DEPARTMENT OF INFORMATION TECHNOLOGY

V R SIDDHARTHA ENGINEERING COLLEGE

VIJAYAWADA-7

VOLUME : 1 | ISSUE: 2 | JUNE 2026

Vision

To evolve as a centre of academic excellence and advanced research in Computer Science and Engineering discipline

Mission

To inculcate students with profound understanding of fundamentals related to discipline, attitudes, skills, and their application in solving real world problems, with an inclination towards societal issues and research

The Department of Information Technology was started UG Programme in "Information Technology" in the year 2000 with an initial intake of 60 student s and the intake is enhanced to 120 in the year 2007. In 2011, the department started a PG Program in "Computer Science and Technology (CST)" with an intake of 36 students. The department started a new The Department was established in 2000 with an initial intake of 60 students and has consistently expanded its academic offerings and capacity in response to the growing demand for quality technical education. The undergraduate intake was enhanced to 120 in 2007, further increased to 180 in 2022, and subsequently revised to 120 in 2025 in accordance with institutional and regulatory requirements. The Department has established a strong legacy of academic excellence, having earned NBA accreditation six times for its undergraduate programme, including accreditation cycles during 2008–09, 2014–17, and 2017–20, with the current Outcome-Based Education (OBE) accreditation granted for a period of six years (July 2022 to June 2028), reflecting its sustained commitment to quality education and continuous improvement.PG program in "Data Science" with an intake of 18 students in the year 2019. NBA Thrice accredited the B.Tech. Programme during 2008-09, 2014-17 and 2017-2020 and extended Accreditation till 30-6-2021 (OBE Tier I).

Dr.M.Suneetha
HOD : IT

EDITORIAL BOARD

Table of Contents

AI-Driven Software Engineering and Autonomous DevOps

Dr.V.Radhesyam
Associate Professor

SOFTWARE ENGINEERING

AI-Augmented Software Lifecycle and Intelligent DevOps

• AI-assisted development is transforming requirements engineering, code synthesis, testing, refactoring and software maintenance.

• Large language models can support repository-level reasoning when combined with code search, dependency graphs and project documentation.

• Autonomous DevOps combines continuous integration, automated testing, deployment policies and observability with intelligent decision support.

• Software agents can decompose engineering tasks, invoke development tools and validate outputs through iterative execution.

• Engineering governance requires traceability, secure coding controls, human review and reproducible evaluation of generated artifacts.

EXPERT-LEVEL TECHNICAL EXPLANATION

Modern AI-driven software engineering extends conventional lifecycle automation by introducing reasoning and tool-using systems into the development workflow. A practical architecture connects requirement repositories, source-control systems, issue trackers, code analysis engines, build pipelines and observability platforms through governed AI agents. Repository-aware retrieval provides contextual knowledge before code generation, while static analysis, unit testing and CI feedback act as verification mechanisms. The strongest research direction is not unrestricted code generation but closed-loop engineering in which AI proposes, executes, evaluates and revises changes under explicit quality and security constraints.

ARCHITECTURE

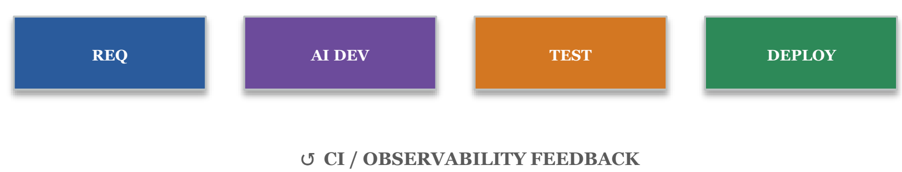

Figure 1: AI-augmented software engineering lifecycle

Dr.A.Srisaila
Assistant Professor(S.Grade)

ROBOTICS

Perception, Planning, Manipulation and Human–Robot Collaboration

• Embodied AI integrates perception, world modeling, task planning and low-level control to enable robots to act in dynamic physical environments.

• Vision-language-action models can connect natural-language goals with visual observations and executable manipulation policies.

• Robot perception combines RGB/depth sensing, LiDAR, tactile information and proprioception for robust state estimation.

• Motion planning and model-predictive control translate high-level task decisions into collision-aware trajectories and actuator commands.

• Human–robot collaboration requires intent recognition, uncertainty estimation, safety envelopes and explainable interaction policies.

EXPERT-LEVEL TECHNICAL EXPLANATION

An autonomous collaborative robot is best represented as a hierarchical intelligence stack. Sensor observations are fused into a spatial-temporal world model, after which a task planner decomposes the objective into executable skills. Learned policies generate candidate actions, while classical motion planning and control enforce geometric, dynamic and safety constraints. Feedback from the physical environment continuously updates the estimated state. This combination of foundation-model reasoning with model-based robotics is important because purely generative policies can be brittle when precise physical constraints and safety guarantees are required.

ROBOT INTELLIGENCE STACK

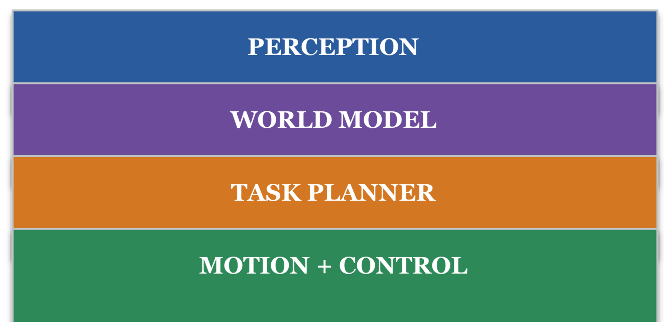

Figure 2: Hierarchical embodied-AI robotics architecture

SENSORS → STATE → PLAN → ACTION → FEEDBACK

Multimodal Generative AI and Foundation Models for Intelligent Applications

K.Madhavi
Assistant Professor

GENERATIVE AI

Vision–Language Models, RAG and Grounded Generation

• Foundation models provide general-purpose representations that can be adapted to text, image, audio, video and domain-specific enterprise data.

• Multimodal models align heterogeneous representations through shared embedding spaces and cross-attention mechanisms.

• Retrieval-augmented generation improves factual grounding by retrieving authoritative external evidence at inference time.

• Parameter-efficient adaptation reduces the computational cost of specializing large models for institutional or domain-specific tasks.

• Evaluation must address factuality, hallucination, robustness, provenance, latency, cost and safety rather than relying only on benchmark accuracy.

EXPERT-LEVEL TECHNICAL EXPLANATION

A contemporary generative-AI platform separates foundation-model inference from knowledge grounding and application orchestration. User inputs are transformed into modality-specific representations and routed to an appropriate model. Retrieval services obtain relevant documents or structured records, which are incorporated into the generation context. Guardrails and output validation constrain unsafe or unsupported responses, while application telemetry supports continuous evaluation. The architecture is increasingly shifting from isolated chat interfaces toward reusable multimodal foundation services that can be composed into domain-specific intelligent applications.

MULTIMODAL FUSION

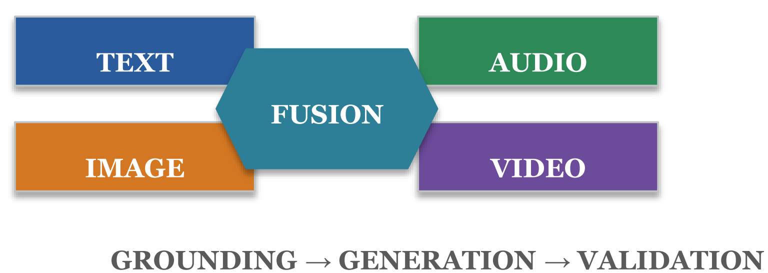

Figure 3: Multimodal foundation-model architecture

G.Geetha
Assistant Professor

QUANTUM COMPUTING

Hybrid Quantum–Classical Algorithms and Emerging Computing Paradigms

• Quantum computing uses qubits, unitary transformations and measurement to process information under quantum-mechanical principles.

• Variational quantum algorithms provide a practical hybrid model in which classical optimizers tune parameterized quantum circuits.

• Quantum machine learning explores quantum kernels, feature maps and variational circuits for classification and optimization.

• Quantum error mitigation and hardware-aware compilation are critical because near-term devices operate with noise and limited circuit depth.

• Meaningful evaluation requires strong classical baselines, realistic noise assumptions and analysis of data-encoding and execution overhead.

EXPERT-LEVEL TECHNICAL EXPLANATION

Near-term quantum applications are predominantly hybrid. Classical systems prepare data and optimize algorithm parameters, while a quantum processor executes parameterized circuits and returns measurement statistics. The classical optimizer uses these measurements to update the circuit, creating an iterative learning loop. Quantum kernels and variational circuits are promising research directions, but their practical advantage depends on circuit depth, qubit connectivity, noise, measurement cost and data-loading overhead. Consequently, rigorous quantum research should evaluate the complete computational workflow rather than treating the quantum circuit in isolation.

HYBRID QUANTUM WORKFLOW

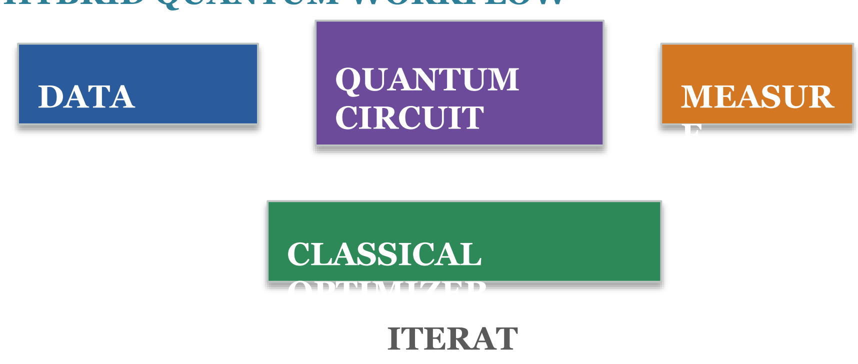

Figure 4: Hybrid quantum–classical computational

G.Jyosthna
Assistant Professor

SOFTWARE ENGINEERING

Microservices, AI Agents, Observability and Autonomous Engineering

• Intelligent application engineering combines microservices, event-driven architectures and AI agents to build adaptive software systems.

• Service meshes and observability platforms provide distributed tracing, metrics and policy enforcement across heterogeneous application components.

• AI agents can monitor operational signals, diagnose anomalies and recommend or execute controlled remediation actions.

• Architecture decisions increasingly include model routing, inference cost, latency, data governance and AI-specific failure modes.

• Digital engineering practices benefit from automated architecture checks, dependency analysis and continuous reliability assessment.

INTELLIGENT APPLICATION FABRIC

Figure 5: AI-enabled cloud-native software architecture

EXPERT-LEVEL TECHNICAL EXPLANATION

The architecture of intelligent applications extends cloud-native engineering with an AI control layer. User services and data services operate through APIs and event streams, while an agentic layer consumes telemetry and contextual knowledge to support diagnosis and optimization. Observability becomes a key feedback mechanism: traces, logs and metrics are converted into operational evidence for AI-assisted reasoning. Governance services enforce access, model policies and auditability. The resulting architecture supports adaptive systems while retaining the modularity and reliability principles of conventional software engineering.

P.MadhaviLatha
Assistant Professor

GENERATIVE AI / AGENTIC AI

Agentic Workflows, Tool Use, Memory and Enterprise Automation

• Agentic generative AI transforms foundation models into goal-oriented systems capable of planning, tool invocation and iterative execution.

• Task decomposition enables specialized agents to handle research, data analysis, document processing, coding and verification.

• Memory mechanisms maintain working context, while retrieval systems provide persistent domain knowledge without overloading the model context.

• Verifier and critic components assess intermediate results and trigger replanning when evidence or constraints are not satisfied.

• Enterprise adoption requires tool permissions, audit trails, human approval gates and measurable agent evaluation frameworks.

EXPERT-LEVEL TECHNICAL EXPLANATION

An agentic automation architecture places an orchestration layer above foundation models and enterprise tools. The orchestrator decomposes a goal into subtasks, selects specialized agents and maintains execution state. Agents access approved tools through a controlled gateway, while retrieval services provide authoritative organizational knowledge. Verification components evaluate evidence, consistency and completion before the workflow proceeds. This closed-loop structure is more robust than a single prompt-response system because planning, execution and validation are explicitly separated, enabling measurable reliability and governance.

AGENTIC AUTOMATION GRAPH

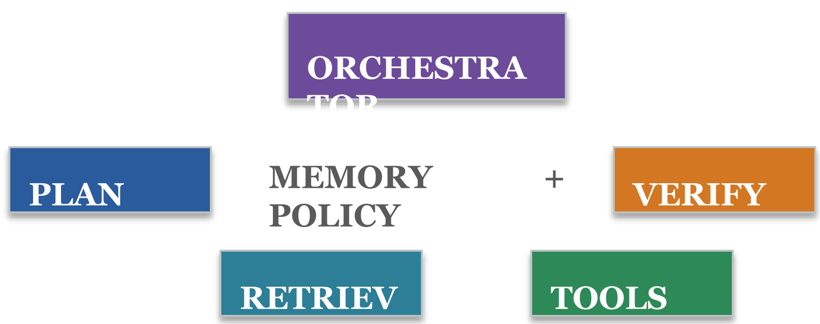

Figure 6: Goal-driven generative-AI agent architecture