Introduction to the Special Issue on Computer Graphics

Paul ten Hagen

1. The major research themes of IS

The contributions to this issue of the CWI Quarterly describe the present day research interests of the CWI Department of Interactive Systems (IS). The research topics are selected from the vast area of computers communicating with the outside world, and are accompanied with the construction of experimental systems capable of such communication. Communication by means of pictures has been and still is a paramount topic. It is the leading theme of the computer graphics group of IS. This group has existed for more than ten years now and constituted the kernel around which the department was formed five years ago. In the sequel a brief overview will be given of our involvement in computer graphics. This also illustrates the dual approach to research combining fundamental studies with experimental systems. First we will introduce the main major research themes of IS.

Human-Computer Interaction as a research theme was at first undertaken to improve basic support for interaction in graphics systems, for instance, to provide better input primitives. Next a language model was devised for the rich syntax of pictures and specification methods were designed that could cope with the vague boundary between syntax and semantics. This work has been going on for six years. It produced one of the first User Interface Management Systems, called Dialogue Cells. This was a working but experimental system because, rather than limiting itself within the support capabilities of the operating systems of that time, it assumed full support for asynchronous communication between parallel processes. Moreover, it contained its own window manager since such systems as X-Windows did not yet exist. Although the multiprocessing facilities have greatly improved, several of the limitations concerning concurrent processing are still present in current window managers. It can be shown from the language model that this severely hampers constructive real time input where several independent processes, which are partly user driven, jointly produce one input complex.

About four years ago a new theme was introduced, namely the use of artificial intelligence techniques in computer aided design systems. This is an area where solving fundamental problems would have great practical consequences. To give a few examples:

- There is a great need to evaluate the consequences of design decisions for manufacturing costs. This requires a metalevel analysis to determine which

CWI Quarterly 3, 175-178

parts of design information should be given to the process that governs the manufacturing rule base. Such a metalevel analysis needs its own rule which is in fact the rule based control structure for lower level reasoning.

- Design information is by definition incomplete. Many analysis tasks require complete information. Thus there is a need for an elaborate default mechanism which uses assumptions to produce complete units fit for analysis.
- Computer based design aids must be highly interactive. As a consequence there is a need to produce assertion- and inferencing-mechanisms capable of user communication which is both in real time and understandable. This creates a demand for very direct manipulations, i.e. directly coupled to the semantics of the processes involved. Hence visualisations of the state of the rule base, e.g. the currently assumed constraint values, must be provided.

Research on these topics is relatively recent. A considerable effort was needed to develop a systematic approach to these complex matters.

2. Computer Graphics research and the outside world

In the various papers in this Quarterly you will find descriptions of research activities. Here I will only trace the interactions between our research and the international community of researchers, manufacturers and users in the area of computer graphics.

At the very beginning we focused our attention on the problem of structuring the information describing pictures in such a way that picture changes could be specified easily and realised efficiently. This was when the state of the art was to have a picture description as a long sequence of instructions. This linear list reflected the temporal order in which a sequential program generates such instructions. Such a linear list could contain state information in the form of modal operators which would change the state until a similar modal operator further down the sequence would redefine this action. In these systems a picture change could only be achieved by regenerating the entire sequence with modifications. In the ILP system (Intermediate Language for Pictures) we introduced a tree structured instruction list for the graphics processor, called a structured display file. To generate a picture such a file needed to be traversed. Changes however could be made easily by redefining individual nodes (or leaves) in the tree. The tree structure helped to categorise graphics information in drawing primitives such as lines (fully geometric) and attributes (such as colour) which could be inherited down the tree. The structuring ideas as defined in ILP (1980) were used in both current international graphics standards: being GKS (Graphical Kernel System), developed between 1979 and 1987, and PHIGS (Programmers Hierarchical Interface to Graphics Systems), developed between 1985 and 1990. GKS did not introduce a full hierarchy but accepted a segmentation of the linear list where each segment could inherit global or local attributes either directly or via a reference. During the definition of GKS we proposed several variants, which were specialisations from ILP, until a satisfactory system was achieved. The PHIGS system used a full hierarchy similar to the one provided by ILP. It copied the attributes and primitives of GKS. The idea of PHIGS is to build fast hardware supported traversers of the hierarchical display file. The PHIGS-standard therefore constitutes a fixed high level interface between programmer and firmware. Firmware is usually not very flexible, neither are in general high level interfaces as they tend to be more specific. Already during the development of PHIGS the need for a more versatile functionality became apparent. Hence, now the work is concentrating on PHIGS+, the successor to PHIGS which provides very rich primitives such as NURBS (Non Uniform Rational B-Splines) and extensions for realism in lighting in the form of a rich set of attributes. The great number of functions in PHIGS+ and its variety make it impossible to implement it in hardware. Hence fast general purpose processors have to be added to the lowest level rendering specific hardware. IS has decided not to follow the PHIGS track. The reasons are that the tree structures, if taken too strictly as done in PHIGS and PHIGS+, make it very difficult to exploit parallel hardware, which is a must for high performance systems. Moreover, newer models for realism require networks of objects rather than strict hierarchies. This is often the case because visualizations are directly derived from the complete application models (e.g. a complex chemical compound). Another reason for bypassing PHIGS is to achieve the desired integration of High Performance Graphics with User Interface Management.

In the Interactive Systems' approach the language based interaction structures (e.g. syntax hierarchies), dominate the more local picture hierarchies. Also the inheritance relations need to be dynamically controlled in order to balance pipelined and/or parallel executions. The first attempt to achieve this integration is an effort which is planned with the ERCIM* cooperation. There the various ingredients needed for such integration will be assembled. IS will contribute the Dialogue Cell prototype, further developed in the so-called Manifold system. This supports fully parallel event-driven control where events can originate from both input and output processes. This is essential for sophisticated feedback mechanisms. Last but not least, we will also contribute to the BONSAI system which supports flexible linking between algorithms and computing resources for dynamic construction of efficient graphics pipelines. Figure 1 provides an overview of the various dependencies of these systems from the IS perspective.

(*) ERCIM: the European Research Consortium for Informatics and Mathematics; participants are CWI (Netherlands), GMD (FRG), INRIA (France) and RAL (UK).

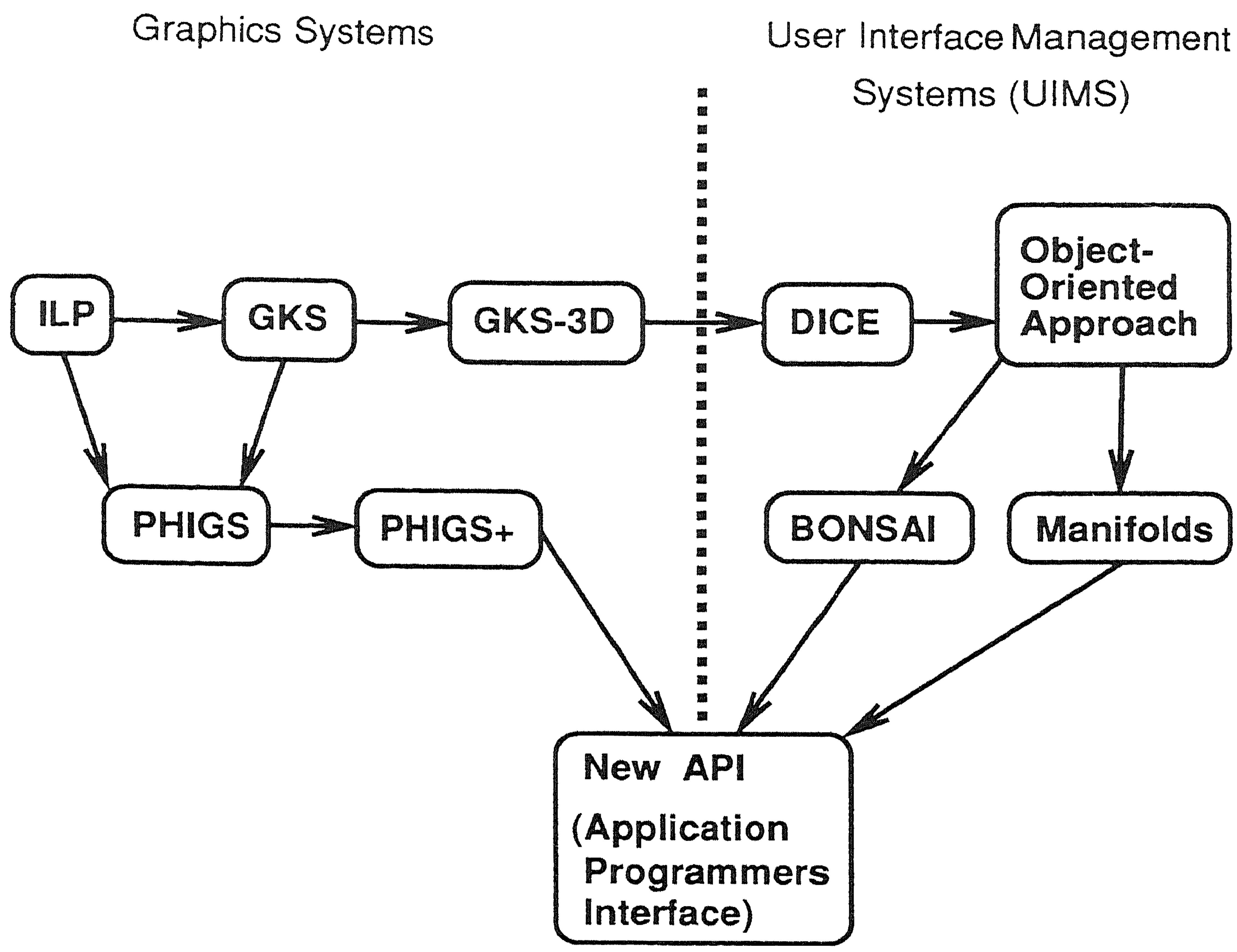

FIGURE 1. Overview of the relations between Graphics Systems and User Interface Management Systems