Chapter 27

Chapter 27 Software Engineering Environments

Learning Objective
... define the basic *support infrastructure* provided for software engineering

Frederick T Sheldon
Assistant Professor of Computer Science
Washington State University

Software Engineering Environments

Integrated environments to support large-scale software development

Objectives

- To discuss the advantages and disadvantages of software engineering environments
- To define the basic support infrastructure provided by software engineering environments
- To describe an architectural model for software engineering environments and the services associated with the reference model
- To introduce PCTE, a proposed framework standard for software engineering environments

Topics covered

Integrated environments
Platform services
Framework services
PCTE

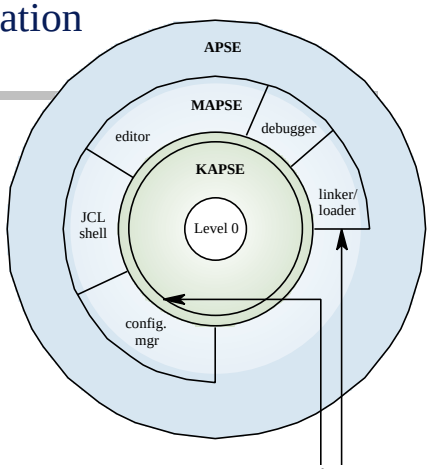
The evolution of environments

The notion of a software engineering environment was proposed in 1980 in the proposals for an Ada Programming Support Environment (APSE)

Three level model

- Kernel APSE extends OS facilities to provide basic infrastructure support
- Minimal APSE provides programming workbench facilities for Ada
- Full APSE provides a complete software engineering environments

The organization of an APSE



Environment use

APSE proposals were far-sighted but environments have not come into common use

- Difficult to define and implement a standard environment kernel
- Cheap PCs encouraged production of simple CASE workbenches
- Data integration requirements are poorly understood
- Impossible to develop UI standards because of rate of change of hardware technology
- CASE benefits less than expected
- Geopolitical changes in late 1980s meant that less defense funding was available

Integrated environments

A software engineering environment (SEE) is a set of hardware and software tools which can act in combination in an integrated way to provide support for the whole of the software process from initial specification through to testing and system delivery

Still a need for SEEs in large projects. These will probably be based around a standard framework

SEE characteristics

The environment facilities are integrated - should provide platform, data, presentation, control, and process integration

The environment is designed to support team-based activities. Configuration management is fundamental to this

Facilities are provided to support a wide range of software development activities

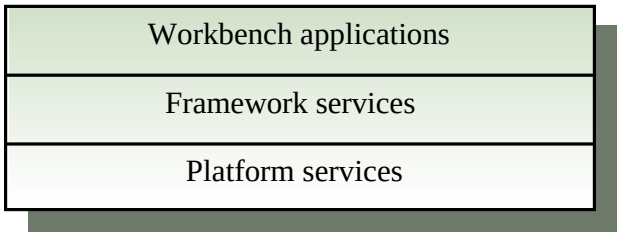
Service model of an SEE

A layered SEE architecture views the system as a number of layers where each layer provides some services to other layers

Platform layer provides basic file, process management and network services

Framework layer provides data management, message and user interface services

Layered model of an SEE



Environment perspectives

Application software developers see the environment as a set of CASE workbenches

SEE integrators see the environment as a set of common services and tools which must be integrated in a particular context

Tool developers see the environment as a set of common services called by tools

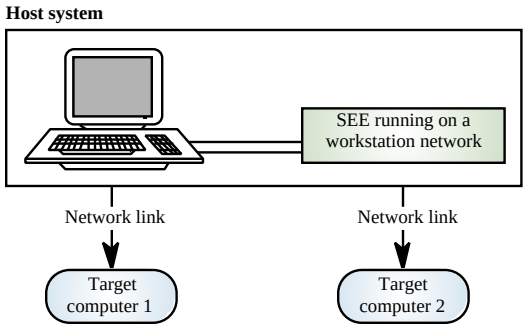
Framework developers see the environment as a set of services which must be implemented

Host and target platforms

The SEE runs on a host platform but the software is often developed for some different target computer

- The software may be for a machine with no development facilities (e.g. a hand-held computer)
- The target machine may be application-oriented (e.g. a parallel processor) and not well-suited to run CASE systems
- The target machine may be used for some other application which must take priority

Host-target development



Platform services

File services

Process management services

Network services

Communication services

Window management services

Print services

These services are usually provided by a distributed network of workstations

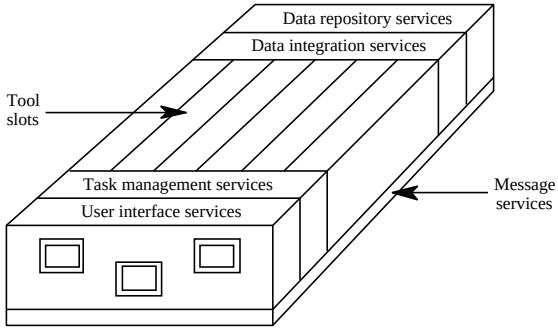
Framework services

Framework services extend the platform services and provides specialized SEE support

Can be discussed in terms of a reference model (the toaster model) which identifies 5 sets of services

- Data repository services
- Data integration services
- Task management services
- Message services
- User interface services

An SEE reference model



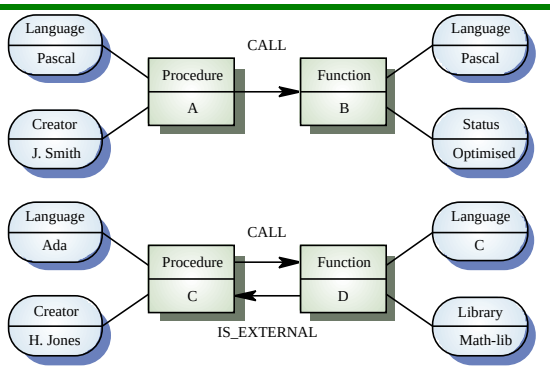
The data repository

Repository provides an object management system (OMS) for naming and managing entities and establishing entity relationships

Repository granularity reflects the minimum size of entity which may be stored and manipulated

- Coarse-grain repository usually manages entities such as files
- Fine-grain repository can manage individual declarations and parts of programs or designs
- Fine-grain systems typically require many more DB accesses than coarse-grain systems

OMS entities and relations



Data repository services

Service	Description
Data storage	Provides support for creating, reading, updating and deleting entities where entities are named, havea set of attributes and may participate in relationships.
Relationship	Provides support for defining and managing relationships between environment entities.
Name	Provides support for entity naming. Entities also have a unique identifier which is assigned by the repository services.
Location	Provides support for the distribution of entities over a network of workstations so has associated operations such as move, copy, replicate, etc.
Data transaction	Provides support for atomic transactions which allow database recovery in the event of a failure.
Concurrency	Provides support for multiple simultaneous transactions.
Process support	Provides process operations such as start, stop, suspend, etc.
Archive	Provides support for the off-line storage and recovery of entities.
Backup	Provides support for recovery of data in the event of system failure.

Data integration

Extends basic repository services to provide specific services for software development

These services may be used as a basis for an integrated configuration management system

Meta-data services allow sub-environments to be created thus allowing different projects to work in their own environment

Data integration services

Service	Description
Version	Provides support for the management of multiple versions of entities.
Configuration	Provides support for entity grouping into named configurations and managed as a composite entity.
Query	Provides access and update services to versions.
Meta-data	Provides facilities for schema definition and management.
State monitoring	Provides triggering facilities which allow particular operations to be initiated when a particular database state is reached.
Sub-environment	Provides support for the definition and management of subsets of the data and operations in the environment and to consider them as a separate, named environment.
Data interchange	Provides mechanisms to import and export data from the environment.

Task management

Concerned with providing support for process integration

Provide operations to define and execute process models

Least well-defined services in the SEE reference model because of the immaturity of this field

Task management services

Service	Description
Task definition	Provides facilities for task definition including pre and post conditions, inputs and outputs, resources required and the roles involved in the task.
Task execution	Provides facilities for supporting the execution of tasks. This may involve specifying task interactions in a process programming language.
Task transaction	Provides support for transactions which involve one or more task executions over a considerable period of time. Recovery from failure should be possible without rolling back the system to its state before the task started.
Task history	Provides facilities to record task executions and to query previous executions.
Event monitoring	Supports the definition of events or triggers which cause some task to be executed.
Audit and accounting	Provides a record of what has been done and what resources have been used in the environment.
Role management	Provides facilities to define and manage roles in the environment.

Message services

Allow tools and framework services to communicate

Two services defined in the SEE reference model

- Message delivery service. Supports tool-tool, service-service, tool-service and framework-framework message passing
- Tool registration service. Allows a tool to register with the message server to receive messages

Implemented in commercial products such as HP's SoftBench

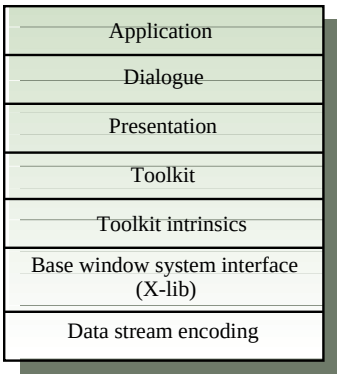
User interface services

Support presentation integration. Based on the X model of user interaction

Higher levels of model are not well-defined so don't allow comparison of environments

Seem to be based on the assumption that platform services will be provided in a UNIX machine. This is not necessarily the case.

User interface reference model



Environment tools

Integrated tools

- Tools which manage all of their data using the framework services and implement their data structures in the object management system

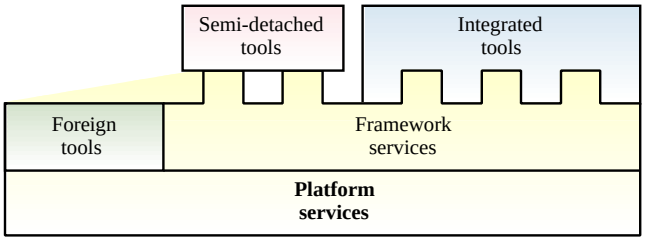
Semi-detached tools

- Less tightly integrated tools. They manage their own data but the files in which that data is stored is managed using framework services

Foreign tools

- Tools which run on the same platform as the SEE but which only make use of platform services

Tool integration



Tool migration

Relatively easy to migrate existing tools to SEEs as foreign tools and, for open workbenches, as semi-detached tools

Full power of the environment can only be realized when most tools are integrated tools

However, to integrate tools, you need a framework BUT no-one will buy a framework until there are integrated tools

PCTE

PCTE (Portable Common Tool Environment) is the best developed candidate for an SEE framework system

Provides data repository and data integration services but must be combined with other systems (e.g. SoftBench and Process Weaver) to provide a full range of environment services

PCTE and the SEE reference model

Services	Description
Data repository	All data repository services are provided by ECMA PCTE with the exception of a backup service.
Data integration	Provides all data integration services apart from a general query service. Some services such as state monitoring and data interchange services are more limited than those proposed in the reference model.
Task management	No task management services are provided apart from auditing and accounting services.
Message	Provides a service for message delivery but no tool registration service.
User interface	It is assumed that PCTE-based environments will use X-windows for implementing their user interface. No specific library is mandated.

Key points

An SEE provides support for a wide range of process activities

SEEs should provide the five levels of tool and workbench integration which have already been discussed

SEEs are generally designed for host-target working

Platform services which should be provided include file, process management, network, communication, windowing and printing services

Key points

Tools which are part of an SEE may be foreign tools (integrated through the platform services), semi-detached tools (integrated through coarse-grain objects) or integrated tools (integrated through fine-grain objects)

ECMA PCTE has been accepted as a potential framework for SEEs