From Paper Plotters to Interactive Multimedia Systems

M. Bakker, P.J.W. ten Hagen

1. INTRODUCTION

CWI's earliest activity in computer graphics dates back to the 1960s, when it manufactured the X1 plotter and engineered a library of ALGOL 60 plot procedures. Today, CWI is one of the driving forces behind the production of two multimedia applications: the ESPRIT-funded *Multimedia Applications Development Environment* (MADE)[2] and the international computer graphics standard *Presentation Environment of Multimedia Objects* (PREMO)[3].

All that time CWI has continuously contributed to the research, development and promotion of computer graphics technology, on scientific level, organizational level, and last but not least on standardization level. This last item is being highlighted in this chapter of the CWI golden jubilee book, since it covers a highly interesting and even sometimes adventurous episode in the history of the interactive systems department.

2. HISTORY

2.1. The need for a graphics standard

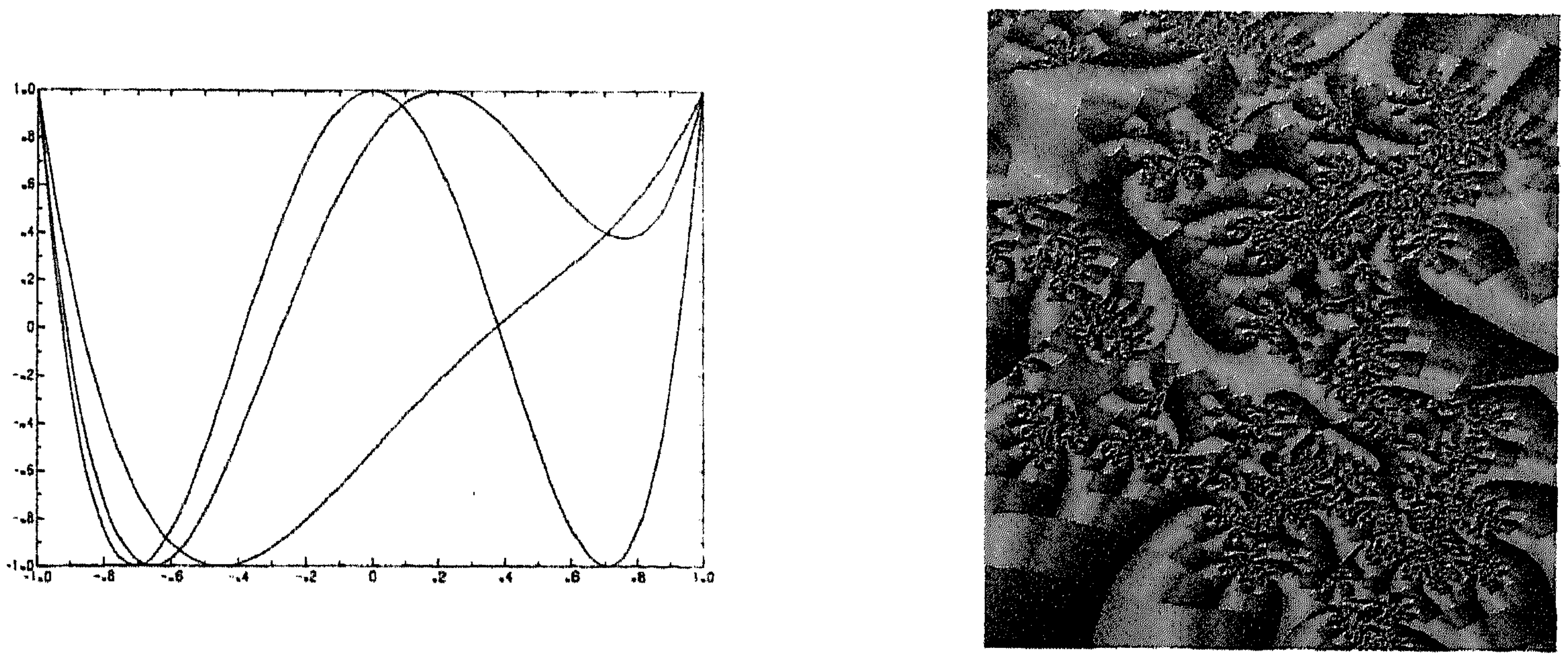
In the 1950s and 1960s, when the computer was in its infancy, computer graphics consisted only of some primitive paper plotters and cathode ray tubes. There was little need for standardization: these plotters interpreted drawing instructions from a paper tape or from a deck of punched cards, which had been generated by some computer program and it was generally accepted that other plotting devices could not interpret these drawing instructions. In that era CWI manufactured its X1 plotter and also wrote an Algol 60 library for this plotter. Numerous CWI publications have been illustrated with figures drawn by the X1 plotter (see also figure 1). That same plotter, by the way, was also used to draw music notes generated by some music program.

In the 1970s, the display devices became cheaper and widely available, and the need for standard plot software increased, especially when Fortran captured the market for simulation software and the porting of Fortran application programs to other configurations became commonplace. It was perceived as problematic that a Fortran program could not run elsewhere for the single reason that the visualization component was totally dependent on the graphics peripherals. There was a *de facto* programming language (Fortran; Algol, C and Pascal were much less wide used) but no *de facto* plotting library: existing libraries like CALCOMP, and GINO only had a modest market share and were hardly compatible with each other. In summary, the virtual monopoly of Fortran in the simulation software market and the absence of a dominating visualization library made standardization of visualization software very pressing. It was these factors that eventually led to the making of GKS (see section 3).

2.2. Participation of CWI

At the time of the first standardization efforts (second half of the 1970s), CWI's computer science department was already for more than ten years doing research in the area of computer graphics; these research efforts resulted in the early 1980s in the graphics language *ILP* (*Intermediate Language for Pictures*), which had an architecture very similar to the architecture of the later graphics standard *PHIGS*. In 1976 CWI joined the working group of experts who had to make *from scratch* the first computer graphics standard. From that moment till today, CWI has been active in the computer graphics standards arena.

CWI introduced an abstract level of functionality which could bring order and structure in the functional diversity typical among graphics programming libraries at the time. This structuring allowed for a consequent separation of geometrical and non-geometrical aspects of picture parts and put the binding between the two under application control. The latter formed the basis for elementary feedback mechanisms for interaction. For instance, a picture element could change colour when pointed at by rebinding it to a new colour. Moreover, these mechanisms would work the same across implementations on different platforms. This facility, that interactive applications could become portable, was hitherto unheard of. It had the additional effect that a new generation of graphics workstations was



(a) Then

(b) Now

(a) A simple graph drawn in 1971 by the X1 plotter. The X1 plotter interpreted drawing instructions on a paper tape generated on the X8 computer by an Algol 60 program using the X1 plot library.

(b) An Escher-like fractal. Courtesy Noel Giffin, Tri-University Meson Facility, California. It is 1024 x 768 pixels in GIF89a format made available in 1993 on the internet. The fractal was coloured using level decomposition methods and is generated from a formula using a simple square root function. It was produced using the fractint formula system and a formula of Giffin.

Figure 1. Computer graphics then and now.

developed by the computer industry which closely followed this functional architecture, although its design was aimed at a software layer rather than the underlying hardware.

Even in today's graphics architectures these structuring principles have been maintained. The field has matured to the extent that now industry standards have taken over the role of the ISO standards; for instance, since 1994 the machine independent layer is based on OpenGL which originates from Silicon Graphics. The knowledge of standards making has been transferred to industry. Even for the methods of consensus building, industry closely follows the procedures pioneered by the graphics R&D community.

3. THE FIRST GRAPHICS STANDARD: GKS

The first steps towards an international standard were set in 1976 at an International Federation for Information Processing (IFIP) workshop in Seillac, France, where some dozens of people from industry and from academia in the US, the UK, Japan, France, Germany, The Netherlands, Norway, etc. convened for the first time to convert their computer graphics expertise into an international graphics library.

Many, many other meetings would follow (see, e.g., [1] for a detailed report), all over the world, initially under responsibility of IFIP, later under

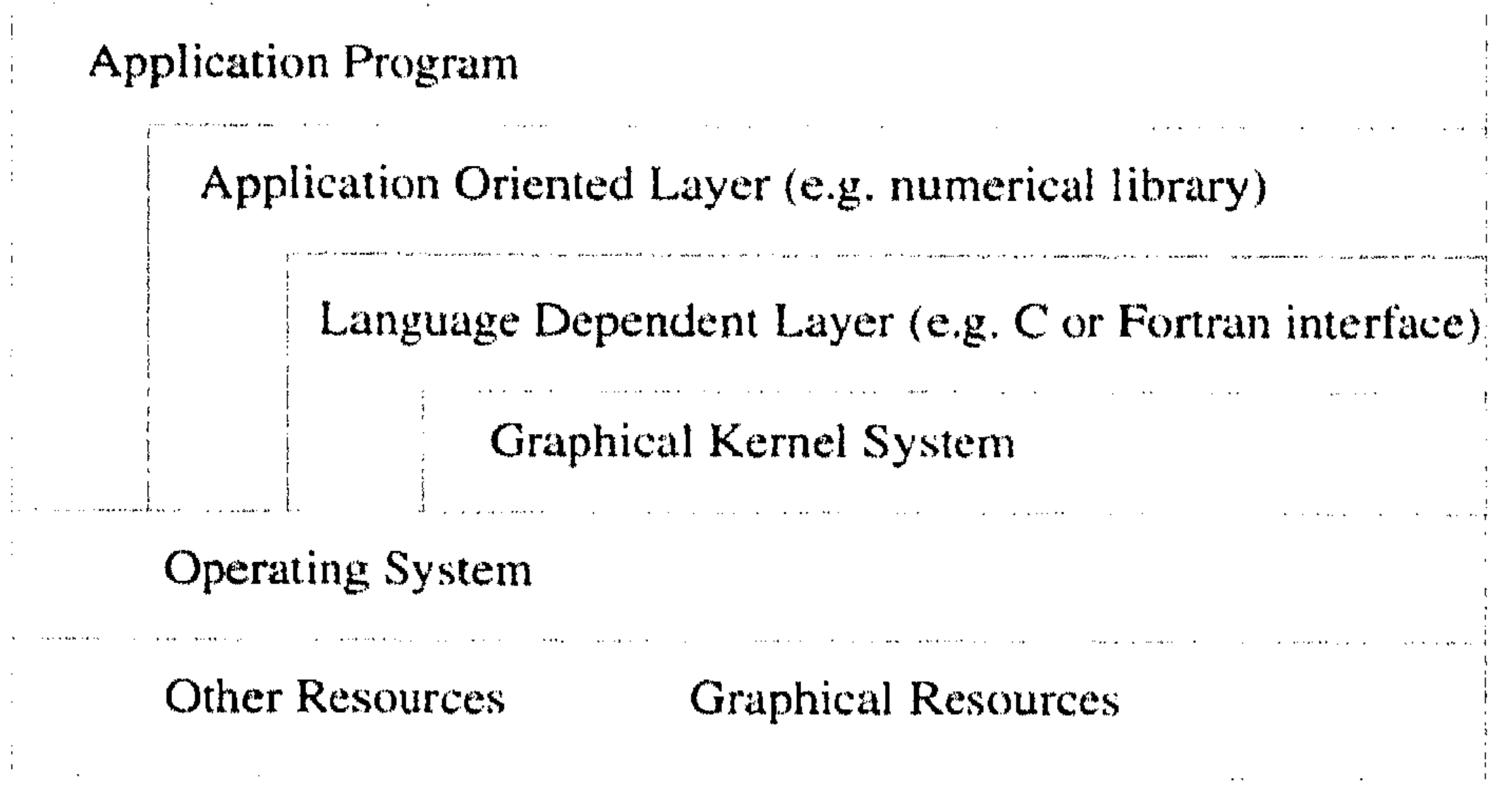


Figure 2. The GKS layer model.

responsibility of the International Organisation of Standards (ISO). In the period 1977-1981 several drafts of candidates for the first graphics standard were produced and revised—amongst them the American standard CORE—but in 1982 the *Graphical Kernel System* (GKS), as it was called then, got stable forms and was registered by ISO as a Draft International Standard, the next-to-last version. In 1985 GKS was published by ISO.

3.1. The strength of GKS

In accordance with the layer model in figure 2, GKS enables application programs to visualize (geometric) data in a *device-independent* way. This strongly facilitates the porting of application programs to other configurations, which was till then problematic. GKS addresses a wide range of graphical *workstations*, including laser printers, photo-typesetters, abstract metafiles, and interactive graphical workstations with window managers like SunView, News, and X-windows.

This flexibility of GKS was realized by, inter alia, the following:

- Separation of geometry (e.g. vertices of a polygon) and attributes (e.g. filling style) in the definition of graphical output.
- Definition of three coordinates systems in the viewing process: a *World Coordinates* systems on application level, a *Device Coordinates* system for the graphics device, and a 'dimensionless' intermediary *Normalized Device Coordinates* system for the composition of the picture.
- Adoption of abstract *device-independent* concepts like *workstation, logical input device, viewport, linewidth scale factor, colour index,* etc.

3.2. The importance of GKS for CWI

From the very beginning CWI has been actively involved in the development of GKS. This involvement included the writing of technical contributions, chairing the GKS working group, attending and hosting meetings, and engineering a pilot implementation (in C) of GKS, thus illustrating the feasibility of the GKS objectives. In a later stage, CWI also edited the ISO C binding of GKS.

When GKS emerged in 1982 as an international standard, there were only few implementations available — one of them was the CWI implementation. At the same time there was a widespread demand for GKS. As a consequence, the computer graphics project group of CWI has been kind of GKS agency for some years: in the period 1982-1987 both a C and a Fortran implementation of GKS were developed, documented and tested by CWI. In 1985 the marketing and vending of GKS was transferred to a professional software house. Today CWI is still receiving revenues from GKS.

3.3. The impact of GKS on computer graphics (standards)

The influence of GKS on computer graphics and its standards has been enormous. For the (European) software industry, GKS provided a *device-independent* interface between application software and graphics devices. This property of GKS made them free in purchasing visualization hardware of their own choice.

In the standards arena, both in industry and ISO, GKS also had much influence. Not only was GKS adopted all over the world as a national standard, but its terminology—workstation, polyline, to take some examples—and its methodology were widely adopted, both in later ISO standards—like PHIGS and Computer Graphics Metafile—and industrial standards—like X-WINDOWS, PostScript, and OpenGL.

4. INTEGRATION OF GRAPHICS AND MULTIMEDIA STANDARDS

4.1. The new generation of multimedia standards

Graphics and text are media which can be generated by computer programs using basic system support. In contrast, sound, moving pictures and video images are usually captured from the outside world and mixed into presentation schemes. Hence the first generation of multimedia applications could only be off-line editing systems comparable to desk top publishing tools.

The second generation of multimedia systems tries to base itself on computer generation of all media, and at the same time then reap the benefit of merging the various media into integrated presentations automatically. In this more generalized multimedia system concept the use of externally captured source material becomes merely a special case.

It has been recognized that the size and complexity of modern information systems require multimedia presentations in order to be able to effectively communicate and at the same time require that these presentations are generated in real-time on demand.

Moreover, this type of interaction and output generation must be provided by powerful services capable of combining information from distributed sources, who provide data but no built-in presentation functionality. This calls for another level of integrated, uniform functionality based on recognized standards. These standards go way beyond agreements about common exchange formats, which were sufficient for the first generation multimedia systems.

The second generation multimedia applications can only be successfully developed if the modern advanced technology is used. The design of the standard functionality assumes that such advanced techniques are available. Examples are distributed object systems, multi-threaded concurrent systems and efficient synchronization support. On top of this extendible and dynamically adaptable object, classes must be supported.

Each of these features can be justified by some application programmer's need. For instance, the enormous variety of low level presentation functions must be reduced by object specialization to a workable subset for a given application, thereby making a better match between conceptual and concrete functionality; adaptive methods must be used to produce object instances which behave sufficiently efficient in a given situation. The major area which is addressed by this standard, is the area of virtual reality.

Virtual reality is not only a new gadget for the entertainment industry, it is the ultimate means to communicate computer-based information making full use of all human cognitive powers.

4.2. The first and second generation of graphics standards

GKS and its sister ISO standards *PHIGS*, *Computer Graphics Metafile*, and *Computer Graphics Interface* are so-called *first generation* computer graphics standards, which were completed in the 1980s. These standards only address text and graphics. In spite of important differences in their functionality, they share a common architectural approach, which has resulted in implementations that are large monolithic libraries of a set of functions with precisely defined semantics. They reflect an approach towards graphical software libraries predominant in the seventies and the eighties. However, these standards have little chance of providing appropriate responses to the rapid changes in today's technology, and in particular, they fail to fit into the software and hardware system architectures prevailing on today's systems.

When the revision process of GKS started in the late 1980s (the second edition was published in 1994), it soon became apparent that a second generation of graphics standards was needed. These new standards should also address more modern technologies which had emerged in the late 1980s, such as

- Programming environments supported by windowing technology and open distributed processing.
- Advanced rendering methods like *ray tracing* and *radiosity.*

- Modern insights in software engineering, such as the use of object-oriented specification methods.
- Modern presentation techniques, in particular multimedia and hypermedia technology using international standards like, e.g., HYTIME and MHEG.

The ISO subcommittee responsible for the development and maintenance of graphics and image processing standards recognized the need to develop a new line of graphics standards, along radically different lines from previous methods. To this end, a new project was started at an ISO meeting at Chiemsee, Germany, in October 1992. Subsequent meetings resulted in a draft for a new standard called PREMO (Presentation Environment for Multimedia Objects)[3]. Publication of the final text is expected in 1997.

4.3. PREMO

General architecture. Underlying all of PREMO is a concise conceptual framework, comprising a description technique, an abstract object model used for the definition of data types and the operations upon them, and the notion of components which contain and organize the PREMO functionality needed to address specific problem areas.

Object Model. In PREMO, a strong emphasis is placed on the ability of objects (e.g., enhanced geometric data) to be active. This feature of PREMO stems from the need for synchronization in multimedia environments. Conceptually, different media (e.g., a video sequence and a corresponding sound track) may be considered as *parallel* activities that have to reach specific milestones at distinct and possibly user-definable synchronization points.

Events, Event Model. The PREMO framework includes the notion of non-objects, primarily for efficiency reasons. Non-objects have no requests defined on them, they cannot take part in subtyping and inheritance hierarchies. *Events* form a special category of PREMO non–object types, and are the basic building block for the PREMO event model. Events and their propagation (described by the event model) play a fundamental role in the synchronization mechanism.

Components. The object model, the event model, the concept of non-objects, etc., give a conceptual framework for all the basic notions in PREMO. *Components* allow for a structuring of PREMO in terms of the services provided.

A component in PREMO is a collection of object types and non-object data types, from which objects and non-objects can be instantiated. A component can offer *services* usable in a distributed environment, or it may

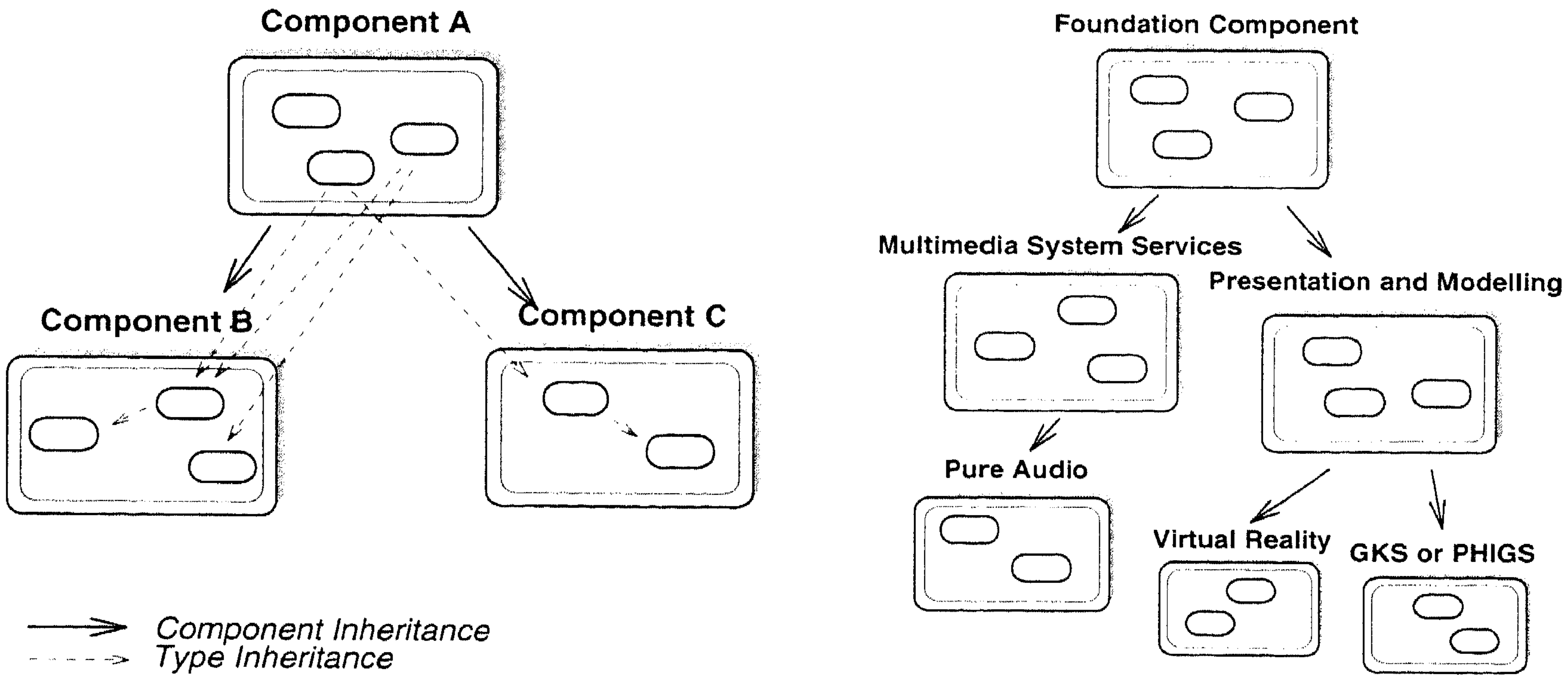


Figure 3. PREMO component model

be used as a set of objects directly linked to an application.

Underlying all PREMO components is a *Foundation Component* providing functionality which is necessary for all PREMO components. It is mandatory that all other PREMO components inherit from this foundation component (see also figure 3). The rules for components form the basis, in conjunction with the object model, for the properties of configuration, customization, extension, and interoperation. PREMO will furthermore include the specification of some other components, namely:

- A component for multimedia system services.
- A modelling, presentation, and interaction component, which will provide the basis of components inherently related to modelling, geometry, traditional computer graphics, etc.

REFERENCES

1. G. ENDERLE, K. KANSY, G. PFAFF (1987). *Computer Graphics Programming—Graphical Kernel System*, Springer-Verlag, Berlin-Heidelberg.
2. I. HERMAN, G. REYNOLDS, J. DAVY (1994). MADE: A multimedia application development environment. *CWI Quarterly 7*, 27-46.
3. I. HERMAN, P. TEN HAGEN, G. REYNOLDS (1994). PREMO: an ISO standard for a presentation environment for multimedia objects. K.R. APT, A. SCHRIJVER, N.M. TEMME (eds.). *From Universal Morphisms to Megabytes: a Baayen Space Odyssey*, CWI, Amsterdam, 347-362.