make it possible to walk through the earlier mentioned design phases. The basic information for electronic design is the definition of the network of the circuit or system being designed. The network manifests itself in two ways:

- The functional network, i.e. the way electrical or digital functions are denoted and connected (for example, a network of NANDs), and
- The network of the product, i.e. the way components are positioned and interconnected on a Printed Circuit Board.

Both representations are depicted in Fig. 2.

The most important of these representations is the "functional" manifestation of the network. The various nodes (functions) in the network are location-independent. The importance is the unique definition of the function (transistor, NAND, flip-flop) and the interconnections. This information will be used by every CAD process, be it layout, simulation, test pattern generation or documentation. The "product" manifestation is important for the actual realization of the circuit or system.

Together with other important parameters the functional and product network descriptions define the input part of the ARCADE system. This input is now uniquely defined in the ARCADE input process.

Two modes of entering this information in ARCADE are available within the Standard ARCADE Interface Language (SAIL):

1. *Textual.* An alphanumeric problem oriented input language, based on a concept developed at Stanford University [19], is available for alphanumeric text. With this input language which is a subset of SAIL, called the Network Description Language (NDL), the network descriptions can be coded and interpreted by a SAIL language processor. The information is then converted into a specific format that can be used by other processes in the ARCADE system.
2. *Graphical.* In SAIL graphical mode, the designer can digitize a hand-drawn schematic diagram and enhances this "network" information with additional parameters – through the use of interactive displays or work-stations – that cannot be easily entered with digitizers. The information entered in this way is also converted to that same format mentioned under Point 1.

The "format" referenced to the two input modes is a part of what in ARCADE is called the POLI-interface. The Problem Oriented Logic Interface (POLI) contains a number of standard formats. The information in the network (functional and

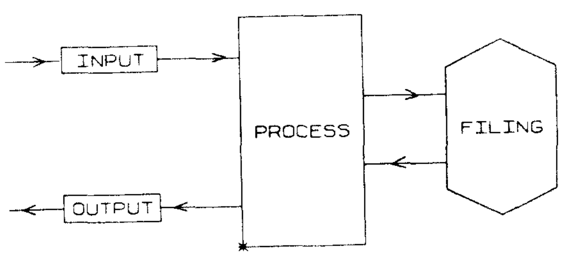

Fig. 1. An Information Processing Flow.

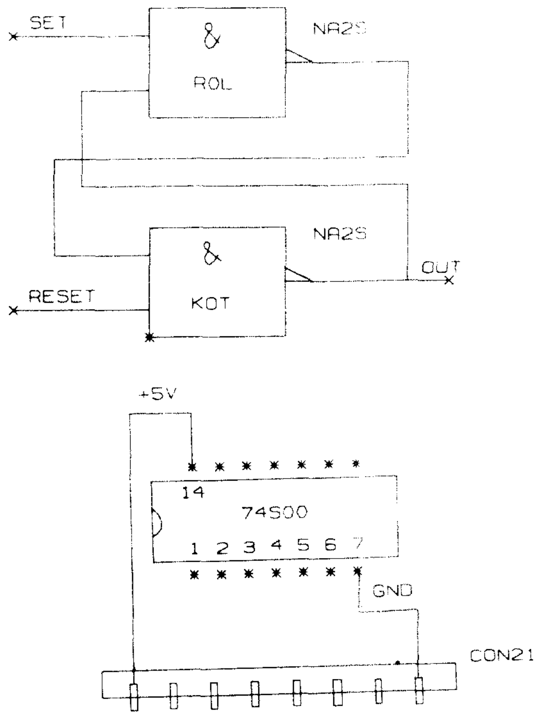

Fig. 2. A Functional and Product View of a Network.

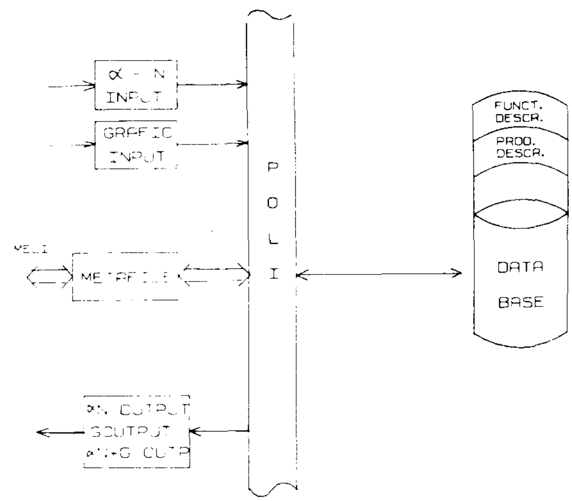

Fig. 3. Schematic of Part of the ARCADE System.

product) is converted to the POLI-format by a program. Since this program converts the network data into the POLI model it is called the *Network Modelling Technology* program. (In subsequent parts of this paper the notion "*Modelling Technology*" will appear again.)

The information in the network, converted in this way to the POLI-format, can then be stored in the ARCADE data base for later usage. It is important to stress that the POLI-format is a well-structured format which constitutes the backbone of ARCADE. The network POLI description can be transferred to the data base in such a way that all parts of that information can be accessed quickly and easily. This means that the POLI-format, which consists of a serial string of data elements and their interrelations will be converted from POLI-format in a differently-structured data base format.

The representation of Fig. 1 is now more precisely visualized in Fig. 3.

3.1.2. ARCADE Processing

The actual processing in ARCADE is executed by the already existing CAD application programs such as SCI-CARDS, PRINCESS, PHILPAC, LASAR, GOAL, etc.

To this end existing CAD programs are extended with a program that can read the network information in POLI-format and convert it into the format of the particular program that will be executed. The reverse process can also be carried out. In addition, the information generated by the various CAD programs can be translated into other sections of the POLI-format. For example, the information about the analogue electrical behaviour of a circuit, as can be calculated with PHILPAC, is translated by a "*analogue behavioural modelling technology*" program into another section of the total POLI-format definition. The various CAD programs will thus accept and/or generate POLI-formats depending on the respective CAD functions which they perform.

To be even more precise, the extra software that makes it possible for a layout program to execute its process in an ARCADE environment, must be capable of handling the network modelling technology and the "*layout modelling technology*". The latter translates the geometric information generated by the layout program to the part of the POLI that handles geometric information in a special format. Hence, the program that links a layout program such as SCI-CARDS to ARCADE must be able to read and write a number of modelling technologies in POLI-format.

At this point a new notion will be introduced. As discussed, POLI handles information which can be identified as "character-string" oriented. In the example above one is suddenly confronted with graphic (or geometric) information. For that reason the Logic Interface is subdivided into two related interfaces:

1. POLI, already introduced, and
2. Graphic Oriented Logic Interface (GOLI).

Instead of referring to the graphic oriented formats as a part of POLI, GOLI-formats have been categorized as an extension of POLI. The reasons for doing so will be explained in this section.

Programs which will only use the POLI-format are, for example, the simulation programs. The PHILPAC-link program must handle the network modelling and the analogue behavioural modelling technologies. That is to say, the PHILPAC-link must read the related POLI-formats and translate these into PHILPAC network statements. The results of PHILPAC must be converted by the PHILPAC-link program into the POLI-format that handles the analogue behavioural modelling technology data.

If, however, PHILPAC had been equipped with the ability to generate graphic pictures (e.g. histograms) that information would have to be converted into a special GOLI-format.

For logic simulators and test-pattern generators, a digital behavioural modelling technology and a related POLI-format is required.

The now complete ARCADE operations and functions are shown in Fig. 4.

When all available CAD programs have been "linked" to ARCADE in the way described above a very powerful system has been created. Of course data can only be transferred via the POLI-GOLI concept, but that can as such be seen as the "CAD-Bus".

In addition the links may have to be extended to accept information other than "network modelling technology" information. A router which is more advanced than its predecessors should be able to accept routing information stemming from them for further processing. Hence, such router programs should also have the facilities in their link-programs to read and understand "layout modelling technologies."

A final observation is justified at this point. Much conversion has to be carried out in the various link-programs. However, if the POLI-GOLI is well established and well accepted, the new developments in CAD software can use the formats of POLI and GOLI directly in their input and output modules. A large amount of processing dedicated to translating the POLI-GOLI formats back and forth can then be avoided.

3.1.3. ARCADE Output and Information Retrieval

In the previous section, how the POLI and GOLI interface is capable of handling various "modelling technologies" has been elucidated. Each of these technologies exhibits parts of the information about the circuit or system under development. When sufficient tools (CAD programs) are available and linked with ARCADE, the total information about the product is nearing completeness.

This information will be passed to a data base where one part contains all the data for one specific

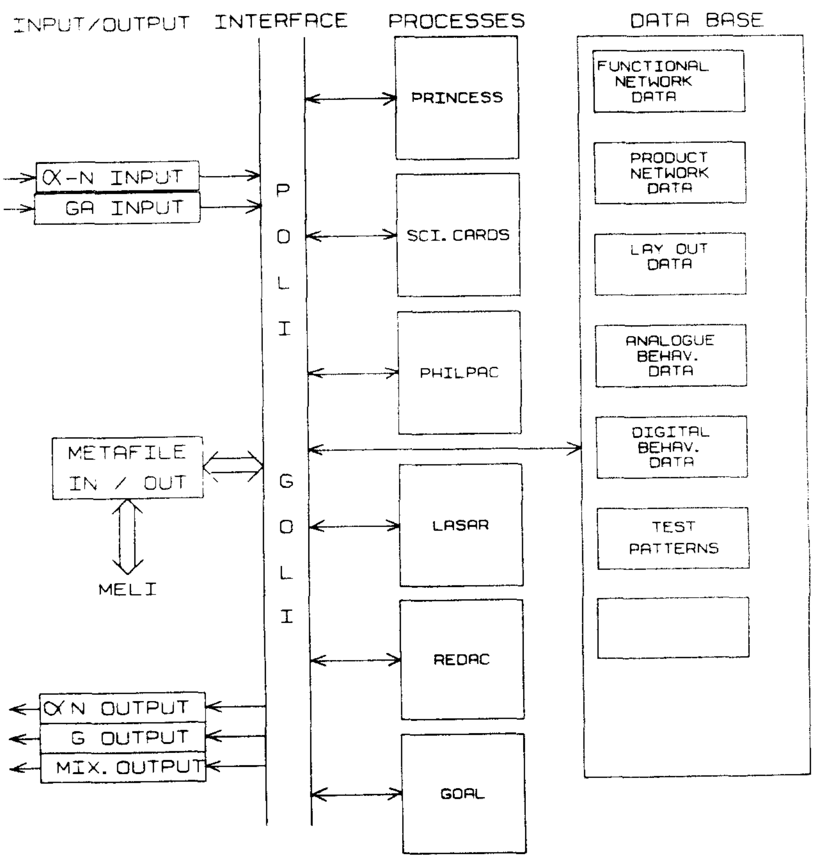

Fig. 4. Full ARCADE System Configuration.

product. This being the case the possibility of generating documents for and about that product (Technical Product Documentation, TPD) using the now available information, has become reality.

What is needed next is a documentation output station that is capable of extracting information from the data base. Powerful text- and graphic-editing programs must be supplied to model or remodel the information, to include illustrations (available in the data base) in the text of the document and to insert references to extensive bibliographies of reports, articles, standards and procedures.

With appropriate document composition and formatting tools, the required documentation can be made. To be more precise, it is anticipated also that texts relating to the product that is being developed will be added to the data base. Such text will have its own "*text modelling technology*" format in the POLI. A work-station consisting of an alphanumeric and a graphic visual display unit will be equipped with local intelligence to edit whatever text or design data is available in the ARCADE data base or add text at the station itself. The same features are to be available for graphics.

The documentalist, having access to the design data and having the ability to edit this data and compose other data can build the files for the documents he or she is preparing. A formatting (or document layout) program can then compose a document for a printing (or typesetting) instrument for the production of a hard copy (e.g. paper, film, etc.) of the document.

The documentalist will have the freedom also to specify in what form the document will appear. This can be in a predefined layout of a book, a manual or the Standard Philips TPD form.

Since the Philips TPD is an essential standard in which the products of the Philips' group of companies are described, the POLI and GOLI interface will be equipped with a "*page set modelling technology*" to support the TPD, the page set structure, and its contents not only in the interface, but also to make it possible to have the TPD composed gradually, piece by piece in the ARCADE Data Base proper.

For the text-editing facilities of the documentation work-stations, use will be made of the locally available text and graphics editors. The graphics editor is one being constructed within the ARCADE project. Finally, the files which contain the text and graphics information have to be composed into a document. Special commands must be provided, for example, to insert files in the documents, insert figures, make provisions for the proper subdivision into chapters, sections, subsections, and to create the required facilities for referencing to other material.

Such a command language is now being defined.

3.1.4. ARCADE Configuration Management

So far, none of the available CAD programs is suitably equipped with administrative functions. And, what is more, the especially important facilities for conducting an appropriate Configuration Management Procedure within CAD are absolutely non-existent. [2]

When all design data is contained in one product data base; when, in addition, much of that data has been generated at different times and has undergone changes and revisions; and further, when the data has been invoked by different CAD programs (within the ARCADE concept), it is not unlikely that certain major difficulties may occur.

These difficulties may comprise the following:

- design data has been lost
- design data reflecting obsolete input data has been used
- the particular design data to be used has not yet been released officially for use by other designers
- the particular design data is stored under different names
- data that has been generated by CAD programs is related to slightly different designs of almost the same article in one department, etc.

To avoid such difficulties ARCADE is equipped with powerful Configuration Management functions and capabilities.

Products may consist of assemblies or sub-assemblies and these may consist of parts. A part can be a PCB, an LSI, a connector, a transducer etc., and all these parts belong to the same (sub-)assembly. It will be recorded in the system which parts belong to what (sub-)assembly and also which (sub-)assemblies belong to what prod-

[2] The author refers to the CAD programs which are commercially available.

uct. In addition, during the design time, modifications in the design of parts, (sub)-assemblies and products will receive new edition numbers, and the manager of the design department may even create new versions of the parts, assemblies, etc. This is done while maintaining the editions and versions of older designs. The design manager may also combine parts into a new part, he or she may delete editions or versions and may authorize certain designers or design departments to have access to the data of the design of certain parts only.

When defining a version of a specific product it is thus possible to combine various versions (or editions) of all the parts into a version (or edition) of a (sub)-assembly and also combine these assemblies into a product.

Full control of the configuration of a system can in this way be achieved. Also the dates of the final creation of the parts or (sub)-assemblies can be recorded, and the name of the persons who are authorized to provide further information.

The Configuration Management Data must be added to the Input Data of SAIL, processed by the SAIL language processor and contained in a "*configuration modelling technology*" format of POLI. A special program (Configuration Handling Input Processor) called CHIP, will keep track of the editions and versions of parts, sub-assemblies and products in the data base. The link programs of the CAD tools, attached to the ARCADE system will have to provide ARCADE with the specific Configuration Data and the CHIP module will then locate the required data and determine whether an authorization exists to pass that information to the CAD program that requested it.

Summarizing, it is impossible to access or store information without the CHIP knowing about it, and consequently it will also be impossible for any program to participate in the ARCADE structure without handling Configuration Information.

3.2. Data Processing Techniques in ARCADE

It has been argued in Section 2 that one of the deficiencies in the existing CAD situation or one factor in the current crisis (if the word is appropriate) is the absence of software engineering principles in the construction of CAD application programs. The various methods that can be employed in order to create a system which can be regarded as an appropriately engineered system have also been elaborated on in Section 2.3.

Only the software engineering tools applied in ARCADE will be mentioned in what follows. Full descriptions of the techniques can be acquired in the literature or other reference material, as indicated in the short overview that is presented now.

3.2.1. Language Processing Techniques

The processor of the SAIL input language was defined and constructed with the aid of POLPAC [20], a generalized input processing system, developed at the ISA department of the N.V. Philips' Gloeilampenfabrieken in Eindhoven. POLPAC accepts a BACKUS-like description of the syntax of the language statements that one wants to use, and converts these into tables. The elements in the tables can be read, and specially constructed little programs interpret the elementary data and subsequently generate the POLI-format.

When one wants to change the syntax of the language it is often sufficient to change the syntax rule and generate a new processor. When the semantic meaning of the statement(s) change(s) as well, the programs supporting the translation process have to be altered. Changes are, however, carried out much more easily than in specifically developed language translators.

3.2.2. Graphic Processing

The highest level of graphic processing takes place in the actual application, the lowest level is that of representing the graphic pictures on a screen or a printer/plotter device.

The user will perform the graphic actions by manipulating graphical objects, which is available in the system through a specific "model" of such an object. The user can steer or control the graphic process by means of control functions, often available as "menu" options on the display; the user can build graphic representations by using elementary graphic symbols (e.g. the representation of a NAND-function, or Philips logos), and can also define the carriers for the graphic information (forms, viewports). ARCADE offers the user a general purpose graphic editor which will enable him or her to easily manipulate and use the functions and items mentioned above. This way of working is achieved with three distinct Graphic Software Modules:

- The representation and manipulation of the ob-

jects is carried out by the General Purpose Graphic Processor (GPGP). The main features supported by the GPGP are: representation of electronic networks, assigning graphic layers [3], issuing the graphic information to defined carriers, making provisions for the page mechanism, etc. The GPGP also translates all input, output, control and graphics information in a standard set of subroutines, as defined by the American CORE Standard [5].
- The collection of subroutines is next handled by DI-3000, a product of Precision Visual Inc., which among other functions transforms all graphic real coordinates into normalized coordinates. DI-3000 is also capable of transforming and manipulating graphic text symbols and it can scale pictures to other sizes, etc. Most importantly it provides for an interface to the device drivers, that will accept DI-3000 output.
- It also creates device oriented information that enables a device to generate the required pictures on the screen of a VDU, on a plotter, on a matrix- or laser printer, etc.

The set of graphic processing techniques enables the engineer to generate additional graphic input- and output-modules and add them to ARCADE via the GOLI-interface.

3.2.3. Data Base Management Techniques

Instead of creating a special Data Base Administrator and Management System whithin the constraints of the project development, RAPPORT (developed and marketed by Logica Ltd. in the U.K.) was acquired. RAPPORT was enhanced to properly cater for the performance limitations of this DBMS. Like many Data Base Management Systems RAPPORT could not handle the search and transfer of "bulk" data in an efficient way. Special routines were developed together with Logica Ltd., and added to it. RAPPORT is being used to store, retrieve and administer the information which is stored in the data base and which is delivered by the POLI interface.

The latter function also demonstrates another importance of POLI. In case RAPPORT has to be replaced by another technique (e.g. a Hardware Data Base System) only the programs that translate POLI into the structure of the RAPPORT DBMS have to be changed. If the CAD programs had been connected directly to the ARCADE Data Base all those links would have had to be replaced.

For information which does not need the structured way of storing data, CADABA – a portable sequential filing system – is used. CADABA was developed at the Research Laboratories of Philips in Eindhoven.

The CHIP module administers that which is stored in RAPPORT, as well as that to be handled with the use of CADABA.

3.2.4. Document Formatting Systems

The document formatting tool which is currently used in ARCADE is the SCRIBE system, developed at Carnegie Mellon University and marketed by Unilogic Ltd. [16]. SCRIBE is currently able to integrate the generation of text and pictures within one document. It is therefore required that the document producing equipment, i.e. a printer or a plotter, can
- print or plot textstrings, and
- plot graphic illustrations.

The text is entered via the available text editors and the illustrations, via a graphic editor. The latter has been developed within ARCADE with the engineering tools mentioned under Section 3.2.2.

3.3 Hardware Considerations

All essential functions of ARCADE have been described in general, but hopefully in sufficient detail for the concept of ARCADE to be understood.

The system is implemented on the VAX 11-780 computer and the software developed is written in Concern Standard Fortran. Almost all the engineering tools (SCRIBE, RAPPORT, POLPAC, CADABA, GPGP, etc.) are either available on IBM equipment or they have also been written in the type of Fortran that can be implemented on IBM equipment rather easily. As such, a high degree of compatibility between DEC and IBM equipment has been achieved.

It must, however, be observed that many so-called CAD tools are not so easily interchangeable. SCI-CARDS is not available on IBM and similarly, the layout programs of Computervision and REDAC (to be attached to ARCADE) are

[3] A graphic layer is a collection of geometric information, which combined with other graphic layers define a drawing – or a certain part of a drawing.

implemented on various non-IBM equipment.

This means that, although it is possible to transfer the management part and the interfaces of ARCADE to IBM, it is less likely that the total operation, including the typical CAD programs, will ever run on equipment other than DEC equipment.

However, where physical connections between DEC hardware, Computervision hardware, and IBM hardware are available it is possible to transfer data via the POLI-GOLI formats from DEC to IBM and *vice versa*. As a consequence, it is important to remark that this specific interface will serve as the main interface between the Philips' CAD systems and possible non-Philips CAD implementations. In addition the POLI-GOLI can accommodate the connection of ARCADE to other, i.e. material management systems, on a variety of computers, provided that the POLI-GOLI is implemented on such hardware.

4. How ARCADE Can Be Used

The ARCADE System provides the various development departments with a powerful and universal CAD development system to which many modules can be added and to which interfaces can be attached that will secure the immediate transfer of design data to other systems.

However, ARCADE is also flexible in the sense that there is no specific recipe for how the system is to be used in a specific real environment. Furthermore, in accordance with what has already been stated, nor should there be such a recipe.

What follows, then, is only given to the reader as an information aid to comprehending one possible way of working with the ARCADE system. It is assumed in the following that the organizational departments are the Commercial Department, the Electrical Development Laboratory, the Engineering (layout and detailing) Department, the Production environment and a Documentation Centre. By "Development or Development Departments", is meant the Electrical Development Laboratory and the Engineering Department. The responsibility for running and administrating ARCADE is assumed to rest within Development.

With those assumptions the following way of working is possible.

4.1. Commercial Department

Although it is not yet foreseen that the Commercial Departments will use ARCADE intensively, their information about the type of products that will have to be developed can ensure that those products are assigned a name that can be used in uniquely defining that product. The product name, as such, will be entered by the data base administrator and all (sub)-assemblies or parts that will be designed in the range of the product will be attached to that name.

Later, when completely defined and accepted specification methods become available, the Commercial Department will have to enter that data into the ARCADE product data files as well.

4.2. Electrical Design Laboratory

The Electrical Laboratory is responsible for the design synthesis and the design verification. Up till now they provided the engineering, grosso modo, with the schematic diagrams and determined the type of components – together with the engineering department – that were to be used in realizing the circuit or system (or its relevant (sub)-assemblies or parts).

With or without the involvement of the commercial department the product that has to be designed will be subdivided into a number of (sub)-assemblies and within the (sub)-assemblies the parts, i.e. the circuits, will be determined. All the (sub)-assemblies and the parts will be given unique names (or code numbers). The names of the parts are related to one or more (sub)-assemblies, and the (sub)-assemblies will be given names that are attached to the unique name of the product. In certain situations it is of course also possible that the (sub)-assembly be related to more than one product. The naming conventions are essential, first, in order to administer all the work in a laboratory or development department and second, in order to assure that the configuration management aspects are properly incorporated in the design activities.

The situation with the ARCADE provides the Electrical Laboratory staff with the opportunity to enter the functional description of the circuit immediately into the ARCADE system. After having done so, they can execute the appropriate simulation runs. An analogue simulation program will

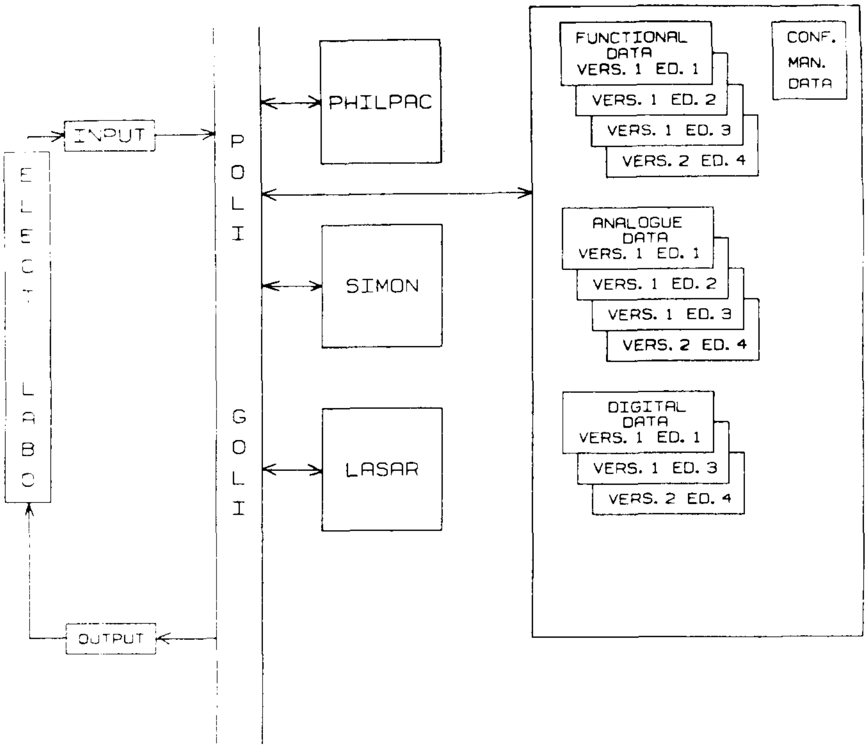

Fig. 5. ARCADE in the Electrical Laboratory.

pass the analogue behavioural data to the ARCADE system and a logic simulation run will do the same with digital behavioural information.

After having evaluated the results of the simulation(s) programs, it is possible for the laboratory to change the design of the part and again perform one or more simulation runs. The changed part has to be identified. It is thus possible to add a new edition of the circuit to be designed and also add the new edition number to the results. The latter is of course necessary to define uniquely what data belongs to exactly what part.

It is also possible to delete older editions of the design, or at some point in time introduce a new version number. At what points in time during the design process such new versions are being issued is a matter of the management of the laboratory.

At the end of the Electrical Laboratory design cycle the data base contains the data as given in Fig. 5. The same figure identifies the input processes that may have been used and the type of programs that were used to generate the behavioural information.

At the end of the work carried out in the laboratory, the Engineering Department will continue, using the data which is now available in ARCADE.

4.3. The Engineering Department

The first action to be performed by the engineering group is to check what the name of the product is, what are the names and code numbers of their (sub)-assemblies and what are the names and/or code numbers of the parts. Let us assume that a PCB has to be made for a certain part. For that part, the functional information is already available in the data base and at the same time it is known what edition/version of the part has to be realized in hardware (e.g. a Printed Circuit Board).

The Engineering Department will first of all add the product data of the ARCADE System (as stated in the previous subsection this is done in cooperation with the Electrical Laboratory). Since all of the network information – both functional and product – is stored in the ARCADE system, it is possible to start the execution of all other relevant CAD processes.

The engineering group will do the layout based

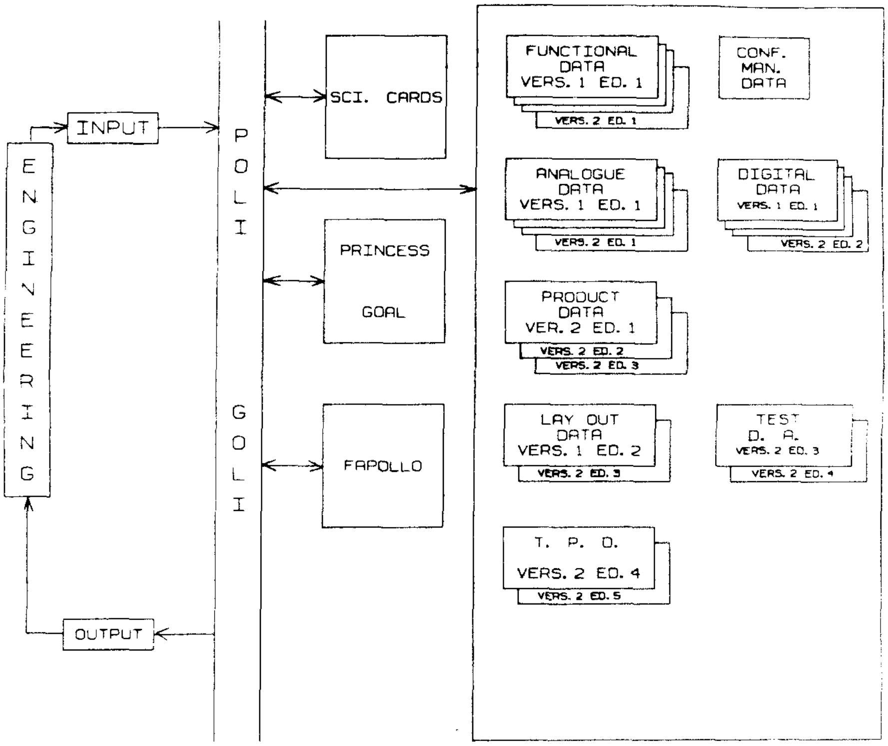

Fig. 6. ARCADE in the Engineering Department.

on the information in ARCADE and can employ a placement program and a routing program to achieve the goal. If it appears that, after having obtained a first result, changes have to be made to the layout, without the necessity of changing the functional description of the part, a new edition of the layout can be generated. If changes have to be made to the functional data as well, then new editions/versions will also have to be issued by the Electrical Laboratory.

The layout information is also stored in ARCADE and if the Engineering Department is responsible for generating test jobs or generating the production data (drilltapes, tapes with information for insertion machines, etc.) then more CAD processes can be initiated by the engineering group which will result in data that will be added, again with the appropriate edition/version numberings, to the ARCADE data base.

In the environment of the author, the Engineering Department is responsible for the creation of the Technical Product Documentation. Again, programs will be available that can create the assembly drawings, the schematic diagrams and other types of documents which belong to the TPD set of documentation.

In each of these cases the Engineering Department can initiate a run that wil yield the required results. The results will always be added to the ARCADE data base, no matter whether or not immediate drawings or documents are obtained in the current process.

Thus, at the end of the engineering phase (Fig. 6 shows the CAD processes used by the Engineering Department) the results of the programs indicated are now available in ARCADE.

4.4. The Production Unit

It is highly feasible that the manufacturing departments will have the possiblity to access the ARCADE information directly. They can be authorized to do so in order to create the necessary factory documentation by using whatever TPD has already been generated in the development phase. A powerful work-station with which such editing can be carried out is foreseen in the ARCADE concept (as previously stated).

In those instances where the production unit will have to generate or regenerate production data (drill information), an immediate access to ARCADE may be useful. However, as indicated already at the beginning of this section, such access is not mandatory. It may well be that in specific organizations the Production Department will have to use the Technical Product Documentation as issued by the engineering group and work from there. Nevertheless, such data will most certainly be available in the form of computer files and the translation by production will be carried out by computers instead of humans. ARCADE will play an important role in establishing the appropriate interface between development and manufacturing.

It is important to note that the Production Entity will never add or modify data in the released versions or editions of the ARCADE data base. It will either retrieve the essential information, or start processes that will operate on ARCADE data. It is not anticipated, in the current philosophy also to add newly generated information to the ARCADE Data Base. However, the ability is not excluded to add specific manufacturing data or specific information on the quality of the design data or actual designs as special product information files to the ARCADE Data Base.

With the further introduction of programmable equipment in the factories special post-processors, will be able to use the ARCADE information with the objective of generating data which can control the particular automated functions. In the future it is likely that this information will be added to the ARCADE Product Files as well.

The involvement of the Production Unit in ARCADE is visualized in Fig. 7.

4.5. The Documentation Department

All relevant information on the parts, (sub)-assemblies and the entire product is at some point in time available in or available through ARCADE. It is more than likely that the Documentation Departments responsible for the installation manuals, the user manuals and commercial books and brochures will have access to this product information. The documentation work-station concept will give the Documentation Departments the possibility to work on the ARCADE information, without changing it in the ARCADE environment proper. However, if it appears to be important for the total product description to also add specific documents to the ARCADE data base then it may be

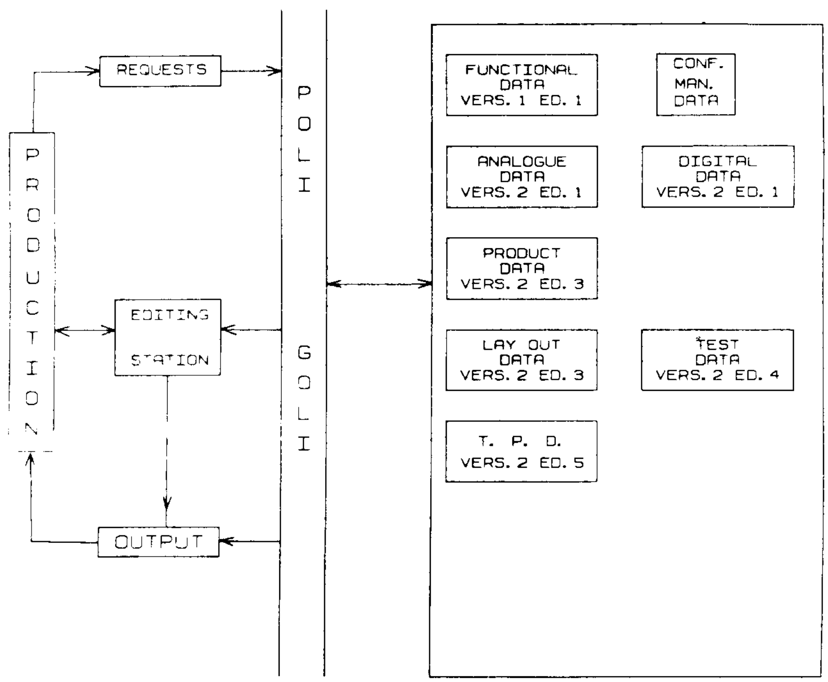

Fig. 7. ARCADE and the Production Unit.

worth considering doing so. It has at that time to be established clearly who will be responsible for the contents of those documents and for the fact that these are indeed residing in the ARCADE Product Files.

4.6. Other Departments and ARCADE

In the last two subsections it was conjectured that the Manufacturing Department and Documentation Groups will have access to the information contained in ARCADE. Of course this also holds for all the other departments that may have an interest in the ARCADE data.

The departments which will certainly exhibit such an interest are the various Logistics Groups and the Administrative Departments – the former since ARCADE will certainly contain data important for the appropriate planning of development and manufacturing, the latter since ARCADE will also contain information about the product itself that can be used to perform certain administrative calculations.

As with production and documentation, the data will be accessed, but never changed by such departments, and in specific cases it may be considered whether the ARCADE product data in the ARCADE data base can be enhanced with data generated by other departments.

5. The Economics of ARCADE

The Computer Aided Design programs which have been used in various development departments have most certainly brought about impressive savings and related benefits. The lead-time of the product under development has decreased especially drastically in those environments where one could eliminate the so-called "breadboarding" and replace this breadboarding by simulation models and techniques. This is also true for the more successful applications of layout systems. The reduction in the number of draftsmen and layout personnel required contributed to yet another range of savings.

What has never really been calculated is the economic benefit that CAD contributed through the drastically increased *quality* of the product, as well as the 100 percent reproducibility.

Savings in the use of expensive materials is yet another aspect that should not be underestimated. Especially in the electronics industry where copper and gold are important ingredients, the savings in such expensive materials through better optimized layouts and a reduction in the number of prototypes that have to be made have become important factors.

Of course there is another side to all this.

- Expensive equipment has had to be installed and maintained.
- Training and retraining of the development engineers and draftspeople has become a necessity today. Retraining becomes even more important with the quickly growing changes in technology itself.
- Organization structures have had to be changed, and working environments adapted.
- The introduction of "software" experts has increased the "loads" on yearly budgets most obviously.

This quick overview of the economic pro's and con's has to be viewed in the context of how CAD is currently introduced and used. In the present situation the advantages are not exploited fully because of the lack of the required integration in the use of CAD in the Development Departments. This in turn manifests itself in a situation that is based partly on automation and that has partly retained conventional methods of working. This has led to difficulties in communication between departments and department employees, but particularly in the manual processing of data and the manual transfer of information from one process to another. Because the system is only partly geared to automatic processes, the introduction of errors because of the lack of the necessary skill and a conventional way of working, has no doubt increased. It often happens that a layout that has been produced does not coincide with the verified circuit that should have been represented in this layout. This also means that various iteration operations are carried out before a product is in line with its specification.

When the improvements that ARCADE can bring about are fully exploited, an additional number of benefits and savings can be realized. It should also be noted that introducing the necessary improvements does not mean that a proportional amount of new equipment has to be purchased. Generally speaking, that equipment is already available and only needs to be adapted or extended.

The following is a list of savings and benefits that can yield a reduction of as much as 15 percent of the total development costs:

1. Saving in the development of in-house, local CAD software, especially intended for input, output, graphic and data base management sections of such software.
2. Savings as a result of greatly simplified training and education of designers. They have to learn only one input method and input language.
3. Savings as a result of the automatic generation of technical product documentation and the simplified modification of this documentation.
4. Savings on searching in data handbooks and libraries, once these have been added to ARCADE in electronic form.
5. Savings in the coupling of CAD programs with one another.
6. Savings in the storage of design results of components and parts and (sub)-assemblies, due to the handling of an electronic in place of a drawings file.
7. Savings due to simplified communication of data between development and factories.
8. Savings in modification operations by the use of adequate initial design data. Those savings may become gigantic after the introduction of appropriate product specification methods [11].
9. Savings due to a greatly reduced maintenance of CAD software.
10. The ARCADE approach leads to the development of a CAD culture which will itself lead to increasing the use of CAD, resulting in an increased productivity per capita.
11. The introduction of a consistent architecture with standard input, output, storage and interface methods has a beneficial effect on communication between designers, design departments, manufacturing, quality control, service, material management, and logistics personnel.
12. Faster adaptations to the changing requirements of CAD methods become possible.
13. Improved standards in the development and introduction of standard data elements will have a positive effect on the various company functions.
14. The shortening of lead-times lowers the interest loss on outstanding development funds.

It must be observed that all of these savings and benefits require a strong desire both for a more disciplined approach in working, and for more advanced standardization practices.

6. Concluding Remarks

CAD has been introduced in this article. The introduction was followed by a number of observations on the actual state-of-the-art in applying CAD. These observations revealed a large number of deficiencies. ARCADE attempts to remedy those deficiencies. This is discussed at length in Section 3. One exemplary way in which ARCADE can be used has been described, as a practical aid to understanding ARCADE concepts. In closing, the possible benefits that ARCADE can bring about were enumerated.

However, a systems approach, and this is also true for ARCADE, will never yield the objectives and anticipated results if the *whole* of the Company is not strongly in support of its introduction.

7. Acknowledgements

The author is indebted to many individuals who, as a team, specified [1] and designed the ARCADE philosophy and its implementation. CAD people from a large variety of Philips development groups have also whole-heartedly participated. Further, discussions which the author has had with his colleagues in IFIP's Technical Committee on Computer Applications in Technology have added insight in the Information Processing aspects of ARCADE.

The many discussions with the CAD group of Bell Laboratories at Whippany, U.S.A., initially triggered the work at Philips.

The availability of Computer Programs such as the DEC VAX 11/780 EDT-editor, the ARCADE Graphic Editor and the Unilogic SCRIBE system made it possible to compose and generate this document in a minimun amount of time.

The author very specifically acknowledges those who were crucial in reviewing and editing the manuscript: Messrs. E. Bolier and G. Marechal and Ms. G. Karlmark.

8. Abbreviations

ARCADE	Architecture of CAD for Electronics
CAD	Computer Aided Design
CAM	Computer Aided Manufacture
CHIP	Configuration Handling Input Processor
DBMS	Data Base Management System
DEC	Digital Equipment Corporation
GKS	Graphic Kernel System
GOLI	Graphic Oriented Logic Interface
GPGP	General Purpose Graphic Processor
IBM	Internation Business Machines
IC	Integrated Circuit
IEC	International Electro-technical Commission
IFIP	International Federation of Information Processing
ISO	International Standardization Organisation
LSI	Large Scale Integration
PCB	Printed Circuit Board
POLI	Problem Oriented Logic Interface
SAIL	Standard Arcade Interface Language
SCI	Scientific Calculations Incorporated
TPD	Technical Product Documentation
VDU	Visual Display Unit
VLSI	Very Large Scale Integration

References

[1] J. Desmet, J.-C. Emond, G. Karlmark, L. Konijn, G. Marechal, J. Matthys and P. Turkenburg, Arcade Systems Specifications – Volumes 1 through 11. System Specification UB12ARC, N.V. Philips' Gloeilampenfabrieken, 1982. ARCADE Prototype Project Development.

[2] G. Bogo et al., CASSANDRE and the Computer Aided Logical Systems Design, in: Information Processing 1971. Proceedings of the IFIP Congress 1971, pp. 26–32 (North-Holland, Amsterdam, August. 1971).

[3] F. Branin, G. Hogsett, R. Lunde and L. Kugel, ECAP II – A New Electronic Circuit Analysis Program, IEEE Journal of Solid State Circuits, ●●● (1971) 146–166.

[4] Double Issue of *Computers in Industry* – In Memory of Steven Anson Coons, Vol. 3, Nrs. 1,2 (1982).

[5] Status Report of the Graphic Standards Planning Committee of ACM/SIGGRAPH, (ACM/SIGGRAPH, 1978).

[6] R. Eckert, G. Enderle, K. Kansky and F.J. Prester, GKS' 79, Proposal of a Standard for a Graphic Kernel System (Technical University Darmstadt, 1979).

[7] Advanced Statistics Analysis Program – ASTAP. General Information Manual (IBM, 1973).

[8] Working Group Graphics Standard. ISO – Draft International Standard, edition 7.2 (Working Group Graphics Standard, 1982).

[9] C.Y. Lee, An Algorithm for Path Connections and its Applications. IRE, ●●● (1961) 346–365.

[10] S.K. Lee and W.S. Reed, Finite Element Mesh Generation Employing Satellite Graphics, in: Proceedings of the 13th Design Automation Conference, pp. 1–6. (IEEE, June, 1976).

[11] Douglas Lewin, Product Specification and Synthesis, in: J. Musgrave (ed.) Computer Aided Design of Digital Electronic Circuits and Systems. pp. 25–40 (North-Holland, Amsterdam, 1979).

[12] K.J. Loosemore and R.C. Anson, GAELIC – an Automatic Solution to Complex Chip Design, in: Electronic Design Automation, pp. 102–105 (Institution of Electrical Engineers. 1981).

[13] D.O. Pederson et al., A Simulation Program with LSI Emphasis, in: Proceedings of the International Symposium on Circuits and Systems, pp. 1–4 (IEEE, 1978).

[14] P.E. Roberts and K.T. Wolski, Development of a Digital Test Generation System, in: J. Musgrave (ed.), Computer-Aided Design of Digital Electronic Circuits and Systems, pp. 183–186 (North-Holland, Amsterdam, 1979).

[15] Douglas T. Ross, Introduction to Software Engineering with the AED-0 Language. Report ESL-R-405. M.I.T. DSR Project No. 71425. Massachusetts Institute of Technology, October, 1969).

[16] B.K. Reid and J.H. Walker, SCRIBE – Introductory User's Manual, 3rd edn. (Unilogic Ltd., 605 Devonshire Street, Pittsburgh, PA 15213, 1980).

[17] I.E. Sutherland, Sketchpad: A Man-Machine Graphical Communication System. Technical Report 296. Lincoln Laboratory (Massachusetts Institute of Technology, January, 1963).

[18] S.A. Szygenda, Simulation of Digital Systems. Where we are and where we may be heading, in: J. Musgrave (ed.) Computer-Aided Design of Digital Electronic Circuits and Systems. pp. 41–56 (North-Holland, Amsterdam 1979).

[19] W.M. van Cleemput, An Hierarchical Language for the Structural Description of Digital Systems. In Proceedings 14th Design Automation Conference, pp. 377–385 (Association for Computing Machinery, 1977).

[20] Peter C. van Twist, POLPAC Users' Guide (Philips' Gloeilampenfabrieken, Corp. ISA, Building SAQ-2, Eindhoven, Netherlands, 1977).

[21] Jakob Vlietstra, An Overview of Computer Aided Printed Circuit Board Design in an Electronic Industry, Computers in Industry, 1(1) (1979) 41–58.

[22] Jakob Vlietstra, The Reality of Computer Aided Design, in: CAD/CAM as a Basis for the Development of Technology in Developing Nations (North-Holland, Amsterdam, 1981).