PCTE (PORTABLE COMMON TOOL ENVIRONMENT)

A. Alderson

Rapporteur: Mr. G. Tomlinson

1

PCTE is an ESPRIT project whose primary output is the specification of an interface upon which IPSEs may be implemented.

I will begin by outlining the project and the history behind it.

I will then describe the elements of the interface and consider the Object Management System (OMS - the database) and the User Interface in more detail.

Finally, I will discuss the politics that now surround the development of PCTE.

SLIDE 2

PCTE is not an IPSE. It is a basis on which to develop toolsets and IPSEs.

The ESPRIT Software Engineering Task Group recognised that within ESPRIT there would be a need to transfer software, as well as results, between the parts of the ESPRIT Programme. To enable this a common basis of powerful basic mechanisms for building and operating tools was needed. This basis had to be portable between many different machines. The PCTE project is producing this basis.

The need to develop this basis on a timescale useful to the rest of ESPRIT is a major constraint on the project and in many ways has limited their ambition and focused attention on pragmatic solutions.

The major decision has been to advance the portability objective by basing PCTE on UNIX System V. PCTE is a System V Interface Definition very cleverly extended.

A by-product of this decision has been the compatibility which PCTE maintains with the wealth of software which exists for UNIX. Such software is compatible witn PCTE at binary level.

The other major decision was to have networks of powerful single user, bit-mapped screen workstations as the target hardware and so to permit very advanced user interfaces to be defined.

slide 3

There are six collaborators in the PCTE project.

Bull supply the project management and are responsible for the basic enhancement of UNIX including the OMS. Bull, as we shall see later, have other interests in PCTE.

Siemens are responsible for the User Interface. ICL are responsible for the distributed nature of PCTE.

Nixdorf and GEC are developing applications to test the value of PCTE. Nixdorf are producing the Configuration Management System (CMS) and GEC are producing the Knowledge Based Programmer's Assistant.

Olivetti are developing an Ada based version of PCTE.

slide 4

The origins of PCTE are interesting and include most of the major European initiatives into IPSE design.

The oldest identifiable root is the Ada Support System Study carried out in 1979 by Software Sciences and Systems Designers for the MoD. This was in turn a major input to Stoneman which was produced for the US DoD.

The Ada Compiler Study by Bull and Siemens was another root which in itself led directly to Alsys.

Stoneman was the source for the UKAPSE Study undertaken by Software Sciences for the DoTI. Interestingly Olivetti were involved and one of the PCTE designers Nando Gallo was involved. His experience together with input from the Ada Compiler Study was brought to the definition of PAPS, and Ada Program Support environment innvolving Olivetti, DDC and Christian Roffsing.

In parallel Bull had undertaken the ALPAGE development. This also sprang from Stoneman and involved Olivier Roubine, a major contributor to PCTE.

The scene was now set for PCTE which commenced in 1983 with the collaborators we have already introduced.

The PCTE project however is only specifying the interface and producing prototype implementations. The first commercial quality implementation will be Emeraude whose partners will be BULL, Syseca and Eurosoft. This implementation will be for the Bull SPS7.

Recognising the need for other implementations ESPRIT have funded the Sapphire project to produce Emeraude based implementations of PCTE on the SUN, VAX, VAXstation, and IBM-PC. The Sapphire partners are Software Sciences, CAP, GIE Emeraude and Aberystwyth.

ESPRIT have also funded the PACT project involving Bull, to develop basic support tools, such as a schema definition processor for PCTE, and the PAVE project led by GEC to develop and VAX/VMS implementation of PCTE.

ESPRIT are also to fund the very broadly based SFINX consortium, which will port the products of ESPRIT projects to run on the Emeraude implementation of PCTE.

Finally, to ensure fair play, the CEC is to set up a standardisation body to be responsible for the specification and evolution of the PCTE interface.

slide 5

The PCTE interface addresses all of the support mechanisms expected of a modern operating system.

Execution mechanisms address the execution of programs as processes. This is similar to UNIX but a process must have a working schema, which details the parts of the database it wishes to use. The usual UNIX calls are available (FORK, EXEC) plus some new ones to START or CALL other programs without forking.

Communications mechanisms permit the process to communicate with the database and with users through FILES, DEVICES and the VIRTUAL TERMINAL. Again the expected UNIX calls are used.

Interprocess communication permits process to process communication through PIPES, MESSAGES, SIGNALS and SHARED MEMORY. The MESSAGES provide extended facilities over UNIX System V. Again familiar UNIX concepts and calls are involved.

The OMS is discussed in detail later.

Activities provide concurrency control. Each process involves an activity which may acquire resources. Acquiring a resource requires a lock to be associated with that resource. These locks vary in strength from the usual UNIX lock, anyone can do anything at anytime, to single process access. Locks can be released explicitly, or may be released explicitly (e.g. end of process). If locks are made in the context of a transaction, then any resource still locked if the transaction is aborted will be restored to its state at time of locking. Transactions can be nested.

Distribution provides for user access to the resources of all processors on the network and access to the distributed database. It is effectively transparent, although the user can exercise control using facilities of Execution Mechanisms for processes and by assigning volumes of the database to particular disk drives.

In order that existing UNIX tools may be compatible with PCTE, the UNIX system calls must be emulated. Many of the calls occur in areas already discussed. The major problem is the emulation of the UNIX directory structure since PCTE uses objects to implement PCTE directories.

The User interface is discussed later.

slide 9

PCTE presents an Entity, Relationship, Attribute data model of entities (called objects) with attributes and relationships. However, objects may have a special attribute called the Context. This is not accessed as an attribute but using UNIX file primitives. It acts as the contents of a file. The OMS has schema facilities which allow the definition of objects. PCTE arranges that UNIX filestore can be defined as a special case of this data model, so that existing UNIX tools can execute under PCTE.

The PCTE OMS is in fact a very clever generalisation of UNIX filestore. The Content represents the file contents while the ability to apply attributes to the object is seen as an extension to the attributes UNIX holds for files (owner, creation date, access permissions etc).

The elements of a pathname are interpreted as values of Link attributes in the PCTE OMS. By defining an appropriate schema an existing UNIX tool is unaware of in this change of interpretation.

slide 7

The attributes and relationships which kinds of objects are to have are declared in the schema.

PCTE views the schema as the means of integrating tools around commonly accessed data structures.

The schema is defined by the aggregation of SCHEMA DEFINITION SETS. Each SDS may define some object types and some attributes and relationships.

The working schema is a tools external schema. It is a partial view of the actual schema. It indicates what data the tool has an interest in. The working schema is an aggregation of SDS.

SDSs may be added or modified at any time without 'stopping' the system. Of course, there is no effect upon a Working schema until the new SDS is added to it.

slide 9

PCTE offers a very limited set of attribute types - integer, date, string and boolean. They are all scalar types.

Attributes in PCTE have initial values which are supplied either in the schema or by default by the system. There is no concept of the unset value.

There are no composite types such as set, bag or sequence which I consider a limitation for the efficient construction of software engineering tools.

[illegible]

[illegible] extended relationships [illegible] great power.

They are intersected [illegible] rules. If one user [illegible] generated and [illegible]

[illegible]

Name

Access [illegible]

Inherit : [illegible]
[illegible]

:

:

Visible [illegible]

Definition type [illegible]

Key access [illegible]

slide 11

The interpretation of pathnames as links is illustrated by the simple example of the path:

FRED.C

This is interpreted as:

the current object has a link of name C which is arity many. The particular link to be chosen as the one with key FRED.

A particular link may be registered as preferred. If C is preferred then FRED identifies the same object as FRED.C.

There is a single (virtual) object which represents the root of the database. Thus we may interpret UNIX pathnames such as

/USR/USERS/ALBERT/FRED.C

These are also representations of HOME(~), etc.

slide 12

The PCTE User Interface is intended primarily for advanced single-user workstations offering a bitmap display and a mouse. It is intended for a multi-process environment, where several parallel processes may require user interaction. The User Interface provides windows each of which emulates a terminal. The user changes working context simply by interacting through a different window.

Windows may overlap each other and a particular window may be 'buried' or 'popped into view'. They may also be iconised.

Windows are simply structures through which data may be viewed. The data is held in a frame which is a data structure of type: text, multi-font text, graphics or bitmap. The tool interacts with the frame and the user views data by moving the window over the frame.

Three sets of functions are provided for:

- basic editing to allow the end user simple and easy modification of objects on the screen,
- window management,
- menu management to display lists of command options from which one or more may be chosen. Menus may be both 'static' or 'pop-up'.

slide 13

Architecturally there are two processes involved in the implementation of the User Interface. The Application, or an Application Agent process, interacts with a frame through a virtual terminal (a PCTE object type). The user interacts, through a User Agent process, with the window.

All user interaction is interpreted by the User Agent. Some interactions involve changing the content of the window by moving a viewport over the frame. The viewport is actually just a mapping which depends upon the frame type. Other interactions involve changing the content of the window and hence the frame by editing. The User Agent and Application may need to interact to ensure that the frame contains the required data.

The current specification allows only one viewport per window and one viewport per frame. It is intended that multiple viewports per window (and perhaps per frame) will be introduced later.

slide 14

Finally it is worth taking a brief look at the politics surrounding PCTE, since they provide a good example of the difficulties which beset stragegic products such as IPSEs.

PCTE represents an area of conflict between the UK DoD (AJPO) and various European interests.

The UK DoD initiated the CAIS Programme (Common Ada Interface Set) in response to Stoneman. CAIS is intended as the interface for all APSEs. The first attempt CAIS1 has been specified and implementations have been developed. However, CAIS1 was panned by the critics.

UK DoD responded by commencing specification of CAIS2 and awarding an implementation contract to Softech.

However there is a third aspect. There is a Requirements and Definitions document (RAD) specifying the requirements for a CAIS interface. Interestingly not just CAIS1, but CAIS2 also, fell short of meeting these requirements. Furtner PCTE is closer to meeting the RAD, than either CAIS specification.

The ideal, and technically most sensible solution would be to converge CAIS and PCTE to a universal standard meeting the RAD. However, DoD AJPO will not countenance this.

This has to be seen in the context of the US attitude to advanced technology which is becoming increasingly protectionist. It must also be seen in the context of their attitude to NATO as reflected in the Nunn Amendment, which reflects US irritation at what they see as the low level of expenditure by European nations on defence. This they believe entitles them to dictate to the others technologically.

This is important because the DoD, MoD and NATO commitment to Ada makes defence the major area for IPSE sales.

Europe could respond with a PCTE2 specification which meets the RAD, but given the US attitudes, it is not clear what benefit this would have.

slide 15

The European initiative is now strongly behind PCTE. Esprit, the French Government, Alvey, and possibly even MoD, are making major financial commitments to PCTE implementation, utilisation and standardisation.

My personal view is that Europe should ignore CAIS and develop PCTE. We must not forget the US propensity for embargoing advanced technology sales, especially when they are not doing the selling. It would be difficult for them to interfere with PCTE based sales.

The choice of course is a difficult one to make, but all current evidence is that Europe is ahead in this branch of technology. If we wish to stay there we must reject CAIS and develop our own ideas currently PCTE and perhaps EAST in the future.

PCTE

A BASIS FOR A

PORTABLE

COMMON

TOOL

ENVIRONMENT

PCTE

NOT AN IPSE ITSELF

A COMMON BASIS FOR TOOLS
IN ESPRIT

POWERFUL BASIC MECHANISMS FOR
BUILDING AND OPERATING TOOLS

PORTABILITY BETWEEN MACHINES
DERIVED FROM UNIX

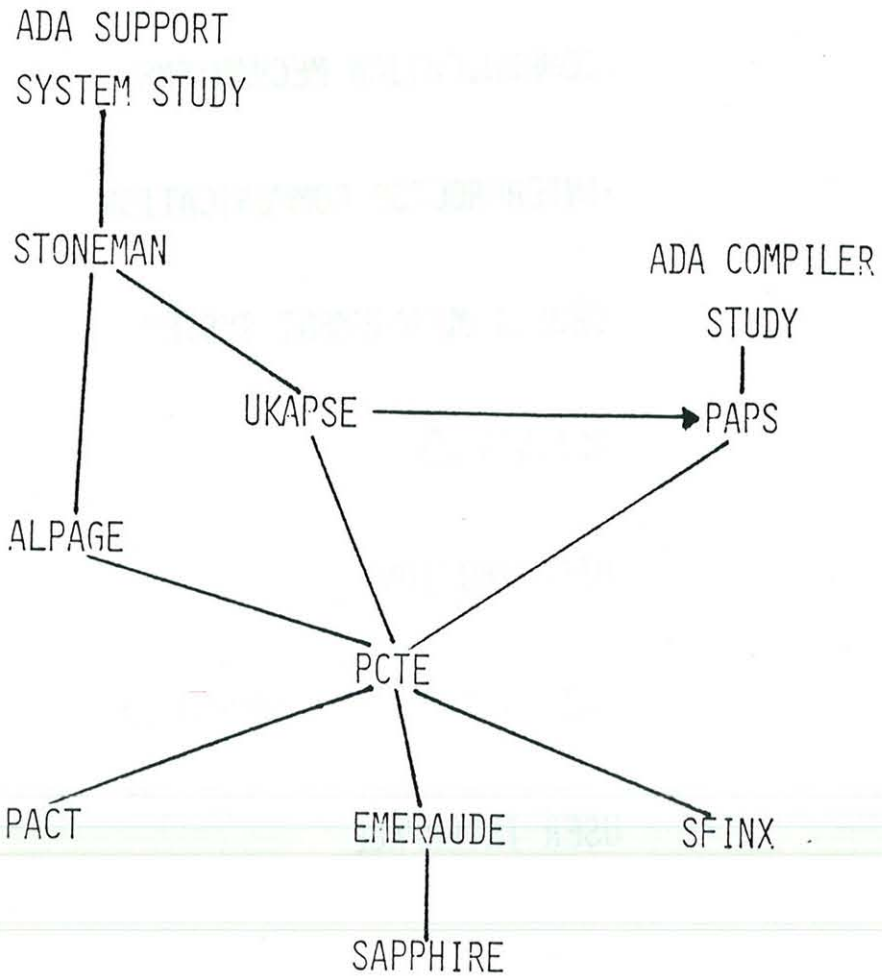
COMPATIBLE WITH EXISTING
UNIX TOOLS

TARGET ARCHITECTURE POWERFUL
SINGLE USER, BITMAP SCREEN
WORKSTATION

PCTE COLLABORATORS

BULL	-	BASIC MECHANISMS OMS
SIEMENS	-	USER INTERFACE
ICL	-	DISTRIBUTION
NIXDORF	-	CMS
GEC	-	KBPA
OLIVETTI	-	ADA BASED VERSION

ORIGINS OF PCTE



PCTE

EXECUTION MECHANISMS

COMMUNICATION MECHANISMS

INTERPROCESS COMMUNICATION

OBJECT MANAGEMENT SYSTEM

ACTIVITIES

DISTRIBUTION

UNIX SYSTEM CALL EMULATION

USER INTERFACE

OBJECT MANAGEMENT SYSTEM

OBJECT

A DATABASE ENTITY

WITH:

ATTRIBUTES

RELATIONSHIPS

SPECIAL ATTRIBUTE

CONTENT (OF FILE)

GENERALISES FILESTORE

UNIX IS A SPECIAL CASE

SCHEMA

THE MEANS OF INTEGRATING TOOLS
AROUND COMMONLY ACCESSED DATA
STRUCTURES

SCHEMA IS A COLLECTION OF OVERLAPPING
SCHEMA DEFINITION SETS (SDS)

WORKING SCHEMA

A TOOLS EXTERNAL SCHEMA
A PARTIAL VIEW OF ACTUAL SCHEMA
COLLECTION OF SDSs

INCREMENTAL SCHEMA MODIFICATIONS ALLOWED
BY ADDING, MODIFYING SDSs

OMS

OBJECT TYPES

BASIC TYPES

- FILE
- PIPE
- MESSAGE_QUEUE
- CHAR_DEVICE
- BLOCK_DEVICE
- OBJECT (HAS NO CONTENTS)

EVERY OBJECT HAS AN ANCESTOR TYPE WHOSE PROPERTIES IT INHERITS

EVERY OBJECT MAY HAVE FURTHER PROPERTIES

TYPES FORM A HIERARCHY

OMS

ATTRIBUTE TYPES

INTEGER

DATE

STRING

BOOLEAN

NO COMPOSITE TYPES

HAVE INITIAL VALUES

LINKS

UNDIRECTIONAL POINTER FROM ORIGIN
TO DESTINATION OBJECT
BUT MAINTAINED AS BIDIRECTIONAL PAIRS

LINKS HAVE TYPE CHARACTERISED BY:

NAME
ARITY (ONE, MANY)
CATEGORY (COMPOSITION, REFERENCE, IMPLICITY)
STABILITY
DESTINATION TYPES
KEY ATTRIBUTES
ATTRIBUTES

INTERPRETING PATHNAMES

FRED.C

- ARITY MANY LINK NAMED C OF CURRENT ENTITY
- KEY VALUE FRED
- IDENTIFIES ENTITY POINTED AT BY THE LINK

FRED

- C PREFERRED
- IDENTIFIES SAME ENTITY

THERE ARE INTERPRETATIONS OF

ROOT /

HOME ~

PARENT ..

GIVES UNIX PATHNAMES MEANING
/USR/USERS/ALBERT/FRED.C

USER INTERFACE

MULTIPLE WINDOWS

FRAMES

- TEXT
- MULTI-FONT TEXT
- GRAPHICS
- BITMAP

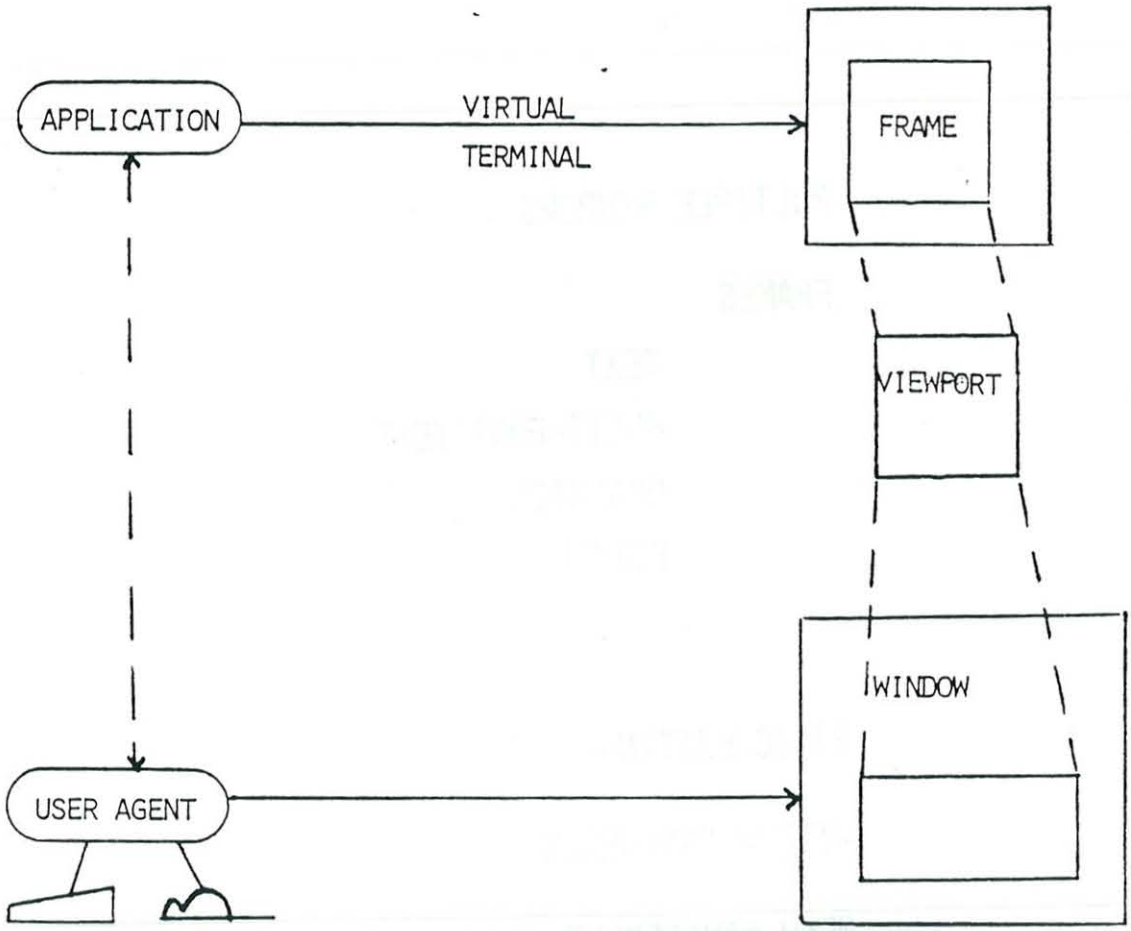
BASIC EDITING

WINDOW MANAGEMENT

MENU MANAGEMENT

PCTE

USER INTERFACE



FRAME TYPES:

- TEXT
- MULTI-FONT TEXT
- GRAPHICS
- BITMAP

THE POLITICS

US DOD V EUROPE

US CAIS PROGRAMME

CAIS1

CAIS2

RAD

US NUNN AMENDMENT

MOD NATO ADA

EUROPE PCTE2