

## **COMPONENT BASED DEVELOPMENT (SE-O-02)**

### **AIMS:**

The development of modern, distributed enterprise-wide computing systems is a complex activity. Issues of scale and distribution have added to the general pressures of the 'software crisis' - the inability of the software industry to match expectations created by the rapid evolution of computing hardware. One approach targeted at these problems is component based development (CBD).

CBD has been the 'holy grail' of software engineering since the famous NATO (North Atlantic Treaty Organization) conference on software engineering coined the term in 1968. Technological development over the last decade has at last resulted in the emergence of a component market - but CBD also changes the nature of the software development process. This module provides the historical background in order to understand the issues and difficulties inherent to CBD which have held back its growth, while providing insights into the current situation through exposure to an industrial strength CBD process. It provides a practical introduction to the key modelling techniques used at different stages in a CBD project in order to identify and specify business components as 'independent units of deployment and replacement'. It illustrates how object-oriented principles underlay these techniques, and shows how they can be fitted together in an overall end-to-end development process.

The intention of the module is to provide students with a practical, integrated overview of CBD from requirements gathering through to the deployment of a component-based system. A large-scale case study, already part implemented, is used to take the student through all of the workflows of the CBD project life-cycle.

The module introduces relevant theory including, non-exhaustively, the concept of CBD; the history of CBD; objects versus components; business components and software architectures; design-by-contract; UML as a visual language for CBD; problem-space and solution-space modelling; the object constraint language for specifying pre- and post-conditions on operations and comparing development processes that aim to support CBD.

The module also lays a foundation in practical skills and understanding of CBD which will be of potentially immediate advantage to the student in her future profession. Use case driven techniques are explored in depth, specification modelling using a subset of UML together with relevant profiles are practiced, and implementation in the context of specific component execution environments are explored to give the student a balanced and in-depth base of theoretical and practical knowledge in this area.

### **LEARNING OUTCOMES:**

Upon successful completion of this module, the student will be able to:

- critically discuss and explore the key concepts in component based development, and its impact on business computing and on the software engineering discipline;
- understand the theoretical basics of CBD;
- apply appropriate techniques to derive detailed and rigorous component specifications from a set of user requirements;
- apply practical component development skills to implement and deploy significant business components in the context of a particular component execution environment.

### **SYLLABUS CONTENT:**

- Background.
- Component based software architecture.
- UML for component based development.
- CBD '96: requirements, specification, and provisioning workflow.
- Business objects.
- Component technologies (e.g., EJB, Microsoft technologies, CORBA).
- Combining components – case studies: accessing legacy code and wrappers; web services; components of the shelf (COTS).
- System integration issues.
- Tools.

**PREREQUISITES:** None

**RECOMMENDED ASSESSMENT:** Coursework and unseen paper