Lecture-28

Computer-Aided Software Engineering

Computer-Aided Software Engineering

"Automating the process ..."

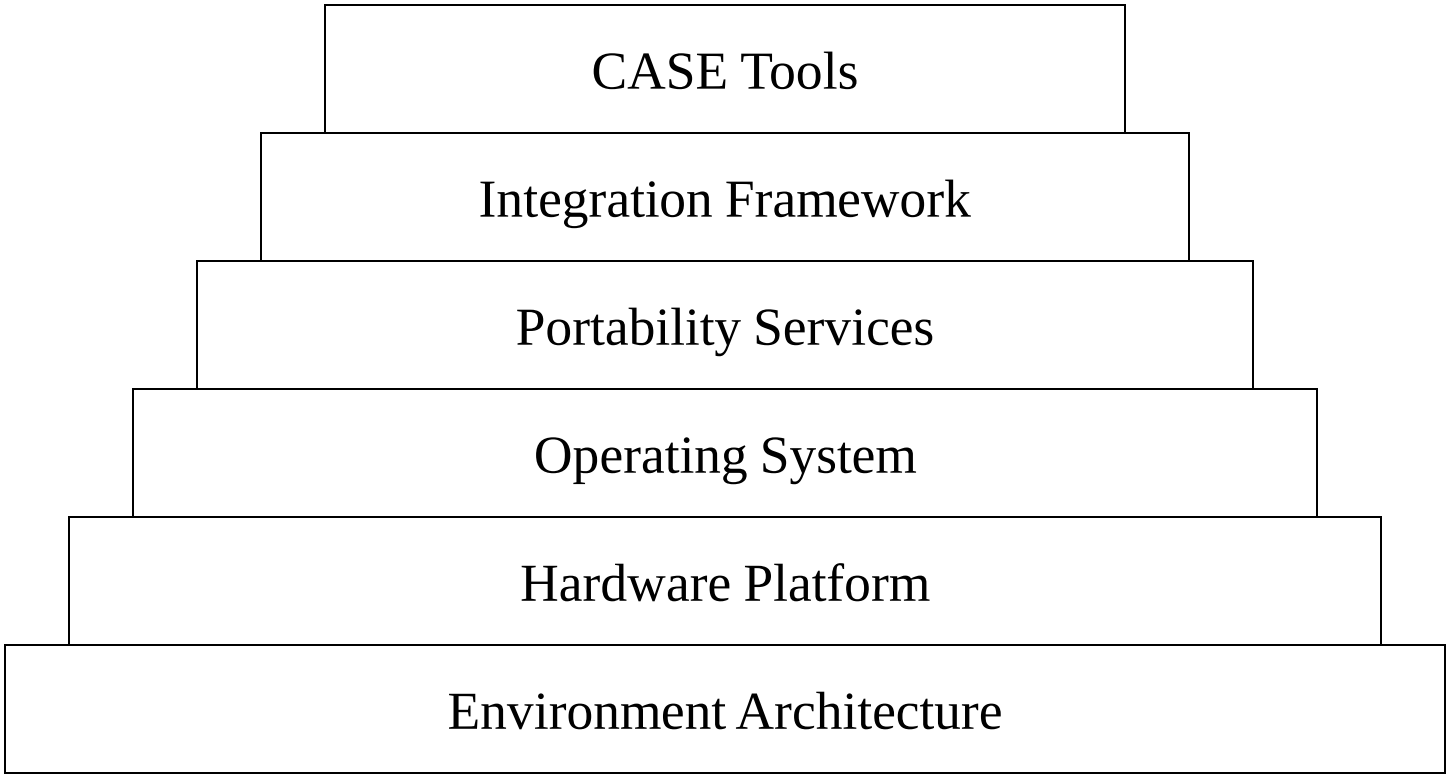
Lecture Objectives

- To understand the role of automation in the software engineering process
- To describe the different types of CASE tools
- To discuss the importance of integration among the different CASE tools

What is CASE?

- CAD/CAM - Computer-aided design & manufacturing
- Automated support for software engineering process
- Provides engineer with ability to automate manual activities and improve engineering insight and quality
- Can be single tool or complete environment

Building Blocks for CASE



Taxonomy of CASE Tools

- Business Systems Planning
  - Information Engineering Tools
  - Process Modeling and Management Tools
- Project Management
  - Project Planning Tools
  - Risk Analysis Tools
  - Project Management Tools
  - Requirements Tracing Tools
  - Metrics and Management Tools

Taxonomy of CASE Tools (Continued)

- Support Tools
  - Documentation Tools
  - System Software Tools
  - Quality Assurance Tools
  - Database Management Tools
  - Software Configuration Management Tools
- Analysis and Design Tools
  - PRO/SIM Tools
  - Interface Design and Development Tools
  - Prototyping Tools

Taxonomy of CASE Tools (Continued)

- Programming Tools
  - Integration and Testing Tools
  - Static Analysis Tools
  - Dynamic Analysis Tools
  - Test Management Tools
  - Client/Server Testing Tools
- Maintenance Tools
  - Reengineering Tools

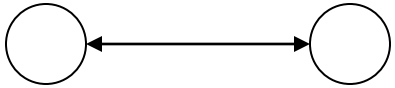
Integration Options

- Individual Tool (Point Solution)
- Data Exchange
- Tool Bridges & Partnerships
- Consortium & Standards
- Single Source
- IPSE

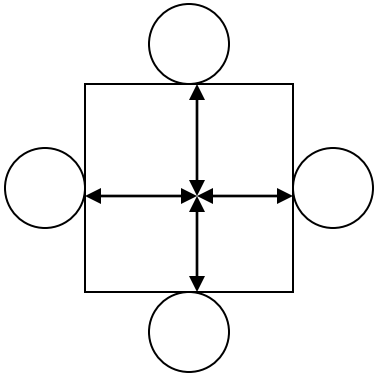
Integration Options Diagram

Point Solution

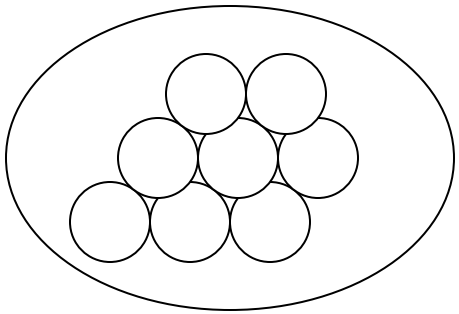
Data Exchange



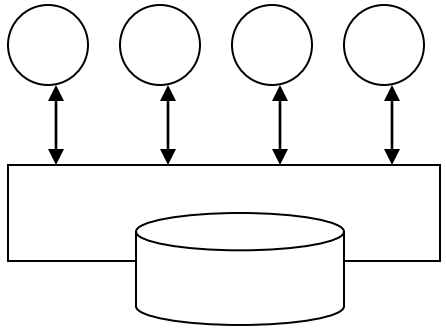
Tool Bridges & Partnerships



Consortium & Standards



Single Source



IPSE

Integrated CASE (I-CASE)

- Integration of a variety of tools and information that enables *closure* of communication among tools, between people and across the software process
- Combination of CASE tools in an environment where interface mechanisms are standardised

I-CASE Features

- All tools sharing SE information
- Change of one item tracked to other items
- Provide version control and configuration management
- Direct access to any tool
- Automated support for integration of tools & data into standard WBS

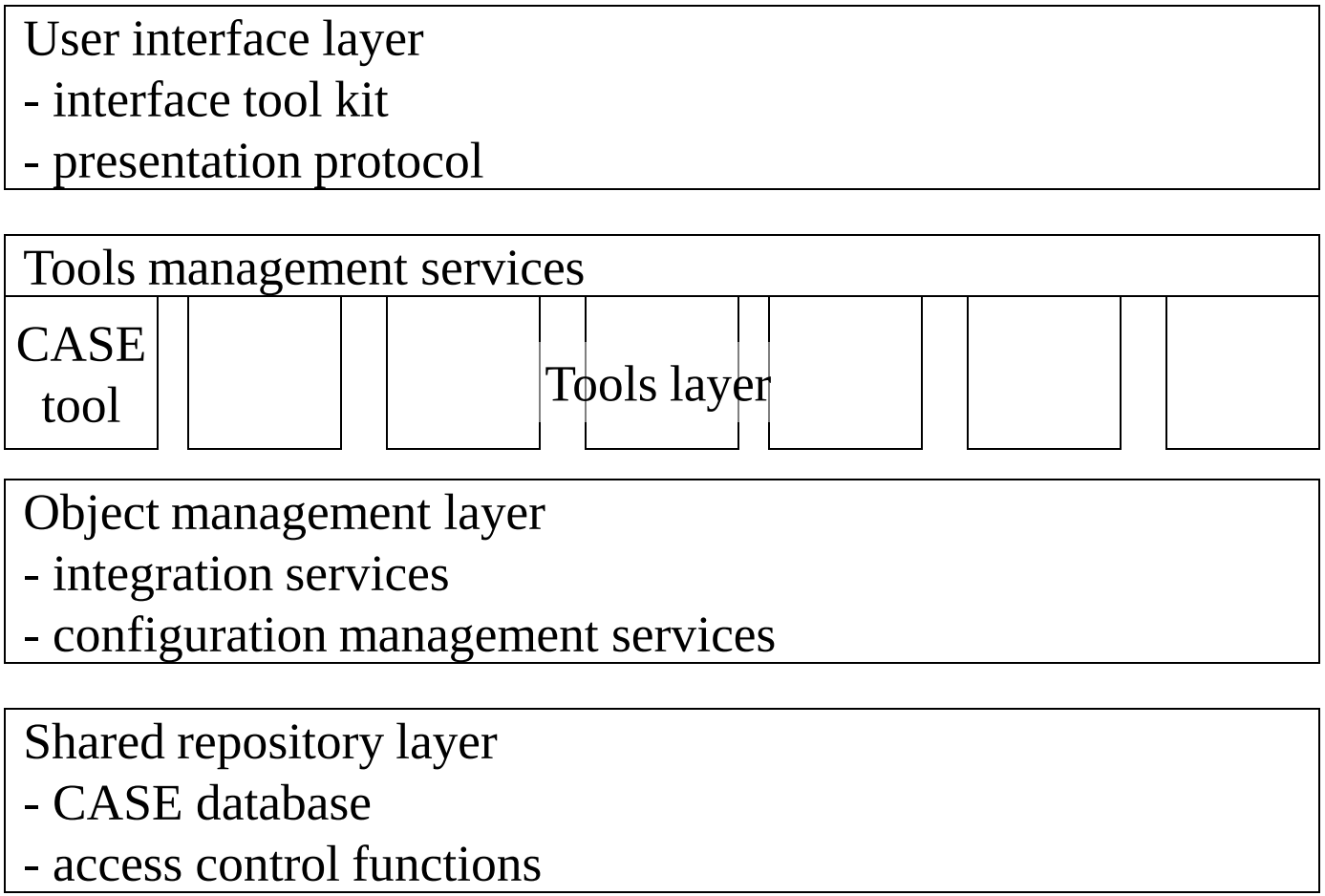
I-CASE Features
(Continued)

- Consistent look & feel for each tool
- Support communication among engineers
- Collect management & technical metrics

Benefits of I-CASE

- Smooth transfer of information from a tool to another and one SE step to the next
- Reduction in effort to perform umbrella activities such as SCM, SQA and document production
- increase in project control
- Improved coordination among staff members in a large software project

Integration Framework Diagram



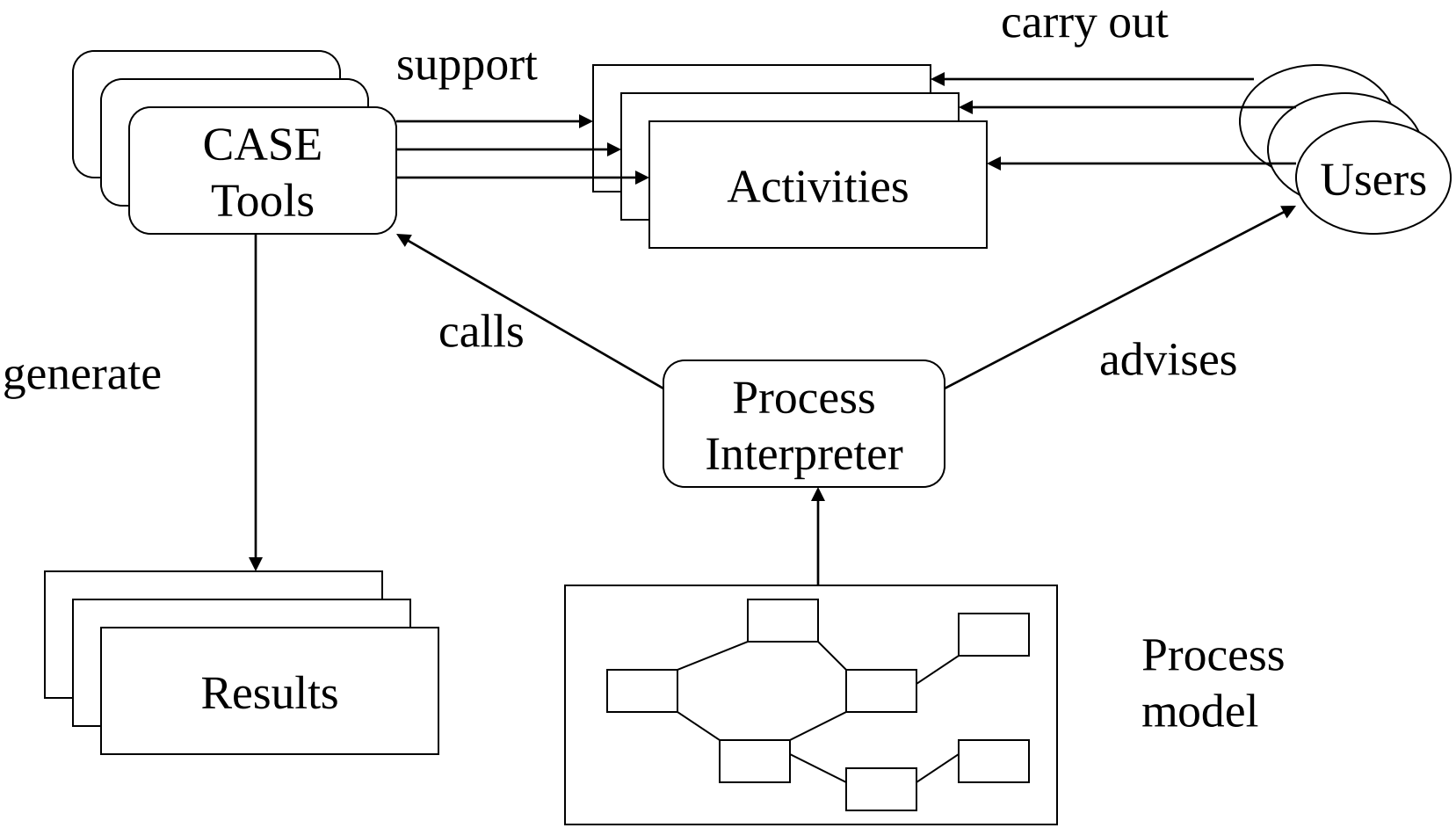
Integration Framework

- User interface layer
  - incorporates standardised interface toolkit with common presentation protocol
  - human-computer interface, display objects, guidelines for same look & feel
- Tools layer
  - tools management - control behaviour of tools
  - coordination of tasks, e.g. multitasking

Integration Framework (Continued)

- Object management layer
  - configuration management functions
  - integration services - standard modules that couple tools with repository
- Shared repository layer
  - CASE database
  - access control functions - enable object management layer interact with database

Process Integration



CASE Workbenches

- Set of tools which supports a particular phase of the software process e.g. design
- Advantage - tools can work together to provide more comprehensive support
- Common services can be implemented and called by all the tools
- Integration possible through shared files, shared repository, or shared data structures

Programming workbenches

- Language compiler
- Structured editor
- Linker
- Loader
- Cross-referencer
- Prettyprinter
- Static & Dynamic analyser
- Interactive debugger

Analysis and Design Workbenches

- Diagram editors
- Design analysis and checking tools
- Repository
- Repository query language
- Report definition and generation tools
- Forms definition
- Import/export facilities
- Code generators

Testing workbenches

- Test manager
- Test data generator
- Oracle - generates predicted results
- File comparator
- Report generator
- Dynamic analyser
- Simulator

Examples of CASE Tools

- With Class - object-oriented design and code generation
- Eiffelbench - object-oriented programming and debugging
- Oracle Designer/2000 - integrated CASE environment

With Class

- Design objects - identifying attributes and operations
- Specifying relationships
- Diagramming for various methodologies
- Code generation for various languages

Eiffelbench

- Based on Eiffel language (an object-oriented language)
- For development and debugging of program
- Consists of tools such as:
  - Project Tool
  - System Tool
  - Class Tool
  - Feature Tool
  - Object Tool