

INTEGRATED PROJECT SUPPORT ENVIRONMENTS (IPSE)

INTRODUCTION

By the late 1970's, with the ever increasing demands for larger and more complex systems, it was becoming clear that the computer industry needed more help with software development than was at that time available. This situation was first formally recognised during the Ada development program in the USA. This resulted in a succession of Ada Programming Support Environment or APSE recommendations resulting in the arrival of the "Stoneman" document in 1980 and the STARS (Software Technology for Adaptable, Reliable Systems) program in 1981. During the early 1980's the Alvey committee came to similar conclusions and detailed the need for an Integrated Project Support Environment or IPSE. The latter represents a multi-language support environment such that the APSE is effectively a subset of the IPSE.

This session will concern itself with what progress has been made with IPSE's to date and in particular, it will consider the likely affect of their arrival on system reliability and maintenance.

WHAT EXACTLY IS AN IPSE?

During the course of the last year or two the term IPSE or Integrated Project Support Environment has come into ever more common use in the computing community, but what does it mean? Well a half dozen different computer users would probably come up with a half dozen different definitions as indeed would different software suppliers. For the sake of clarity, during this session I will define an IPSE as follows:-

A closely integrated set of software tools mounted on top of a database and interacting with a variety of users via a common interface.

So how does this differ from a conventional set of project support tools set up on a standard operating system such as UNIX or VMS?

A complete IPSE which has been "tune in" to a particular project development environment will exhibit four further properties:-

- * it will incorporate a complete set of tools to support the whole development cycle, project management, configuration management and quality assurance;
- * these tools will be closely integrated to allow maximum communications and transfer of information between them. They will fit together like the pieces of a jigsaw, so they can be used in close harmony with one another; thus enabling the user to mould the project into a logical whole rather than a set of disjointed tasks;
- * all project details are collected together in a single database offering good cross referencing facilities. This in itself will provide enormous benefits

to configuration management;

- * there will be project wide control mechanisms providing access to both the tools and the database. The intention here is to provide ease of use with regard to the complete tool-set.

Alvey proposed three generations of IPSE's offering increasing degrees of integration between the tools. A 1st generation IPSE is represented by a collection of tools organised around a file base and intended to span all phases and aspects of system development. Such IPSEs would probably be UNIX based, but offering a rather more easily used user interface than UNIX traditionally offers.

A 2nd generation IPSE has more of a true database structure rather than the simple filebase exhibited in the 1st generation. It will also be multi-language, multi-method and distributed. 2nd generation IPSEs, integrate tools at a much lower level than simply holding them in a database as isolated components. Consequently many of the tools incorporated will be purpose built. Although no 3rd generation IPSEs have yet been developed or are even on the drawing board, it is envisaged that they will use a knowledge based system. The intention is to allow the user to move freely between tasks and their related tools rather than invoking each tool serially on completion of its predecessor.

Whilst the Alvey Committee have proposed three distinct generations of IPSE there has and will be incremental development through the generations rather than the productions of distinguishable products.

Another important feature as yet unmentioned is the ability to incorporate existing tools or to develop new tools within the IPSE framework. Naturally enough the more closely integrated the tool-set the more difficult it is to introduce "alien" tools such that they can communicate at a sufficiently low level with the existing tool-set.

Consequently this "open endedness" represents an important extra feature particularly in a second generation IPSE.

WHAT IS CURRENTLY AVAILABLE?

The term IPSE has been adopted by the software supply industry and has been applied to a wide variety of support environments which don't always fully match the requirements laid down by Alvey. The products currently available can be conveniently split into two distinct groups: those aimed principally at the commercial DP market and those which have been developed for use throughout the computing industry. This session will look briefly at a small selection of IPSEs in each area. No attempt will be made to look at any of these products comprehensively nor is this selection exhaustive.

Commercially oriented	(BIS/IPSE (Maestro
General purpose	(ISTAR (GENOS

Commercially Orientated IPSEs

Programmers workbenches are primarily aimed at assisting and controlling the coding and in-use support of on-line data processing systems and have been available for a number of years now. Recently attempts have been made to extend the facilities offered by these programmer support environments with the emergence of such products as BIS/IPSE, Maestro and others.

BIS/IPSE

At the heart of the BIS/IPSE is a proprietary relational database which sits on top of UNIX and hides it behind its own user interface.

It is a "closed" environment, that is to say it does not provide facilities for incorporating alien tools. However it does offer a range of support facilities for:

- project management
- documentation management
- development support

and in so doing implicitly recognises the roles of the:

project manager	project controller
librarian	system modeller
project worker	typist
system administrator	standards administrator

There are a selection of project management facilities including standard task lists, to assist with task identification, together with scheduling and resource allocation algorithms and full graphical support.

Documentation management represents the primary method for control of project development. It is in effect BIS/IPSEs configuration control mechanism. Throughout development the author has exclusive change access rights. When the software unit has been formally approved, it enters the project library where it can only be changed with specific authorisation.

BIS/IPSE provides development support for MODUS and SSADM as well as offering a free format diagram editor. The editor enables the user to decide upon whatever diagrammatic notation appears to be most appropriate but it must be remembered that any associated rules will not be recognised.

At time of print, the BIS/IPSE costs between #15,000 and #100,000 including hardware depending on the desired configuration.

Maestro

Maestro was developed a few years ago by Philips Electronics and originally appeared as a programmer's workbench. However, it has recently been extended to incorporate facilities for the rest of the project team. Maestro is loaded onto a Philips P7000 processor, which can then be linked to the users mainframe computer or to other P7000s via Maestro-net. The various

members of the development team can then access the system via terminals or special micros such as the Philips P3200 for access to the graphics workstation.

The level of project management support amounts to basic planning and scheduling facilities with a bar charting graphics feature. However Philips are currently (June 87) in discussion with Hoskyn's regarding the possibility of linking PMW, the Project Managers Workbench, to Maestro.

Maestro specifically supports a number of design methods as follows:

Yourdon
De Marco
Chen and Merise

In addition to this the tool-set includes a powerful diagram editor which enables the user to define a desired set of symbols together with an associated set of rules. In this manner it can be said to support any graphical analysis and design tool.

A Maestro system is likely to cost somewhere in the region of #150,000 depending on precise requirements and specific configuration details.

General Purpose IPSEs

It must be said the major IPSE developments have taken place in this area. Those IPSEs which have so far emerged, tend to be rather closer to the Alvey description than their commercially orientated counterparts and are not surprisingly rather more expensive. Both IPSEs mentioned here are open ended, so it is for the user to decide on the specific method he/she prefers to follow. This is particularly true in relation to system development tools.

ISTAR is produced and marketed by Imperial Software Technology in London. It can be run on a variety of different processors or indeed networks of processors provided that they have UNIX. It will become available under VMS towards the end of 1987 hopefully.

ISTAR is an open ended second generation IPSE which seeks to offer support for all project staff across all phases of the life cycle.

Underlying any project is a contract data structure in which each project task is treated as a "contract" for a "client". For example a specification might have a contract associated with it detailing acceptance criteria, development schedule, reporting requirements and standards. The initial project is itself specified as a contract and can then be broken down into a series of subcontracts. The subcontract hierarchy is implicitly distributed so it is straightforward enough to set up an ISTAR system on a loosely coupled network of processors, where the transfer medium could be by direct linkage or even via floppy disc.

A user of ISTAR will create a set of "Transfer Items", which contain items of project work, in his/her work database. At the end of each session these "Transfer Items" are "Exported" to the appropriate contract database where they are gathered into logical groups called "Configuration Items". During each work session all relevant Transfer Items will be "Imported" from the "Configuration Item" in the Contract database and further developed using the ISTAR tool-set.

Periodically it is necessary to transfer Configuration Items between contracts within the hierarchy at which time the Configuration Item is "frozen" and cannot be further updated or removed by the user.

There are 4 tool-sets provided with the system:

- Management
- Technical Development
- Configuration Management
- Tool Building

The management tool-set contains a classical selection of tools to deal with both project planning and monitoring. There is no automatic transfer of information between the tools, but their input and output formats are compatible such that they are easy to use in conjunction with one another.

Only a few technical development tools are provided. These are clearly intended for real time developments particularly in the area of telecommunications. With the very wide variety of application areas it would be impossible to provide a comprehensive set of development tools, so the intention is that what is offered should be viewed as no more than a starting point. It is important to remember that ISTAR is open ended and can therefore be extended to incorporate any desired development tools depending on particular user requirements.

ISTAR offers configuration control essentially at two levels:

- i) via the contract database hierarchy with all the associated rules.
- ii) via the configuration management tool-set which provides:
 - * a means of manipulating Transfer and Configuration Items;
 - * version control including the distinction between "successors" and "variants";
 - * the ability to build a set of Transfer Items into a single logically meaningful Transfer Item;
 - * problem/error reporting and control;
 - * library management.

The Tool Building tool-set allows the user to expand the tool-set to match his/her specific requirements. This expansion can be achieved in one of two ways:

- * take an established method and develop a tool specifically for use within ISTAR;
- * package an existing tool so that it fits into the ISTAR framework and can provide information in a convenient format for use in other tools.

ISTAR clearly represents a very powerful new arrival in the computing community. Currently it will cost any potential user between \$200,000 and \$1,000,000.

GENOS

GENOS is being developed and marketed by GEC Software in London. Currently it runs under UNIX but it is likely to be available under VMS later this year (1987). The GENOS Framework is based on the Ten/Plus screen editor, one Framework process per user being started/terminated on entry/exit to/from GENOS. A distributed system is implicit via the GENOS Daemon, which is a single process resident on each separate CPU, handling all communication.

All project information is stored in a "Project Description File". This will be used to store a series of logical project work units or "Project Views". The project structure is specified by the level at which Project Views are defined. For example, a hierarchical structure can be specified by defining some Project Views within another Project View. Each Project View consists of three distinct components: Files (datasets), Tools (Tool-sets), Data flows, where a dataset represents either individual files or directories of files. These datasets form the "Virtual File Store" for a particular Project View. Then only files and tools allocated to a particular "Project View", are accessible to a user logged in to that "Project View".

A small set of specially designed tools may be used either independently or as part of a GENOS tool-set:

- GECOMO
- G-TASKPLAN
- G-CONTROL
- G-CM
- VADS

Like GENOS itself, they all use the Ten/Plus editor as their user interface. It might be noted that G-Taskplan and G-Control are not classical project management tools, but each effectively incorporate what is usually offered as a series of tools for project planning and control respectively. GECOMO is an estimating tool based on the COCOMO estimating model and VADS is an Ada toolset. Apart from configuration control facilities implicit in the actual GENOS Framework, no specific configuration control package is currently available, although G-CM emerged towards the end of 1987.

GENOS is very much an open ended environment, to the extent that no technical development tools other than for Ada development are offered. It is intended that any user can and should incorporate existing tools and/or user defined tools using the Task Description Language provided and the Ten/Plus editor.

IMPROVED RELIABILITY AND MAINTENANCE

For anything as powerful and all embracing as an IPSE to succeed and make a significant contribution to the computing industry, full commitment is essential from both management and technical users. Improved communications and a set of well defined access rights should result in more efficient system development, but will the final product be any more reliable? or any less in need of post delivery modifications and improvement?

System Reliability

The purchase of an IPSE will not in itself improve reliability, however if its facilities are used properly then in all likelihood more reliable software will result. The production of reliable software relies on a number of factors:

- * the use of an harmonious set of methods and tools throughout the development cycle to ensure that the software produced satisfies the users original requirements and is complete;
- * the generation of adequate test specifications, a test plan, test data and target results;
- * satisfactory quality control procedures.

With sufficient commitment and a reasonable open-ended IPSE, a toolset can be tailored to specific needs with a view to producing a comprehensive, integrated and appropriate toolset. The smooth interface between development tools within an IPSE together with an adequate set of verification and validation procedures will minimise the risk of accidental deviation from the system requirements. Care must be taken to ensure that the toolset incorporates adequate testing facilities to support a detailed test plan. Finally the fact that all project information is gathered together in a single area with full cross referencing should help to simplify the quality control task. In short, with the support of suitably detailed procedures, there is no reason why more reliable software should not be produced.

System Maintenance

The proper use of an IPSE should significantly reduce maintenance problems. System maintenance generally incorporates: system modification, system enhancement and general error fixing. A careful systematic approach is likely to minimise the need for error fixing, however the need for system modification and enhancement is unavoidable. A well structured product with full and meaningful documentation is essential. Consequently the comprehensive configuration control and extensive cross referencing facilities on a single logical project database commonly present in an IPSE, will prove invaluable.

CONCLUSION

Potentially, the arrival of IPSEs will have a very significant influence on system development and will doubtless reduce the effort requirements during maintenance. However it should be remembered that IPSEs are in there infancy and will doubtless mature and hopefully become less expensive in real terms over the next decade. Even at this early stage the 4 IPSEs mentioned, that is:

BIS/IPSE
Maestro
ISTAR
GENOS

have a lot to offer and are under continuous development.