



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.


C O M M O N S D E E D

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor.



Noncommercial. You may not use this work for commercial purposes.



No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Computer graphics standards for man modelling

K Case, M C Bonney* and J M Porter†

Department of Manufacturing Engineering, Loughborough University of Technology
Loughborough, Leicestershire, UK

*Department of Production Engineering and Production Management, University of Nottingham
Nottingham, UK

† Department of Human Sciences, Loughborough University of Technology.
Loughborough, Leicestershire, UK

The human being is arguably one of the most nonstandard and unpredictable components of all systems. Thus, in many application areas of computer-aided design, there is a need to model the physical aspects of humans alongside models of workplaces and equipment. The paper briefly describes the SAMMIE system, a long-established and successful computer-aided design system that has fulfilled this requirement across a wide range of application areas. Recently, much development work has been aimed at incorporating a range of graphics and data-exchange standards into the SAMMIE software. These experiences are described, together with observations on their apparent limitations and advantages. In particular, the Programmer's Hierarchical Interactive Graphics System (PHIGS), its extension to PHIGS-PLUS, the Computer Graphics Metafile (CGM), the Initial Graphics Exchange Specification (IGES), and X-Windows with the PHIGS extension (PEX) are considered.

graphics standards, PHIGS, metafiles, ergonomics

Ergonomists contribute to the overall design process through their specialist knowledge of human capabilities and how these relate to tasks. This expertise potentially ranges right across the highly varied functionality of humans, including the physical, physiological and psychological aspects. The application areas are similarly wide-ranging, and include simple domestic products such as furniture, and highly complex systems such as the avionics of military aircraft. Human capability is a major parameter of all designs (even highly automated manufacturing systems have to be installed and maintained by humans), and thus ergonomists justifiably consider that their input should be made at the early formative stage of design. However, they are frequently prevented from attaining this objective, and ergonomics is frequently only considered at the evaluative and corrective stage. Where it exists, this problem could be associated with two factors: the relative lack of authority that ergonomists have within their organizations, and a lack of suitable tools for their specialist task. The first

(organizational) factor is possibly closely associated with the second (technical) factor, as ergonomists may effectively be excluded from the design process by the lack of a communication medium. Whereas the design engineers, stress engineers and process planners can all communicate through traditional engineering drawings or computer-aided design (CAD) systems, the ergonomist has traditionally been concerned with evaluating designs using real people together with physical models (mockups or prototypes). This situation is beginning to change, with specialist CAD systems becoming available to ergonomists. The SAMMIE system (System for Aiding Man-Machine Interaction Evaluation) is one such system that is particularly designed to focus on the physical aspects of design for human use. Typically, this applies to situations where the workplace or item of equipment is subject to limitations to the ability of the operator to fit into his or her environment, reach controls and see his working environment. The method is briefly described and illustrated in the first part of this paper, and further details can be found elsewhere^{1,2}.

The second part of the paper is concerned with the impact that computer-graphics and data-exchange standards are having on the development and use of the system. The SAMMIE system is no different from many other CAD systems in this respect, but it provides a useful vehicle for describing experiences gained during a substantial period of development over the last two years. These experiences are viewed from two perspectives: the benefits to the end user of the software, and the benefits to the developers/vendors.

The end users' principal areas of concern relating to standards revolve round the issues of data exchange and graphical output. SAMMIE can be used either as a design-originating system or as a tool to evaluate designs from other systems. In the first case, the SAMMIE system is used to build approximate models to determine some gross design parameters as they affect the human users. Thus, for example, in car design, it may be possible to establish satisfactory head clearances, leg room etc. within the overall design criteria laid down for interior package design (see Figure 1). At the end of this process, it may be adequate to take this relatively small amount of generated information, and use it as the starting point for mechanical design within a more traditional CAD system. However, ultimately this approach is unsatisfactory, as it:

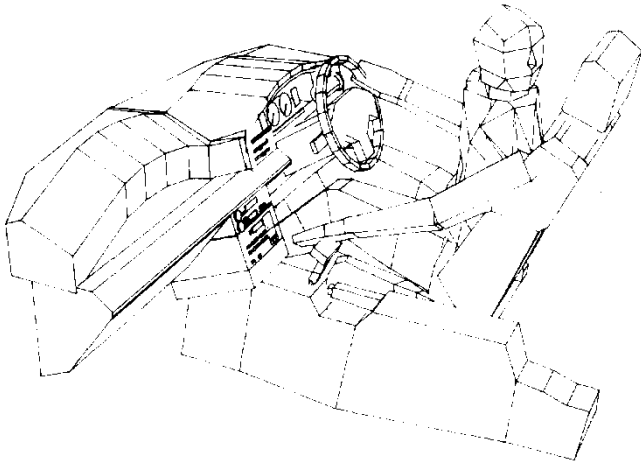


Figure 1. Interior package design for automobiles

- separates the ergonomic and engineering aspects, whereas integration should be looked for,
- is unsuited to the normal iterative procedures of design,
- precludes input from conceptual design systems.

Hence, the need is established for 2-way data exchange, rather than a simple onward transmission from the SAMMIE system to a subsequent CAD system. This requirement is further emphasized when SAMMIE is used as a tool for evaluating designs from more traditional systems.

There are good arguments, of course, for standardizing the modelling rather than the data exchange. In this situation, the SAMMIE system would simply become an application module making use of a standard geometric modeller. Many proprietary CAD systems are constructed on this 'modular' principle, but, for the foreseeable future, genuine standardization in this way remains a research rather than a development issue¹.

Of less fundamental significance, but of considerable practical importance to the end user, is the question of graphical output. This has long been a problem area for users, with a wide variety of screen and hardcopy devices, many requiring their own unique drivers. Standards have provided a very direct help in overcoming these problems, although there is still a considerable way to go.

This last problem is also of major concern to the vendor/developer attempting to maintain his software against a background of constantly expanding device availability. However, he or she also has more significant decisions to make regarding the approach to screen graphics. The development of graphical systems such as CAD systems has traditionally involved a major commitment to the graphics software. The maintenance of this software can be a heavy burden, and can be even more so if attempts are made to support the application across a range of hardware platforms. This effort inevitably detracts from the resources available to tackle the application area (for example man modelling in the case of the SAMMIE system), and thus the use of standards offers some hope in this

direction. There is normally, however, a price to pay for the use of standards, as graphical performance can be significantly affected when they are used. (There is some evidence that hardware manufacturers are beginning genuinely to accept these standards and to start adopting them as the native mode for their machines, which, it is hoped, will alleviate this problem).

SAMMIE SYSTEM

The SAMMIE system is a long-established member of a growing family of man models, brief outlines of which can be found in review papers by Dooley⁴ and Rothwell and Hickey⁵. Detailed descriptions of several of the principal modelling systems can be found in Karwowski, Genaidy and Asfour⁶. Models described include SAMMIE¹ WERNER⁷ (a PC-based system developed in Germany), ergoSPACE⁸ (a Finnish system based on AutoCad), the Dutch TADAPS system⁹, and long-standing US Air Force systems called COMBIMAN and CREW CHIEF¹⁰. *Computer-Aided Design* has recently published some work aimed at the British motor industry¹¹.

All these models are essentially static or kinematic, and should thus be differentiated from the dynamic models, which are used where external forces on the body are considerably higher than those that can be generated internally by the musculoskeletal system. The modelling of car-crash situations¹² is a typical example of the use of dynamic modellers.

Dimensionality is another important classification issue. 2D man models associated with traditional 2D draughting packages do exist, and these go a long way towards mimicking the use of drawing-board templates. However, it has long been the authors' belief that the nature of human-interface problems is so essentially 3D that nothing short of a 3D approach is adequate. Further, humans work in real environments, where interference with surrounding objects is a major consideration, and thus solid modelling becomes a necessity.

Solid modelling

Almost since its inception, the SAMMIE system has used solid-modelling techniques. The relative imprecision of ergonomics evaluation (when compared, say, with the needs of manufacturing engineering) has permitted the use of a simple form of the boundary-representation solid-modelling technique¹³, where only plane-faceted polyhedra are permitted (see Figure 2). It is possible to define primitives, such as cylinders, cones and spheres, where the maximum difference between the intended surface and the facet is under the control of the user, and colour-shading techniques, such as that developed by Gouraud¹⁴ are available to reconstitute the curved surface. The major advantage of using a simple boundary representation is that an explicit representation for the edges of the model is maintained at all times. This means that redrawing the model need not involve the computationally intense edge determination that is implicit in other modelling techniques (such as constructive solid geometry¹³). As one of the

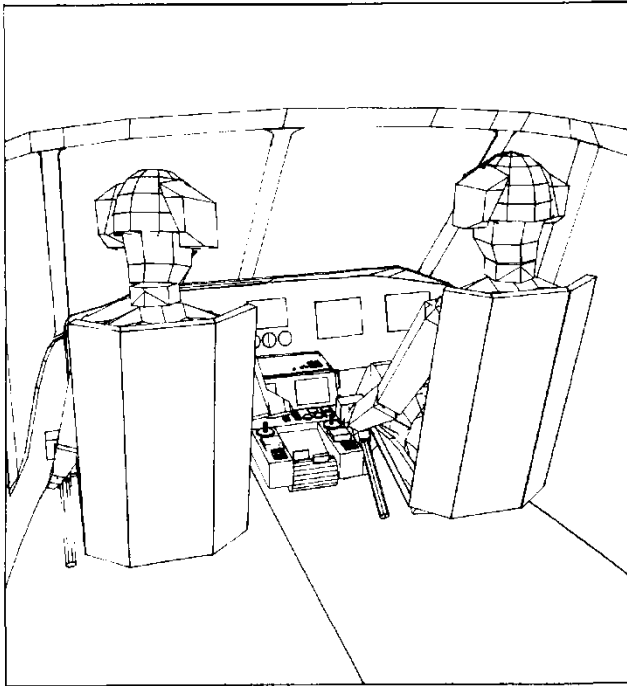


Figure 2. View of a helicopter flight deck with pilots wearing head-up display helmets

Primary tasks of the SAMMIE system is the rapid manipulation of objects in 3D space, it is natural that boundary representation has been chosen as the method of solid modelling.

Data structure

The SAMMIE system uses a hierarchical ring structure to represent the three main modelling aspects:

- geometry and topology,
- spatial relationships,
- logical or functional relationships.

A simple data structure is shown in Figure 3 where, in general, the geometry blocks on the rings are, in fact, ring structures in themselves (although parametrically defined primitives such as cuboids may exist in a compressed form). Thus, within a geometry block, the topology is defined in the classic way, as shown in Figure 4. The implementation uses the well-known 'winged-edge' data structure¹⁵ where there is knowledge of the topological relationships between the faces, edges and vertices of a solid. This information is essential for algorithmic methods, such as interference detection and hidden-line removal, that depend on the 'solidness' of the model.

Spatial relationships are held in the transformation blocks that form the headers of the rings, and they take the form of homogeneous transformations. Traversal of the data structure to generate a particular configuration of the geometry causes these transformations to be concatenated down the structure before application to the geometry contained within geometry blocks owned by a transformation. Modification of a transformation thus affects

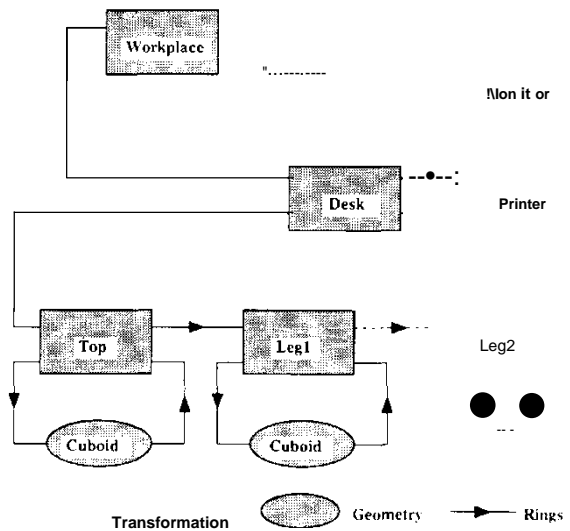


Figure 3. Simplified example of the SAMMIE hierarchical data structure

all geometry further down the hierarchy.

Logical relationships are encapsulated within the structure of transformations (more normally referred to in this context as entities to include the geometry on their rings). The hierarchy is used to directly represent functional structures such as the human being in the 'thigh bone connected to the hip bone' manner. Each entity is logically owned by a higher entity (or by the *workplace* at the top level), and may itself own other entities. User facilities are provided for the manipulation of these ownership relationships as an aid to design and evaluation, and thus, for example, if the man model picks up an item, then this is reflected in a change of ownership.

This kind of data structure is not unusual, but the hierarchical aspects are particularly well developed in the SAMMIE system, as a reflection of the direct relationship between this and the structures being modelled. There is also considerable significance in using this kind of structure

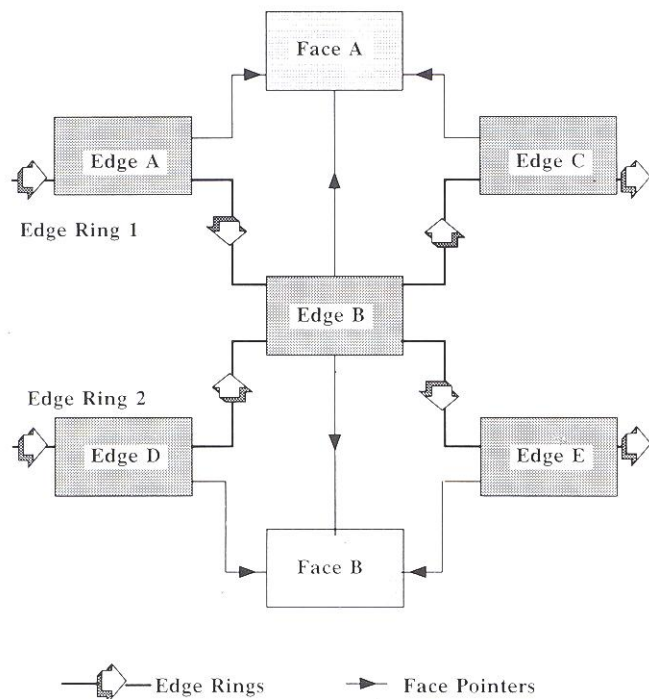


Figure 4. 'Winged-edge' structure of a geometry block [Edge B is on two different edge rings, one for face A and the other for face B. Faces A and B form the 'wings' of edge B.]

in the context of a hierarchically structured graphics methodology such as that used by PHIGS (described below).

Man model

Within the category of 3D kinematic man models, a pin-jointed rigid-link description is invariably used. This approach lends itself to the description of open chain-link systems, of which the human body is a good example. (The body's limbs form sequentially linked 'chains' that terminate at the hands or feet, which are free to move in 3D space, and are thus considered to be 'open'). This model formulation was first proposed by Dempster¹⁶, and has since been used in other application areas, such as the modelling of industrial robots¹⁷ and modular machines¹⁸.

The selection of a representative set of pin joints and rigid links is a major determinant of the performance of the eventual man model. In the SAMMIE system, the modelling includes the ankles, knees, hips, lumbar and thoracic spine, shoulders, elbows, wrists, neck, head and eyes as an approximation to the body joints (see Figure 5). The lack of individual finger joints prevents use of the model for investigating tasks where manual dexterity predominates, and the simplified representation of the spine reduces overall body mobility. Other researchers have considered models of the hand¹⁹ and spine²⁰, but these add a degree of complexity that current posture-prediction algorithms are incapable of handling.

Whatever the chosen degree of complexity, there is a need to provide ways in which the joint and link parameters may be modified to represent human variability. At the fundamental level of the joint and link model, this is

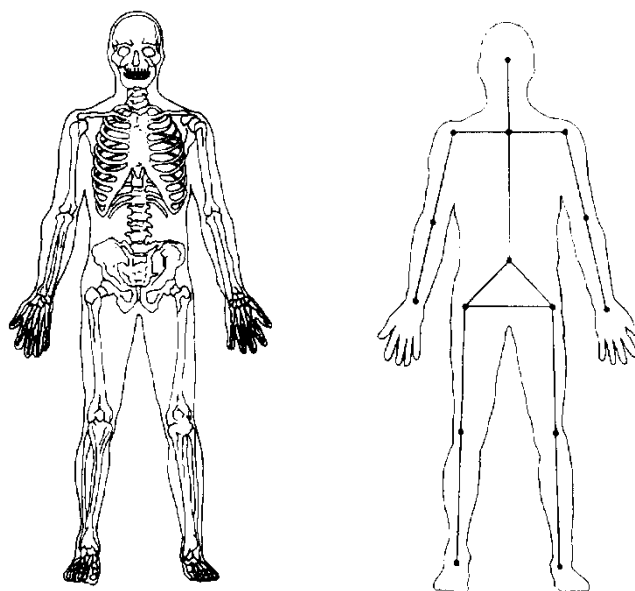


Figure 5. Man model

achieved by the use of data-driven techniques to specify constraints to movement at the joints and the link lengths. Thus, for example, anthropometric data (defining limb lengths) for a variety of human populations is maintained in a simple database as a set of means and standard deviations. The selection of a particular set of data by the user determines the range of man models that can be represented, for example that of British male car drivers²¹, and interactive techniques allow the selection of a particular instance from this population, such as the 95th percentile by stature (the '95th percentile' means in this case that only 5% of the population being considered are taller). Similarly, individual limb dimensions can be controlled by percentile or by the explicit provision of a dimension, and, in this way, it is possible to build a representation of an individual if required. This information is entered into a geometric data structure of the type described above.

The basic 'stick' model is fleshed out by the addition of enveloping solid shapes for each link. The user requires a way of specifying this shape in relation to the particular human population under study, and this needs to be transformed into a complete geometric representation for the solid modeller. The SAMMIE system uses the somatotyping technique devised by Sheldon²², which enables general body shapes to be described on the three 7-point scales of endomorphy, mesomorphy and ectomorphy (fatness, muscularity and thinness). Algorithms are then used to determine the vertex locations within a fixed topological description of each body-segment shape. Combined with the ability to vary linear (joint-to-joint) dimensions, this provides a way of generating a truly infinite variety of human-body forms (see Figure 6).

For the man model to be truly useful, ways of manipulating it relative to the workplace model are required. These are available in abundance, and essentially provide special-purpose spatial transformations that are directly related to the known logical and geometric structure of the man model. The simplest form is a set of joint-extension

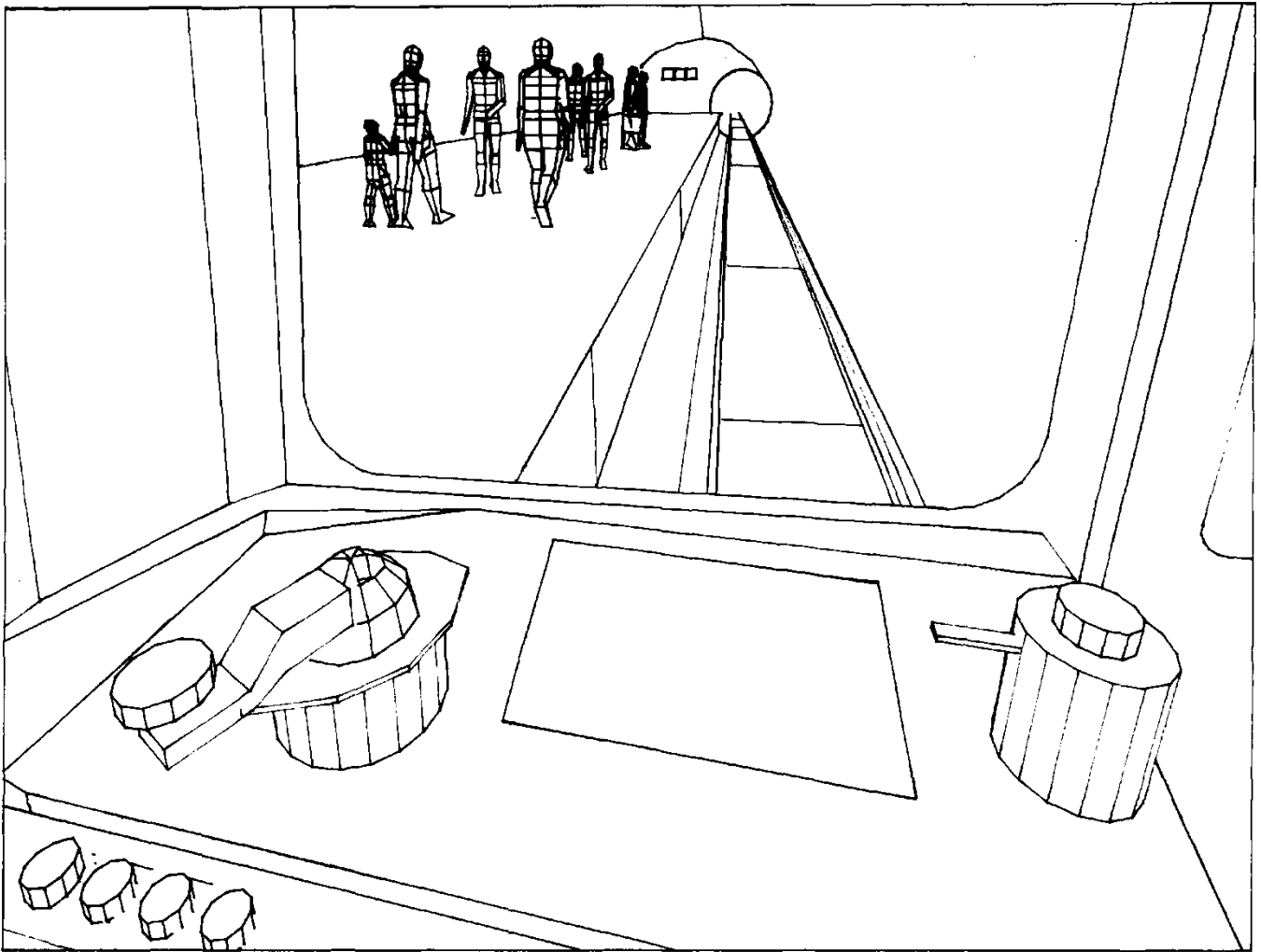


Figure 6. Underground train driver's view of passengers on platform, showing some of the available varieties of human form

angles that describe a body posture, and facilities are available to generate immediately common postures and to create libraries of special-purpose postures (see Figure 7). It is also possible to define an end point in 3D space for one of the chains (a hand or foot), and, from this, determine the location and orientation of the links of the chain by a combination of standard inverse-transformation techniques and special-purpose algorithms. A good description of this technique can be found in Korein²³, and similar techniques for robot modelling can be found in Craig²⁴.

Within the SAMMIE system, the basic technique is extended, principally by making a variety of ways available for specifying the end point. Thus the man or woman can be instructed to reach to coordinates in space, geometric items within the model, or to patterns of points across planes or within volumes. The volumes of reach resulting from the last technique (see, for example, Figure 8) are particularly useful at the early stages of design, when attempts are being made to locate initial controls and displays.

OBJECTIVES OF STANDARDIZATION

The principal objectives of standardization within the SAMMIE system have already been identified as machine independence, graphics independence and data exchange.

Machine independence

Machine independence has been pursued principally within the context of Unix-based workstations, typified by those produced by HP/Apollo, Sun and Silicon Graphics. Most of the SAMMIE system is written in FORTRAN 77, with small elements of C, as these are the only two languages with which there is any hope of achieving portability within the authors' chosen graphics environment (PHIGS)²⁵. Bindings have been standardized for FORTRAN²⁶ and all implementations the authors have encountered provide a FORTRAN interface that complies with this standard. The position with C is less clear, owing to delays in the standardization of C itself. Hence, the C binding for PHIGS is only at the draft stage, and subject to change. All versions of PHIGS that the authors have encountered provide a C interface, but there are very substantial differences in the bindings. Implementations in other languages, such as PASCAL, do exist, but not across a wide range of machines.

Strict adherence to the FORTRAN 77 standard has ensured that 100% portability of code is achieved as far as compilation is concerned, but a small percentage of the code is functionally machine-dependent. For example, the different functional capabilities of the PHIGS implementations that the authors use mean that complete portability is not achievable.

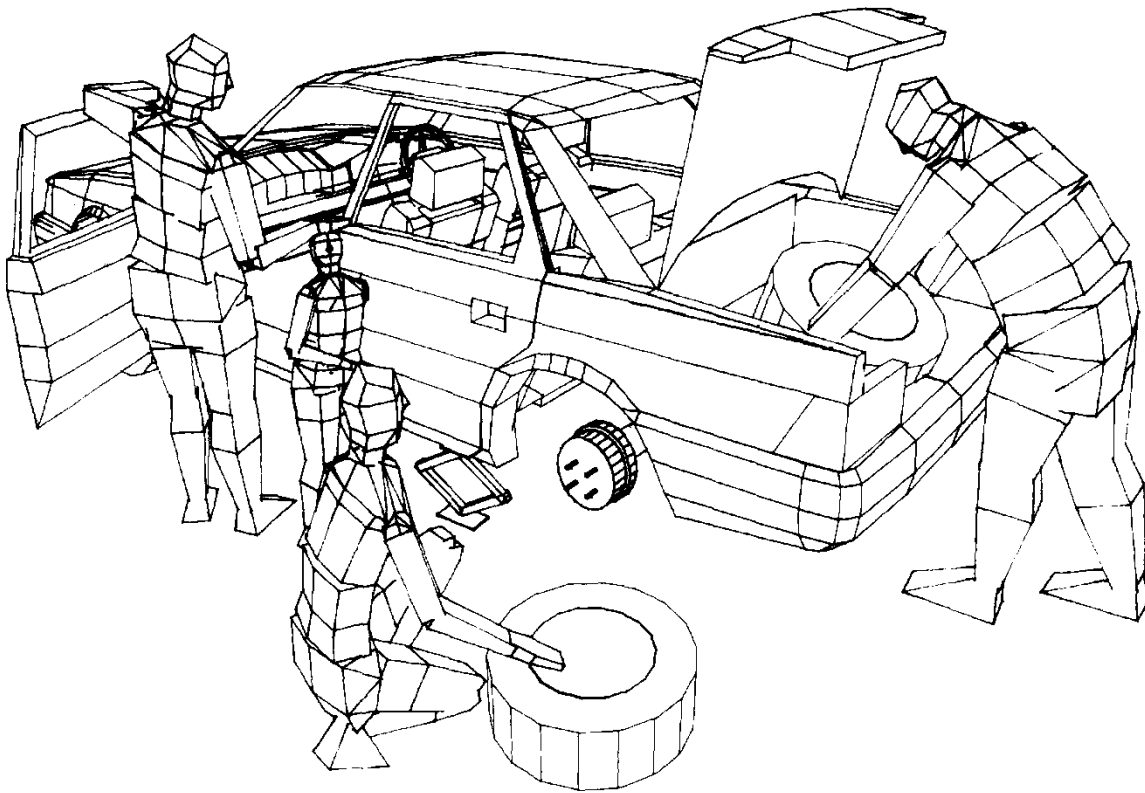


Figure 7. Postures

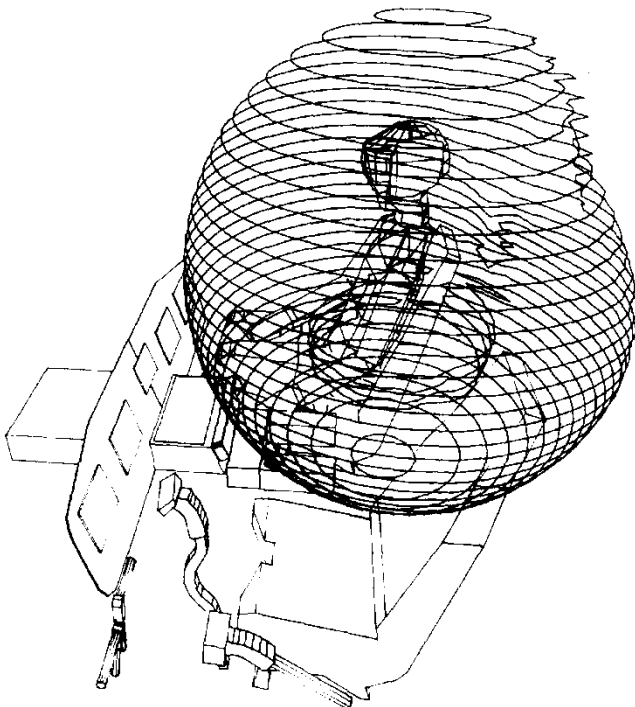


Figure 8. Volumetric contours of reach

Indeed it is this area that is the main barrier to extending the authors' application beyond workstation platforms to non-Unix operating systems and PCs.

Graphics Independence

A measure of graphics independence has been present in the SAMMIE system since its earliest days. Early work in the mid-1970s used the GINO-F graphics package²⁷, a package that, on reflection, had, for its time, some advanced features,

many of which are now incorporated within standards. First attempts at standardization were made with the Graphical Kernel System (GKS)^{28,29} when implementations started to become available from about 1987 onwards. GKS is inherently inadequate for this kind of work, because of its 2D nature, the consequent lack of any viewing capabilities, and deficiencies in picture segmentation. However, the authors felt that the benefits it might bring through graphics standardization across a range of displays and hard-copy devices would adequately compensate for the effort required to add the missing features. Ultimately, the attempt was doomed to failure, because of the very poor implementations (at the time) as far as performance was concerned. GKS defines nine levels to which a particular implementation may conform, so that, for example, at the simplest level, only output to a single workstation can be achieved. However, the standard imposes no performance criteria on achieving functionality. Thus it is perfectly legitimate for GKS completely to redraw a picture each time a change is made to it. This implicit regeneration was widespread for many functions of early implementations of GKS, and precluded its use for highly interactive systems such as the SAMMIE system.

This search for a standard approach to graphics was being carried out at a time when Unix-based workstations such as those produced by Apollo and Sun were becoming the norm for many areas of technical/scientific computing, including computer-aided design. Abandonment of GKS as a candidate for the SAMMIE system's graphics created a difficulty in that, although it was clear that suitable standards (GKS-30 or PHIGS) were on the way, implementations were not available for the authors' targeted machines. Hence,

some intermediate solution was required, and the CORE system³⁰ was selected, largely on the basis of its availability, even though it would clearly very soon be an obsolete approach. The CORE system, like GINO-F, was based principally on the concepts of vector graphics, and, although extended over time into the area of raster graphics, it was clearly a dated approach when the authors started using it in the late 1980s. Despite its limitations, good implementations were available that gave adequate performance for interactive work. Performance would never be optimal for any particular machine, as machine vendors merely paid lip service to standards or portable packages, while quietly maintaining highly proprietary low-level native graphics systems. In the last few years, there seems to have been a change of heart in this respect, and manufacturers are genuinely and enthusiastically offering standard products. Thus, in the area of high-performance graphics, both Hewlett-Packard and Evans & Sutherland have recently announced PHIGS as both their standard offering and their machines' native graphics. The SAMMIE system is now exclusively based on PHIGS and its extension to PHIGS-PLUS. The authors' experiences in this area are described in some detail below.

Data exchange

Data exchange between CAD systems of similar function is reasonably well established in some areas, such as 2D interchange, via the Initial Graphics Exchange Specification (IGES)³¹. The SAMMIE system has been placed in a difficult position by being a solid modeller, an area where standards are only just beginning to make an impact. Simple forms of data exchange have been provided, and these are described below, together with some initial thoughts on how product-data modelling might benefit systems such as the SAMMIE system.

Programmer's Hierarchical Interactive Graphics System

The Programmer's Hierarchical Interactive Graphics System (PHIGS)²¹ has been used for the last 18 months as the sole graphical method within the SAMMIE system. Successful implementations on three major workstations (from Sun, Silicon Graphics and HP/Apollo) have proven the value of the use of standards where more than 95% of the application is completely common to all machines and simply requires compilation. The remaining machine-dependent functions are pre-dominantly, but not exclusively, concerned with initialization functions within a windowed environment, and what could be called the 'bells and whistles' aspects of different, nonstandard fonts, cursors etc.

Acronyms are frequently over-contrived, but, in the case of PHIGS, the name actually conveys an accurate description of the standard. In the first place, it is most definitely a *programmer's* tool, and is not for the novice. It contains some highly complex and sophisticated concepts, which make it impossible to use without spending a considerable time learning its intricacies. To display and control a simple picture using PHIGS is not something that will come easily to the first-time user, who should turn

instead to more appropriate means, such as, possibly, programming using GKS, or using interactive graphic-generation systems (e.g. Freelance and UNIRAS). This statement is not meant as a criticism of PHIGS, which has not compromised its main objective of providing standard graphics for system builders. The comprehensive way in which this has been done has inevitably excluded the less sophisticated users.

The *hierarchical* nature of PHIGS is one of its major advances over its predecessor GKS. A PHIGS-generated display consists of a number of *structures*, each of which can contain descriptions of primitives (lines etc.), attributes (line styles, fonts, colours etc.), spatial transformations (rotations, translations etc.) and control information (enabling or disabling picking, highlighting and visibility). These structures can be arranged hierarchically by including one structure within another via the *execute structure* element. As an execute structure can itself contain execute structures, it is possible to generate a hierarchy of any required depth. Display of a picture then occurs by a structure traversal, where attributes are put into effect, and primitives drawn as they are encountered in the data structure. Similarly, execute structures are part of the traversal mechanism, and will thus inherit attributes from their owners. The spatial transformations can also be concatenated while traversing the structure in a way that is very similar to that used by the SAMMIE data structure described above.

PHIGS has inherited a very rich set of *interactive* features from the earlier work of GKS (and, to a lesser extent, CORE). The input mechanism, using different *logical devices* for locator, stroke, valuator, choice, string and pick functions, each of which can operate in request, sample or event mode, is well described elsewhere³² and will not be elaborated in this paper. The coupling of these to the structure-editing facilities results in a highly interactive system, provided that the particular implementation in use has adequate facilities for dynamic response to these changes.

Clearly, PHIGS is a *graphics system*, in that it goes well beyond the simple provision of a set of subroutines. Implementations incorporate a data structure and data-management techniques, provide convenient ways of relating the graphics work to the windowing system of the host machine, provide for archiving of pictures etc.

Generally speaking, then, the experience with PHIGS, although frustrating at times, has proven to be useful and worthwhile in achieving the authors' aim of graphics independence and system portability between workstations. With the exception of one area, PHIGS and the extension PHIGS-PLUS are providing the authors with all the functionality that they require.

The exception is in the area of integration of the authors' PHIGS application into an X-Window environment³³. The SAMMIE system has long worked on the basis of a user interface consisting of textual screen menus accessed via lightpen on the older vector refresh displays, or by cursor control by mouse on workstations. This approach has been implemented using PHIGS, but with certain detrimental effects on performance. It seems that typical PHIGS implementations handle text very slowly, and this situation is

exacerbated when the more pleasing stroke-written fonts are used in preference to hardware or raster fonts. Quite apart from these performance considerations, there are benefits in other approaches that offer some useful mechanisms for user input (popup menus, forms etc.). These kinds of facilities are readily available in proprietary forms such as Sun's Sunview tool, which can be used to drive a PHIGS graphics window. However, this is in direct conflict with the authors' determination to standardize across workstations, and thus other, standard methods have been sought. X-Windows- based products, such as Motif, Open Dialogue and Open Look, have the required functionality and 'look and feel' qualities, but again these are nonstandard products in an area where it is not yet clear who the winners are going to be. Direct use of X-Windows itself might be thought of as a possibility, however daunting, but, in fact, this is not an acceptable solution at present. X-Windows has many benefits, but capability for high-end graphics is not one of them³⁴, i.e. it is not an alternative to well-developed graphics systems such as PHIGS. Unfortunately, it is not normally possible to use PHIGS in an X-Windows environment, because of the clash between their input models. Once again, the advent of another standard is awaited, in this case PEX (PHIGS Extended X)³⁵, which will support PHIGS and PHIGS- PLUS in X-Windows and X-Windows-based environments.

The authors' implementation of the SAMMIE system in terms of PHIGS raises some interesting questions regarding the relationship between the PHIGS graphical data structure and the SAMMIE geometric solid-modelling data structure. The previous comments on the way in which the SAMMIF system uses a hierarchical data structure to represent the functionality of the human skeletal system in the context of a boundary- representation solid modeller indicates that there is some scope for migrating the SAMMIE data structure into the PHIGS data structure. This is really asking the question as to what extent PHIGS can be considered a modelling as well as a graphical tool. At the present time, the SAMMIE system adopts the inefficient approach of maintaining two linked data structures. All the solid modelling is carried out within the SAMMIE system's own data structure, while viewing and display are controlled by a separate PHIGS structure. Thus a traversal of the SAMMIE data structure for display purposes will cause the building of a structurally similar data structure within PHIGS (a tree rather than ring version of the form shown in Figure 3). Thus the SAMMIE spatial transformations and ownership hierarchy are replicated. The edge descriptions are, at the same time, extracted from the solid model and used to generate line-drawing primitives within the PHIGS structures. The standard PHIGS viewing mechanism is used by the insertion of a view index at the top of the hierarchy, and thus it is modelling coordinates that are described in both data structures. This method ensures that full advantage is taken of any graphics acceleration offered by the hardware. Similarly, the display of objects can be directly manipulated within the PHIGS structure, typically resulting in a very rapid switching on or off of items. The next, and as yet unimplemented, stage is to interact directly with the spatial transformations of the PHIGS

data structure to translate and rotate objects. This would imply PHIGS taking on a modelling rather than graphical role, and, in most circumstances, would result in improved performance. The need for replication of the transformations would be removed, and they could be held and manipulated within PHIGS, leaving the SAMMIE data structure with just a pointer to their location. However, the authors do not believe that it is desirable to attempt the total abandonment of the SAMMIE data structure in favour of the PHIGS one, as no way can be seen of representing the topological relationships required for solid modelling. For example, the 'winged-edge' structure (see Figure 4) is not easily represented within the simple hierarchical tree structures available in PHIGS. The conclusion here is that PHIGS is a suitable modelling tool within its own domain of wireframe, plane surface or NURBS (PHIGS-PLUS) modelling, but not for true solid modelling. The designers of PHIGS will not be surprised by this conclusion, as there was never any intention of extending its capability to solid modelling but unwary potential users may possibly be misled by an apparent ability that it does not in fact have.

The unwary should also be warned about certain performance characteristics of the different implementations of PHIGS. These differences again result from the problems of implicit regeneration that were discussed in respect of GKS. PHIGS provides facilities for the control of how and when the display is updated after a modification to the structure content. The *deferral* mode allows the application to specify that a change should happen as *soon as possible* (ASAP), in which case the entire structure is traversed and redisplayed on every modification to its content, or it can be deferred until the application explicitly requests it (*WAIT* mode). Three *modification modes* are available: *no immediate visual effects* (NIVE), which generally only has relevance in WAIT deferral mode, *update without regeneration* (UWOR), which, for most implementations, has the same effect as the NIVE mode, and *use quick update methods* (UQUM), where partial-refresh methods are used to simulate display changes. The quick-update method of primitive deletion is the imperfect one of 'undrawing' a primitive by redrawing it in the background colour. Similarly, new primitives are drawn with no regard to existing primitives (and thus may incorrectly overwrite them). Not all activities can be simulated by the quick-update methods, and those that cannot require a complete regeneration of the entire display. The lists of functions that can be simulated can be found in the depths of reference manuals, but they are the most important consideration when evaluating the usefulness of PHIGS implementations against a requirement for an interactive application. Given different simulation capabilities, it is necessary to adopt different strategies to cope with frequently occurring circumstances. For example, the simply expressed function of redrawing a structure containing drawing primitives requires different ways of creating the structures and different editing techniques when SunPHIGS³⁶ and Apollo's Domain/PHIGS³⁷ are used. With SunPHIGS, it is necessary to create a new structure, set the attributes, insert the primitives, and close the structure. The structure that is to own the new structure is then opened,

and the new structure added as an execute structure. With Domain/PHIGS, the procedure is similar, but significantly different. The new structure is created and the attributes inserted, but the structure is closed without the addition of the primitives. The new structure is added to the header structure using an execute structure in a way that is identical to the method used by SunPHIGS. Finally, the primitives are added after the reopening of the new structure. The resulting structures are the same, but the method of achieving this was dictated by the simulation characteristics of the two systems (see Figure 9). The same kind of problem emerges with the deletion phase of the redraw function. SunPHIGS can simulate the deletion of a structure or of an execute structure from an owning structure. However, Domain/PHIGS can only simulate the deletion of execute structures, and not structures themselves. The solution is simple, in that, instead of the structure being deleted, its contents are deleted to give the same effect. Provided that the desired effect can be obtained in some way, this may not be seen as too great a problem, but it is in direct conflict with the objective of standardizing the approach and implementation. The conclusion to be drawn from this is that true portability will not be achieved until compliance with some form of performance measure is required in addition to the meeting of the functional specification.

Many other minor irritations will be found in transferring PHIGS programs between different implementations. Only very rudimentary fonts, linetypes, cursors etc. are defined within the standard and provided with a standard binding for their enumerated value. Thus, for example the *set linetype* function is limited to accepting a FORTRAN argument with the enumerated values PLSOLI, PLDASH, PLDOT or PLDASD for solid, broken, dotted and dot-dash lines, respectively, and any other linetypes that are available on a particular implementation have nonstandard bindings.

Similarly, when the echo type for a locator is defined, only a default cursor, crosshair and cross cursor are available as standard. Control of the cursor's colour (which could be crucial for good interface design), and the use of more exotic echoing, such as the use of a rubber-banding line or box, are frequently available, but not defined within the standard. Quirky aspects are present, such as the inexplicable need with Domain/PHIGS to have the locator input device initialized before the pick device can be initialized. This is not explicitly excluded by the standard, but certainly goes outside the spirit of it. Discovering and remedying such problems forms the major part of the work involved in porting software between PHIGS systems.

Graphics metafiles

CAD systems have always required a form of output suitable for producing hard copy. Frequently, this has been a pen plotter driven by the manufacturer's own software (for example the Hewlett-Packard Graphics Language HP-GL³⁸ or Calcomp's Host Computer Basic Software³⁹). Both of these examples are FORTRAN subroutine packages that deal with the relatively low-level requirements of plotters, and generate their own very specific codes for driving the plotter. There is clearly much to be gained by standardizing this process, both at the level of describing the plotter functions, and at the generation of driving code. In essence, this is what a graphics metafile does, by providing a universal description of graphics that is directly acceptable to the plotting device. Plotters then need to have hardware processing capability on-board to interpret the graphics functions into low-level drawing commands. The well-known PostScript has performed this function for several years, but, although popular, it is not a standard. The Computer Graphics Metafile (CGM)⁴⁰ is a standard, and one that is very closely related to GKS and PHIGS. The SAMMIE system has various forms of CGM output, initially based on a general implementation⁴¹ but now using the facilities provided embedded within the various versions of PHIGS that the authors use^{36,37}. This provides a very convenient form of output, as demonstrated by the SAMMIE system's involvement in an experiment⁴² involving the transfer of CGMs between a number of different software products. The artificial scenario in this experiment was that a consortium was bidding for work on a Channel Tunnel contract, and was required to bring together information from diverse sources into one presentation document. A typical piece of graphical output is shown as Figure 10, where the SAMMIE system has contributed some concept-design diagrams, and various other packages have added business graphics, cartography, desktop publishing etc. The final image has been produced on a plotter that has a CGM driver on-board, and can thus directly accept CGMs without any need for processing into a proprietary code. This was achieved on a regular, audience-driven basis, with few problems, except some inconsistencies within packages as to how they described colours.

The above kind of use of metafiles brings immediate benefits in terms of improving the way in which a long-standing activity is carried out. However, the graphics metafile also presents an opportunity for extending the

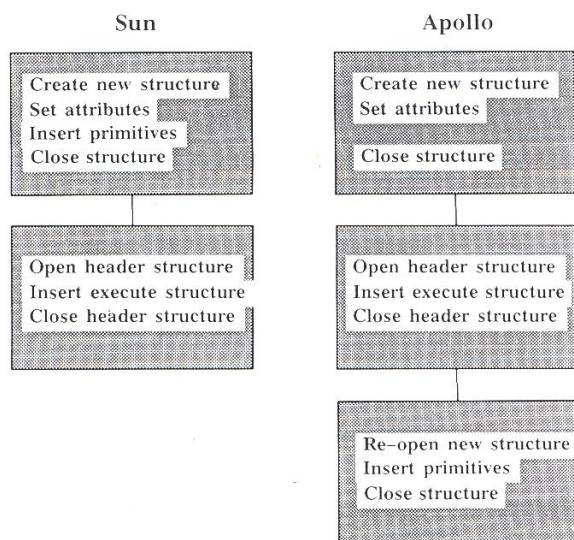
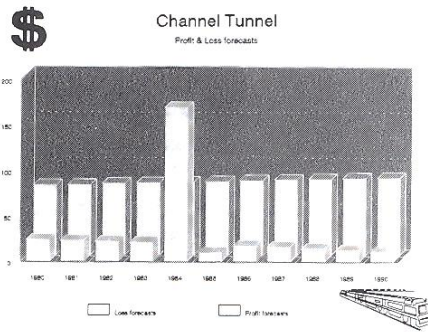


Figure 9. Structure building for SunPHIGS and Domain/PHIGS

CHUNNEL INC REPORT

PROPOSAL FOR NEW CHANNEL LINK

Putting together a proposal for the contract to do the work for the Channel Tunnel is not an easy task. Our scenario involves a new company Chunnel Inc which has been set up to put a proposal together for the Government contract. The key to the success of the project is the way that the consortium of companies which are grouped under this new company can integrate to work on the project. The individual companies all have their own specialities.

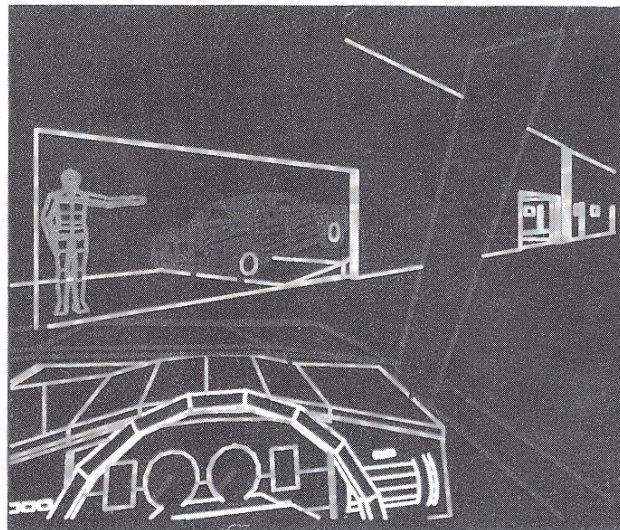


The group involves companies concerned with geological research, engineering work, visualization, marketing and an audio-visual company who handle promotional material.

The companies have inevitably made different decisions about their hardware and software purchases in the past. There was little concern for international standards when the packages were bought but many of the graphics packages have started to include CGM output. There is some hope for integration!

The scenario involves a visit from the representatives of the Government who are investigating the various tenders for the Channel Tunnel work. Chunnel Inc are the only company offering a single tender. This is the strong part of their case. The Managing Director who has put together the consortium is determined to show the companies working together as a well integrated unit. It is vital that the visit goes well.

The MD has decided to gather together representatives from the various forms involved and to set up a mock demonstration of the way that the companies can work together. The aim of the visit is to show that the companies can quickly come up with the answers. The demonstration is set up to show a report being produced which shows some detail of the proposal and some of the costings.



Printed on the Mitsubishi G650 Colour Thermal Printer

Figure 10. Computer Graphics Metafile output

functionality of systems into newer areas. An example of this that is particularly of interest in the context of the SAMMIE system is colour rendering. PHIGS-PLUS offers colour shading/lighting functions that are adequate for the immediate need of model perception during the creation and evaluation stages. However, it would be very beneficial to have more sophisticated facilities available to produce very much more realistic images for presentation purposes. The production of photograph-quality images is a specialized activity, which, at the present time, should be available to, but not part of, a solid modeller. Thus the metafile is one possible way of transmitting data onwards to such packages (the more familiar methods of data

exchange might also be appropriate - see below). In this context, the current form of the Computer Graphics Metafile is inadequate, because of its 20 limitations. A more useful approach may be the PHIGS Archive file, which can be thought of as a 'snapshot' of the Central Structure Store (CSS). Thus the archive file contains structural information together with the necessary 3D information. The file is effectively a graphics metafile, and its form and content is defined by the PHIGS standard. PHIGS archive files can be produced by the SAMMIE system, although, at the present time, the authors have not used these as inputs to advanced visualization software of the type mentioned above. They have been used as a secure format for storing pictures, as

the graphical source for an interactive picture-reviewing facility, and, indeed, as an intermediate form for plot files and CGM files.

Data exchange

Data exchange is most important to the use of the SAMMIE system in the context of exchange of design information between computer-aided design systems. As described earlier, this allows the SAMMIE system to be used as a design-originating system passing on geometric information to more traditional design systems, or it allows the SAMMIE system to import data from other CAD systems, evaluate it from the ergonomics aspect, modify the information, and return it to the originating system. A further use has been identified above, where data exchange may have the objective of transferring information to a system outside the normal area of CAD, such as the visualization packages now becoming available.

The SAMMIE system has several ways of describing the geometry of its model. There is an input language that describes primitives, non-primitive solids, spatial relationships and logical relationships in a clear text syntactic language form. This is a complete model description, and could be used as the starting point for data exchange, but rarely is. As an alternative, a data-definition file containing edge or face descriptions of the solid model may be produced. These are intended to be useful for edge drawing and face-rendering software, respectively. The problem here is, as always, that this is a non-standard approach that is analogous to other vendors' proprietary formats (such as AutoCad's .DXF format), and pre- and postprocessors need to be written to match pairs of sending and receiving systems.

The alternative, standard approach is to use something like the Initial Graphics Exchange Specification (IGES), but this is fraught with difficulties. IGES has an honourable history in the exchange of data between similar CAD systems, principally of the 2D draughting variety, but it is not suited to exchange between fundamentally different systems. Thus it is not possible to contemplate meaningful exchange between, say, a surface modeller and a solid modeller. Hence, in the context of the SAMMIE system, IGES only has potential for communication with other solid modellers. Even this is not currently possible, as the most recent version of IGES (4.0) is only capable of handling solid models of the CSG variety. Version 5.0 has been claimed to be imminent for several years. When it is finally available (at the end of 1990?), it will be capable of handling boundary-representation models, and will thus be of use to the SAMMIE system. In the meantime, the SAMMIE system can generate edge-type files complying with IGES 4.0 (via the data-definition file described above). This is useful, and it has, for example, been used to pass orthographic projections to a draughting package for dimensioning and annotation. However, this could not be described as model-data exchange. The availability of IGES 5.0 should allow the transfer of all the geometric aspects of a SAMMIE model, but, although this would dramatically improve its usefulness, it would still leave the non-geometric aspects unaccounted for. As a simple example, it would be possible to describe the geometry of the man model, but the functionality as a man

model could not be included.

The inclusion of functionality of a model is really in the domain of product-data models, a field that is in a considerably less developed state than that of solid-model transfer.

Currently, the STEP /PDES activities⁴² offer the best hope for the future, but it appears that progress towards establishing a standard in this area is very slow.

CONCLUSIONS

Considerable development effort has been put into the implementation of standards within the SAMMIE system, and there is still much to be done. The overall conclusion is that this effort has been worthwhile in meeting the authors' objectives of reducing the effort required to support and maintain the graphical and data-exchange aspects of the system. Effort can now be concentrated on the authors' particular application area of man modelling and ergonomics evaluation. The authors further believe that, by tracking standards such as PHIGS as they develop, it will be possible very quickly to assimilate new facilities as they appear (such as the lighting facilities offered by PHIGS-PLUS).

This work has not been achieved without considerable pain and frustration, as, inevitably, the standards themselves, and the implementations that the authors use, do not perfectly match the authors' requirements in functionality and performance. There is perhaps a lesson here that the development of standards is a slow business, good implementations take a considerable time to emerge and mature, and it is very easy to try to take decisions too soon.

ACKNOWLEDGEMENTS

The authors acknowledge the original research funding of the SAMMIE system by the then UK Science Research Council (now the UK Science and Engineering Research Council), and continuing support by the University of Nottingham, UK and Loughborough University of Technology, UK.

The rights to the SAMMIE system are owned by the British Technology Group, who have licensed SAMMIE CAD Limited to develop and exploit the system. Most of the standards development described in this paper has been carried out by SAMMIE CAD Limited under the supervision of the authors, who are directors of the company.

REFERENCES

- 1 **Case, K, Porter, J M and Bonney, M C** 'SAMMIE: a man and workplace modelling system' *in* Reference 6, pp 31-56
- 2 **Porter, J M, Case, K and Bonney, M C** 'Computer workspace modelling' *in* **Wilson, J R and Corlett, E N (Eds.)** *Evaluation of Human Work* Taylor & Francis (1990) pp 472-499
- 3 **Pratt, M J and Wilson, P R** 'Requirements for support of form features in a solid modelling system' *Proc. CAM-I* (1985)

- 4 **Dooley, M** 'Anthropometric modeling programs a survey' *IEEE Comput. Graph. & Applic.* Vol 2 No 9 (1982) pp 17 25
- 5 **Rothwell, P L and Hickey, D T** 'Three-dimensional computer models of man' *Proc. Human Factors Soc. 30th Ann. Meet.* Dayton, Ohio, USA (1986) pp 216 220
- 6 **Karwowski, W, Genaidy, A M and Asfour, S S** *Computer Aided Ergonomics* Taylor & Francis (1990)
- 7 **Kloke, W B** 'WERNER: a personal computer implementation of an extensible anthropometric workplace design tool' *in Reference 6*, pp 57 67
- 8 **Launis, M and Lhetela, J** 'Man models in the ergonomic design of workplaces with the micro-computer' *in Reference 6*, pp 68 80
- 9 **Westerink, J, Tragter, A, van der Star, A and Rookmaaker, D P** 'TADAPS: a three-dimensional CAD man model' *in Reference 6*, pp 90 103
- 10 **McDaniel, J W** 'Models for ergonomic analysis and design: COMBIMAN and CREW CHIEF' *in Reference 6*, pp 138 156
- 11 **Okey, R E, Suffell, C and Blount, G N** 'Initial work on a system-independent computer model of a 3D anthropomorphic dummy' *Comput.-Aided Des.* Vol 21 No 6 (1989) pp 393 403
- 12 **Huston, R L, Hessel, R and Passerello, C** 'A three-dimensional vehicle man model for collision and high acceleration studies' *Proc. Automotive Engineering Cong.* Detroit, USA (1974)
- 13 **Requicha, A A G** 'Representations for rigid solids: theory, methods and systems' *ACM Comput. Surveys* Vol 12 No 4 (1980) pp 437 464
- 14 **Gouraud, H** 'Continuous shading of curved surfaces' *IEEE Trans. Comput.* (1971) pp 623 629
- 15 **Baumgart, B** 'Geometric modelling for computer vision' *PhD Thesis* Stanford University, USA (1974)
- 16 **Dempster, W T** 'Space requirements for the seated operator' *WADC Technical Report 55 159* Wright-Patterson Air Force Base, Ohio, USA (1955)
- 17 **Chan, S F, Weston, R H and Case, K** 'Robot simulation and off-line programming' *Comput.-Aided Eng.* (1988) pp 157 162
- 18 **Doyle, R and Case, K** 'The logical and geometric modelling of a universal machine control architecture', submitted to *Eurotech 91* (Jul 1991)
- 19 **Davidoff, N and Freivalds, A** 'Computer-aided design modelling of the human hand' *in Lovesay, E J (Ed.) Contemporary Ergonomics 1990* Taylor & Francis (1990) pp 181 203
- 20 **Freidlich, T** *Unpublished Research Report* Dep. Production Engineering and Production Management, University of Nottingham, UK (1987)
- 21 **Haslegrave, C M** 'Anthropometric profile of the British car driver' *Ergonomics* Vol 23 No 5 (1980) pp 437 468
- 22 **Sheldon, W H** *The Varieties of Human Physique* Harper & Bros., USA (1940)
- 23 **Korein, J M** *A Geometric Investigation of Reach* MIT Press, USA (1984)
- 24 **Craig, J J** *Introduction to Robotics Mechanisms and Control* Addison Wesley, USA (1986)
- 25 *Information Processing Systems - Computer Graphics Programmer's Hierarchical Interactive Graphics System (PHIGS)* *ISO/DIS 9592: (1987)* International Standards Organisation (1987)
- 26 *Information Systems Computer Graphics - Programmer's Hierarchical Interactive Computer Graphics System (PHIGS) FORTRAN Binding* *American National Standard X3.144,1 198x: (1987)* American National Standards Institute (1987)
- 27 'GINO-F' CADCentre, Cambridge, UK (1983)
- 28 **Hopgood, F R A, Duce, D A, Gallop, J and Sutcliffe, D C** *Introduction to the Graphical Kernel System (GKS)* (2nd Ed.) Academic Press (1986)
- 29 'Graphical Kernel System (GKS)' *I50 7942: (1985)* International Standards Organisation (1985)
- 30 **Bergeron, R D, Bono, P and Foley, J D** 'Graphics programming using the CORE system' *ACM Comput. Surveys* Vol 10 (1978) pp 389 443
- 31 **Smith, B, Rinaudot, G R and Wright, T** 'Initial Graphics Exchange Specification (IGES) Version 4.0' US Dep. Commerce, USA (1988)
- 32 **Brown, M D and Heck, M** 'Understanding PHIGS' Template Corp., San Diego, CA, USA (1985)
- 33 **Jones, O** *Introduction to the X-Window System* Prentice Hall (1989)
- 34 **Hess, M** 'Exercising 3-D graphics in a distributed environment' *SunTech J.* Vol 3 No 1 (1990) pp 71 77
- 35 **Rost, R J, Freidberg, J D and Nishimoto, P** 'PEX: a network-transparent 3D graphics system' *IEEE Comput. Graph. & Applic.* Vol 9 (1989) pp 14 26
- 36 *SunPHIGS 1.2 Reference Manual* Sun Microsystems, USA (1990)
- 37 *Programming with Domain/PHIGS* Apollo Computer, USA (1988)
- 38 *Hewlett Packard Industry Standard Plotting Package User's Manual* Hewlett Packard (1987)
- 39 *Calcomp Electromechanical Plotters, Programming* Calcomp Group (1984)
- 40 **Mumford, A and Skall, M (Eds.)** *CGM in the Real World* Springer, Germany (1988)
- 41 **Mumford, A and Liddell, A** 'LUT CGM toolkit guide' Loughborough University of Technology, UK (1989)
- 42 **Mumford, A** *Proc. Eurographics Conf.* (1989)
- 43 **Owen, J and Bloor, M S** 'Neutral formats for product data exchange: the current situation' *Comput.-Aided Des.* Vol 19 No 8 (1987) pp 436 443