SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

INFORMATICS DIVISION

SOFTWARE ENGINEERING GROUP NOTE 149

issued by
R W Witty
8 October 1986

The Alvey IPSE Strategy and its Relatioship to
UK Computer Science Teaching
Newcastle Seminar, September 1986

DISTRIBUTION:

R W Witty
D A Duce
C P Wadsworth
A J J Dick
D R Gibson
P M Hedlund
A D B Cox
B Ritchie
J Bicarregui
B Matthews
M Bertran
F M Russell
Mrs J Cheney
File: Newcastle/Randall

(see next page)

THE ALVEY IPSE STRATEGY AND ITS RELATIONSHIP TO UK COMPUTER SCIENCE TEACHING

DR R W WITTY
Director, Software Engineering
Alvey Programme

ABSTRACT

The Software Engineering component of the Alvey Programme plays a major role in promulgating the IPSE (Integrated Project Support Environment) concept and funding the development of actual prototype IPSEs.

After a brief introduction to the whole Alvey Programme and its Software Engineering component, the talk will concentrate on the Alvey IPSE/ISF (Information Systems Factory) strategy and its progress to date. Some mention will be made of projects contributing tools to the IPSE builders.

Some discussion of the industrial view of the IPSE concept will be followed by some thoughts on the implications of IPSE technology for UK academic teaching and research.

7 August 1986

AN OVERVIEW OF THE ALVEY PROGRAMME

INTRODUCTION

The Alvey Programme is based on the report of the Alvey Committee entitled "A Programme for Advanced Information Technology". The Committee's remit was to advise on the scope for collaborative research in IT in the light of increasing overseas competition highlighted by the Japanese Fifth Generation Computer Programme.

The Alvey Committee reported in September 1982 and recommended a five year programme costing £350m and involving collaboration between industry, academe and the three government departments involved, the Department of Trade and Industry, the Ministry of Defence, and the Department of Education and Science acting through the Science and Engineering Research Council. After consultation with industry and other interested bodies, the Government announced its acceptance of the main recommendations of the report and allocated £200m for the Programme, on the assumption that the remaining £150m would be found by industry. The Alvey Directorate was set up to manage the Programme under Brian Oakley who took up his post on 1 June 1983. A small Steering Committee was set up to oversee the work of the Directorate and Sir Robert Telford was appointed its Chairman.

Although the Alvey Directorate is located formally in the Department of Trade and Industry, it is, like the rest of the Alvey Programme, a collaborative effort. Its staff are drawn approximately equally from the five main contributors, that is from DTI, MoD, SERC, academia and industry, with industry generally paying the salaries of those of its employees seconded to the Directorate. The varied backgrounds of the staff and the fact that they are distributed over a number of locations make it a novel organisation and is another way in which the Alvey Programme is breaking new ground.

The Directorate came into being in June 1983 but most of the staff did not begin work until later in the year. Seven directors were appointed with responsibilities for specific parts of the Programme and for liaison with the three government departments involved and this determined the overall structure of the Directorate. The current Directors are:-

Director of the Alvey Programme	Mr B W Oakley
Deputy Director of the Alvey Programme and Director (Large Projects)	Mr S L H Clarke
Director (Infrastructure and Communications)	Mr K Bartlett
Director (Man-Machine Interface)	Mr S L H Clarke
Director (Very Large Scale Intergration)	Mr R Morland
Director (Software Engineering)	Dr R W Witty
Director (Intelligent Knowledge Based Systems)	Dr D Shorter
Director (Administration)	Mr R Hird

Alvey has a number of novel features of which the co-operative element is only one. It also differs from most other government support for R&D, and certainly that administered by DTI or SERC, in that it is not primarily responsive but is a directed programme. This means that there is a definite strategy with which projects must be consistent. The Directorate is also concerned to establishe the co-operative teams that are best fitted to implement the strategy. The strategy is not produced in a vacuum but is the result of extensive consultation and discussion with interested parties in industry as well as in academe.

The work is being carried out by consortia of firms, academic teams, and research establishments. Practice so far has shown that the average consortium consists of four or more bodies, typically two or three firms and one or two universities. Some consortia contain more partners though, except for work of a standards type, a consortia with many partners would become unmanageable. In virtually every consortium there are universities involved. In addition, there are a number of academic projects of a long range or highly speculative nature, on which it would be unreasonable to expect industry to be carrying out work at this stage in the development. Alvey requires such academic projects to have someone from industry who will take an interest in the work, steering it towards issues of industrial interest when appropriate, and warning the Directorate when industrial participation should be sought. These are known as "uncle" projects.

In all parts of the programme Clubs are being established to bring together and act as an information exchange for those in the UK IT community with a direct interest in the field. In general participants in consortia will be expected to keep the appropriate Club informed of the nature of their work, and the progress being achieved.

Hundreds of proposals have been received by the Directorate, and technical approval has been given to about 187 full industrial projects, and 116 academic "uncle" projects. There are over 100 companies in these approved projects, 65 academic bodies, and 20 Research Establishments. Much of the academic work is concentrated in a few centres of excellence where multi-disciplinary teams from different departments in those universities can be brought together to work on the Alvey projects.

The Alvey Directorate is responsible for developing policy and strategy in four key areas which have been identified as `enabling technologies´, and as such are attracting special support. In addition, the Directorate acts as the UK interface with the ESPRIT programme. There is a small staff of 7 directors. There are also committees which perform an advisory function to the Directorate.

The Directorate maintains contact with contract holders via a telecommunications network. There is in existence an electronic mail system between all participants in the programme, based on an existing SERC network and PSS for wide area communications.

THE PROJECT AREAS

The areas which have been chosen for support are software engineering, man-machine interface, intelligent knowledge based systems (IKBS), and very large scale integration (VLSI). In addition, there are a number of large demonstrator projects which are intended to develop research ideas into workable prototypes.

MAN-MACHINE INTERACTION

Ease of use of the technology was mentioned in the origial Alvey report as being an important factor in the design of information technology products. It is important for designers to incorporate the requirements of the user into the design as problems arise from the lack of communication between designers and their customers. The field of man-machine interaction is multi-disciplinary in nature, combining many aspects of research into human activity and organisational behaviour.

The area as a whole lacks a body of empirical research. The major areas that have been highlighted for consideration are pattern analysis, speech and image analysis, displays and human-computer interfaces.

VLSI

There are few companies actually producing chips in the UK, but 65% of the funds available have been allocated to this area. Production equipment such as etchers and electron beam machines need to be improved as well as improving the design of the chips themselves. Smaller companies are involved in producing this equipment.

Computer-aided-design for VLSI is receiving attention, but is difficult to develop because there are no standard systems in use in the UK. This should change in about 2-3 years time when firms are updating their equipment. The Alvey Directorate is keen to back UK silicon manufacturers because it feels that an indigenous industry is important for the UK.

IKBS

An IKBS system uses inference to apply knowledge to a task. Improving machine understanding of natural language is an important development for furthering the applications of IKBS.

In the UK research into IKBS is being undertaken by a small disparate research community, although there has been a rapid development of industrial interests especially in the expert systems field. Many embryonic research groups have been established, as well as new educational initiatives.

The natural language programme is the smallest one in this area, but is an important research topic because advances here will provide tools for ongoing research.

Standard languages are Lisp, Prolog, and Poplog, but these are limited by the number of dialects in each language.

There is an IKBS awareness and education programme which is surveying the use of expert systems used in business. An expert system starter pack has been produced to encourage firms to introduce expert systems into their business. These can be run on machines in general use, such as the IBM PC.

In addition, there are IKBS distance learning and journeyman schemes. The journeyman scheme is designed to give people employed in industry a chance to go back to university for a short period.

LARGE DEMONSTRATORS

To challenge the idea that the UK is always first in research but slow in exploiting new ideas commercially, the Directorate is funding a number of practical projects which are intended to develop research ideas into workable prototypes. Both industrial companies and universities are collaborating on the development of each project.

The demonstrators include, for example, mobile information systems, which will provide services to motorists linked by cellular radio. These include route guidance, traffic information, mobile electronic office, and mobile IKBS.

A speech-input word-processor and workstation is under development, which will incorporate machine assisted transcription, dictated speech, and will be speaker-adaptive and speaker-independent. This will be a first step towards fifth generation intelligent software.

A major project is a decision support system for the DHSS, which has been set up to design a large knowledge-based (legislation) system. This is intended to improve DHSS services to customers and provide a policy evaluation service. If this project is successful it could become the prototype for expert systems which incorporate legislation.

The `design-to-product´ project will demonstrate the automation of the total production process, from design through manufacturer to maintenance in the field. It represents the next generation of Computer Integrated Manufacturing and will use IKBS techniques throughout to capture, apply and amplify the skills of the human operators.

SOFTWARE ENGINEERING

AIMS

The SE programme will set out to establish tools and methods necessary for the production of high quality, cost effective software of world leading standard. It will be concerned with all stages of the software life cycle. Of equal importance the programme will seek to foster attitudes within UK management and their software staff that will regard SE technology use as normal practice.

In pursuing these aims the programme must recognise a number of important issues:-

- that in many areas existing market and economic conditions will be more significant than considerations of technical excellence.

- that the programme cannot sensibly develop a position that makes the UK totally independent of non UK products and components. It will, however, aim to minimise this level of dependence and establish strategies for "managing" the residual level, not only to keep down the UK import bill but also to make the UK less vulnerable to problems of access and supply.

- that much good work will be done outside the UK not least as a result of the major programmes being undertaken by the USA and Japan. The programme will need to maintain a sensible balance between "in house" work and "buying in" subject to the observation above. Overall there will be a preference for importing ideas and methods rather than products.

- that the need for good levels of collaboration has been strongly emphasised in the main Alvey report. The Software Engineering Programme will fully support this position.

Overall the aim is to provide within the UK, in the next decade, an infrastructure for the production of software that will support UK industry in a manner similar to that provided by Japanese Steel to Japan´s Manufacturing Industries in the 1950s and 60s.

MAJOR COMPONENTS OF THE SE PROGRAMME

The consultation that took place as a preliminary to the production of the Alvey Committee´s Report, and subsequently reconfirmed in recent informal discussions, showed that there are strong levels of agreement in Industry, Government and the Academic Community on the main directions the programme should follow. Three vital objectives are identified:-

(i) **Exploitation** - efforts are needed to ensure that existing methods are effectively used and their benefits gained by industry as a whole. Additionally, continuing efforts are needed to bring the fruits of research out into industrial use, with the associated investment and training.

(ii) **Integration** - work must be directed to establish the development of integrated methods and sets of tools for hardware and software development covering all phases of the system life cycle. The focus for such work will be the production of the Information Systems Factory (ISF) in which the UK will aim for technical leadership.

(iii) **Innovation** - programmes of research and development will be needed to extend the methods and techniques of software engineering. In particular this set of programmes will serve to establish a sound basis for the work undertaken in the **Integration** and **Exploitaion** activities.

In order to give an overview of the activities covered in the programme, together with the relationships that will exist between **Innovation, Integration** and **Exploitation,** figure 1 shows the process of system development sub-divided into the following elements:-

Methods and processes - how things are developed.

Management - the monitoring and control of the methods and processes.

Environment - the work place, tools and equipment used.

FIGURE 1

STRATEGY DEVELOPMENT ELEMENT	Innovation and Understanding	Integration and Implementation	Exploitation and Evaluation
Methods and Processes	Specification V & V Reliability Quality Reusability Metrics	Blend techniques into life cycle method for both hardware and software	Measure Use of Integrated Project Support Environments (IPSEs)
Management	Models of development and maintenance processes and methods	Integrate development methods with management techniques	Evaluate use of IPSEs
Environment	Influence on Productivity and Quality. Relationships with other technologies eg MMI, IKBS	Build Integrated Project Support Environments (IPSEs)	Make IPSEs available via Centres

INTEGRATION

The second major objective identified is for Integrated project support environments. The common understanding of an integrated PSE (IPSE) is that it should contain a compatible set of specification, design, programming, building and testing tools which supports a development methodology that covers the entire life cycle, together with management control tools and procedures, all using a co-ordinated and consistent project database. What is more, given the certainty that no one particular programming language will emerge in the medium term accepted as a standard meeting all needs, such PSEs will require multi-language capabilities. The position is further reinforced by consideration of the considerable levels of investment already made in the programming languages of the 1960s and 1970s.

Similar considerations apply to mixed hardware and software systems. It is clear that there are enough and increasing similarities between the hardware and software design processes, and the administrative and management procedures appropriate to them, for there to be benefit in aiming, in the longer term, to use one PSE for both hardware and software development. Furthermore, it is important that the requirements analysis, functional specification and much of the design work can be done independently of decisions whether particular modules should be implemented in hardware or software. For such modules, their function must be defined their place in the overall design established and their performance requirements known; economic, timescale and other criteria may then be used to determine how they should be implemented.

A fully integrated PSE as just described is exactly the Information Systems Factory, which is the focus of this programme. It is a long term objective for the end of the decade. It is, nevertheless, important to be clear about what the objectives are in order to see how to move towards them and in particular to determine the role of Unix and Ada APSE developments in this process.

The Software Engineering Programme includes two evolving IPSEs (Aspect and Eclipse) which not only bring together existing tools and procedures to improve development cost-effectiveness in the shorter term but are also capable of incorporating new techniques that emerge from relevant R and D projects.

Unix is being used as the basis for these two IPSE developments which are evolving from first generation file based tools to second generation, databased systems. These IPSEs are "open systems" in that they provide "public tool interfaces".

Unix is rapidly becoming a de facto standard over a very wide range of systems and organisations and therefore offers the prospect that:-

- there will be many developments for Unix which can be taken advantage of by the Alvey programme.

- the market for Unix-based development environments and tools is large and growing. These factors should minimise the amount of tool integration and development needed to improve today's Unix generation IPSE.

The second generation IPSE contains two major components not found in the first generation IPSE:-

(i) database-based tool set (rather than file-based) eg as illustrated in CADES.

(ii) distributed project support, eg as illustrated by the "Newcastle Connection".

As (i) and (ii) above are somewhat orthogonal, three approaches to the second generation are being attempted. Aspect and Eclipse embody the evolutionary approach, building on Unix, to produce databased, fully integrated IPSEs. A third project (IPSE 2.5) is a "clean sheet", non-Unix dependent approach. It is strongly biased towards supporting formal methods, and as its name implies, will attempt to include some third generation features.

THE 3RD GENERATION IPSE

The 3rd generation IPSE (or ISF), containing knowledge bases and `intelligent´ tools, requires significant research which must begin now if the 1989 target date for the Information System Factory design is to be met.

It is envisaged that the ISF will be defined as much by market and economic realities as by an technical goals; it will (almost be definition) embody the most cost-effective ways of producing application-specific IT systems available at the time.

An Information System Factory will probably consist of six main subsystems:-

1 Specification and prototyping facilities

2 a software Development Environment

3 a facility for CAD of VLSI and hardware development

4 a database or knowledge base of available software and hardware components

5 the communication systems, both local and wide area, to facilitate co-operative development

6 project management aids

How far advanced these six subsystems are by 1989, and how closely integrated together, depends on technical advances which are hard to predict. Markets will exist for the separate components as well as the unified ISF. The following sketches are probably optimistic in their assumed rate of technical progress, but help to define the aims.

1 Specification and Prototyping Facilities

Specifications of the system under development will be held internally in a formal, machine-manipulable form (which is central to the integration of the whole ISF - since it is used by all its subsystems). There will be extensive facilities to convey these specifications to people such as system designers and the eventual users of the application system - by animating the specifications, producing small prototype systems, question-answering and so on. Completeness and consistency of the specifications will be checked automatically.

2 Software Development Environment

This will go beyond present-day environments in supporting all phases of the software lifecycle and in relating them back to the formal specifications. It will be tailored to support one of several different development methodologies - depending on the application area - and will support a defined style of project management.

3 Facility for CAD of VLSI and Hardware Development

With the emergence of special-purpose hardware architectures implemented in VLSI, the need arises for functions to migrate between software and hardware during the lifetime of an application system. So CAD of VLSI cannot be considered as a separate problem. A VLSI-implemented system must meet the same formal specifications as a software system, and pass the same tests, and vice versa. Therefore the software development and the VLSI facility need to be centred around the same specification method and must communicate with one another.

4 Database of Available Components

To complete effectively in making IT application systems, it will be increasingly necessary to re-use existing software and hardware components. These components will be very diverse - a component could be a software product, an integrated circuit, a sub-routine or fragment of code, an algorithm, a man-machine interface device or one of a set of formal theories about data structures. A database of such components will hold some information common to all of them, to answer questions such as: What does it do? (ie its formal specification) What environment does it require? (language, storage space, power requirements, inter-connections etc) and Can it be adapted to perform a slightly different function? Some components will be general purpose and some will be application specific. Initially such information will be held in a database and searched in various ways; but in the longer term there is a need for automatic reasoning based on the data; this broadens the requirement to an intelligent knowledge based system (IKBS).

5 Communications Systems

A key feature of the ISF is the facility to allow groups of designers and programmers to work co-operatively, even when geographically distributed. This will mean a requirement for high bandwidth communication between co-operating processes, both within site, and between sites. It must be possible to implement the ISF in a distributed manner. It is expected that whilst the basic communications technologies of local and wide area networks will exist to allow this to occur, nevertheless considerable developments will be required to meet the special needs of the ISF, expecially in handling interactive high resolution colour graphics.

6 Project Management Aids

Project planning, management and control methods will be developed. When supported by a comprehensive collection of tools, these management techniques will provide both professional managers and technical staff with the ability to effectively plan and control all aspects of the software development process throughout the life cycle. These management tools must be intimately integrated into the development process to ensure that all the appropriate parameters can be realistically measured.

Thus an ISF, with all six subsystems implemented to a greater or lesser extent, will be an essential prerequisite to compete in producing medium to large scale information systems in the late nineteen eighties and nineties. It will represent a major capital investment for anyone intending to compete in the field.

The discussion so far has concentrated on the development of large, complex application systems; however, similar remarks apply equally to the small systems market. To remain competitive in producing IT products, companies will have to use advanced specification and prototyping tools, application development aids and libraries of components to produce better systems faster. So analogous small-scale Information System Factories may well dominate the small systems field, although market forces will drive them more to a low cost, high volume regime. The greater dynamism and adaptability of this sector means that new approaches are always rapidly emerging and can be rapidly tested in that market. Therefore the Alvey programme will by no means ignore the small systems market; producing and supporting small scale Information System Factories will be an important activity in its own right, as well as a testbed for ideas to be used in the large-scale systems market.

CONCLUDING REMARKS ON INTEGRATION

Thus the strategy for producing the three generations of IPSE requires a controlled set of concurrent and overlapping research and development activities. It is important that the 1st and 2nd generation IPSEs are produced, not just the 3rd generation ISF, because major gains are expected in software productivity and quality from their UK installation and exploitation as well as export scales.

The IPSE strategy proposal does not seek to establish Unix as a long term Alvey standard. It does, however, recognise that at this stage there is a need to establish a starting point which stands a chance of gaining general acceptance and having a wide relevance in the market place. It is clear that further work will be needed to protect against an undue level of dependence on the Unix base. Two lines will be developed:-

(i) An examination of environment portability

(ii) A vigorous drive to assist in establishing sound standards that will allow us to define more precisely our dependence and as a result, enable us to manage this dependence in a more systematic and positive fashion

Overall within the Integration component of the strategy, work will be undertaken to select and encourage the use of existing relevant standards and to sponsor the introduction of, and conformance, to new standards so that:-

(i) the process of defining and producing PSEs is assisted

(ii) the use of PSEs is encourage, since it will be expected that use will give rise to products that will conform to high quality and specification standards

(iii) external dependencies on components sourced from outside the UK can be managed more positively

SOME UK IPSE PROJECTS

1 ASPECT

- Alvey project
- Described at this conference

2 ECLIPSE

- Alvey supported
- Described at this conference

3 IPSE 2.5

- Alvey supported
- Described at this conference

4 ISTAR

- BT, IST private venture
- Described at this conference

5 BIS

- Alvey supported
- Not described at this conference
- Use of IKBS techniques for DP applications

6 THORN EMI

- In house production system
- Commercial product development and maintenance

7 GEC SOFTWARE LTD

- Not Alvey supported
- Unix and VMS based first generation framework
- Product on sale now (GENOS)

8 POPLOG

- Alvey supported
- Not described at this conference
- Sussex University and Systems Designers
- Programming support for LISP, POP 2, PROLOG

9 SAFRA, BAe

- Not Alvey
- In-house semi integrated support for `CORE´ requirements analyses plus SDL´s Perspective Programming support environment

ESPRIT PROJECTS/EUREKA

. PCTE, Sapphire, EAST, EMERAUDE
. GRASPIN (Petri nets and abstract data types)
. RAISE (VDM)

The development of standards is happening very early in the life cycle of IPSEs. Esprit is developing the PCTE, the Portable Common Tool Environment, initially based on Unix. One Alvey IPSE project, Eclipse, is using PCTE as its basis. The US DoD is developing the CAIS, the Common Apse Interface Set. The use of PCTE as a "public tools interface" is being evaluated by all Alvey IPSE projects.

SOME ALVEY PROJECTS CONTRIBUTING TO IPSEs

1 VDM Toolset
2 FORSITE (Z tools)
3 FOREST (Requirements)
4 Pascal Validation System
5 FORTUNE (documentation)
6 MDSE (IKBS for MASCOT)
7 Analyst Assist (IKBS tool)

Such Alvey projects as the above, which are producing tools, will be encouraged, or already plan, to install their tools in one or more of the Alvey IPSEs. Thus increasing the stock of available tools on the IPSEs. Details of these projects are given in the Alvey SE project poster sets given to delegates.

INDUSTRY VIEW OF IPSEs

Introduction

The industrial symptoms of the `software crisis´ are well known: projects are late and over budget in producing unreliable or unsatisfactory products which then prove to be extremely costly to maintain.

The DP community, in particular, complains that its systems are too difficult to develop, too slow to evolve to match the changing requirements of their business needs. This leads to a backlog of unstarted projects.

Management has difficulty in understanding the nature of software and the process of producing it. This is exacerbated by management´s inability to predict, measure, and control the software development process.

IPSE AS A PRODUCT

The IPSE may be viewed as a component in the software development process or as a product in its own right.

Viewing the IPSE as a product are the vendors of IPSEs and their customers.

The IPSE vendors are concerned with such issues as:-

1 How big is the market for IPSEs, now and in the future?

2 How can they make money from IPSE? By purchase, leasing, maintenance, upgrades?

3 Will IPSEs be sold as complete stand alone systems including hardware or will customers buy them as software products only?

4 Will IPSEs be a standard, mass produced item or will they be tailored to individual customer´s requirements ie is it a volume or a bespoke market?

5 How `generic´ or `application specific´ will IPSEs be?

6 Will the market be uniform or split into areas such as DP, embedded real time, home, small firm, large firm etc?

7 How will the customers be motivated? Will customers know what they want or will they need to be educated and helped?

8 What level of after sales service will be needed?

The IPSE customers are ultimately notivated to ask only one question:-

1 What will be the return on the investment in buying an IPSE?

Many businesses regard computing as an expensive overhead and will be reluctant to invest in even more new technology unless a clear case exists as to how it will increase their profitability.

A more enlightened firm will view an IPSE as the repository and guardian of a valuable company asset - the company's software. Even so, a wise management will want to know what return it is getting from this asset.

It is too early in the history of the IPSE for this clear cut accountancy argument to be demonstrable. However, without this evidence industry will be reluctant to take up IPSE technology, so the Alvey Programme is ready to encourage the trial users of IPSEs to evaluate their cost-effectiveness.

The USA has traditionally been more willing to invest in capital items to improve the productivity of its workforce. The Japanese too are always looking to invest in improving their production methods. UK industry is not always as ready to invest in new technology, especially if it requires an element of vision and faith by management because the accountancy argument is not extant.

The independent UK software industry has a tradition of low capital investment per worker. Even today many staff do not have their own terminal and access to powerful computing facilities, let alone a SUN-type single user workstation on their desk and a terminal at home.

A question which the Alvey Programme often asks industry is what happens to the UK if IPSEs are way ahead and we fail to make the necessary investment in time?

The business case for IPSEs must rest on the demonstration of real progress in one or more of the following areas:-

1 demonstrable improvement in quality of software

2 demonstrable improvement in development productivity

3 demonstrable reduction in maintenance/life cycle cost

IPSEs will probably be purchased by customers who have existing software development facilities, existing software systems and whose business requirements change with time. Thus potential IPSE customers are concerned about how costly and disruptive will be the introduction of IPSEs. Is there some smooth, easy transition route from today's methods? Or will it be a traumatic step change involving massive switch overs in hardware, methods etc which will need massive staff retraining or recruitment?

How can a firm's existing applications software be supported by a new IPSE? Is this possible or will it amount to a re-write?

How will the IPSE concept fit in with the new role of the end user as a 'programmer' who uses 4 GLs, high level database query languages etc to prototype or implement his own applications directly?

Will the purchase of an IPSE lock-in a customer to that IPSE vendor's system in the same way as firms are now often locked-in to manufacturers' operating systems?

Will the majority of companies buy an IPSE if its not from IBM?

Will companies buy IPSEs if no standards exist for tool portability, end product portability etc? If IBM does not lay down a de facto standard will current initiatives such as the USA/DoD´s CAIS and Esprit´s PCTE give a coherence to the market place? Or will premature standardisation, possibly two competing standards at that, fossilise an underdeveloped concept before it can blossom?

IPSE AS A PROCESS

An IPSE is part of the software development process. As such it will strongly influence the way companies actually go about developing and maintaining software.

The act of actually developing software is just one activity in a business, even for a software house. So the customers who buy IPSEs will be looking to see how well IPSEs fit in with their other activities. For example, do the IPSE documentation facilities integrate cleanly (are the same as) the office automation system if one is in use? Do the IPSE project management facilities match those in wider use in the company? Does the IPSE work harmoniously with other development facilities such as VLSI/CAD for electronic systems companies or with CAD/CAM for general manufacturing companies or with the corporate information system for financial services companies?

The IPSE is part of the `process´ aspect of a business and so is likely to increasingly influence the competitive performance of more and more companies as software is used in an expanding range of products and services.

Thus the `integration´ aspect of the IPSE is a multi faceted thing. Integration must exist between tools, between data held about software components, between the interfaces shown to the IPSE user, and between the IPSE and the other corporate technical and data processing software systems.

IMPLICATIONS OF THE IPSE FOR COMPUTER SCIENCE TEACHING

For this discussion it is assumed that the IPSE approach to software development and maintenance becomes the `norm´ in the next 10 years. What effect, if any, will this have on Computer Science teaching?

This question immediately raises another. Are the universities aiming to produce `vocationally´ trained computer scientists who can earn revenue for industry as soon as they graduate or are the universities trying to educate engineers in the fundamental principles of computer science? As computer science develops and the volume of both basic theory (eg predicate calculus) and relevant practical knowledge (eg SSADM) expands it will become increasingly difficult to teach a student, in 3 years, all he needs to know to be a fully trained software engineer ready for industrial work.

As more sophisticated methods and tools are developed, the software engineer will need more education and more training, over their whole career span, than he/she receives today.

Can or should a university undergraduate course convey the problems of software engineering which arise from the scale of industrial activity?

If the IPSE, as used by industry, is a major investment then will the universities be able to afford IPSEs to give realistic training to students? Is this a sensible thing to attempt?

A typical second generation (in Alvey term) IPSE will probably comprise:-

1 SUN workstation per person @ £20K each

2 LAN throughout buildings, say £250K

3 Servers for printing, filing, say £200K

4 Big minicomputer, say £250K

5 Software, say £100K

which adds up to a capital cost of well over £1 million with recurrent costs of say £200K per annum. Can the universities afford this level of investment to teach students what to expect in industry?

Perhaps as suggested by the DoD's SEI, universities will construct a `model' IPSE from smaller, less costly components but which will be adequate to give students the feel for an integrated environment supporting all aspects of a project's activities.

This approach still carries the risk, as can happen in today's courses, that students do not really understand that industrial software engineering tackles problems of scale and maintenance; it hardly ever develops software by a clean sheet approach.

The Alvey SE strategy discusses the software process in terms of methods, skills and tools. Computer Science can, should and does teach methods. CS can, should and does teach some of the skills. However not all SE graduates are skilled 10 finger typists which is an essential skill, as will `mouse' dexterity soon be too. Interpersonal and management skills are vitally important but primary responsibility will be for industry to develop these. Word processor and office automation skills are increasingly important to being a good software engineer. Knowing how to manage through such tools in an emerging skill.

Today we are discussing tools. Can, should or do CS departments teach using the same tools as are used by industry, and/or will be used by industry when current research activity is transferred from universities to industry?

This paper offers no answers to these questions. It is merely trying to stimulate a discussion whose central themes are:-

1 Will the IPSE become the `normal' mode of industrial software development?

2 Should UK universities have IPSEs to train students?

3 Is it realistic for universities to have the same facilities as industry?

4 If the subject continues to expand should UK universities train in current industrial practice or educate in fundamental theory?