Innopolis University - Autumn 2024

Architecture of Software Systems

Assignment 6: Attribute-Driven Design

Problem and Context

The task is to design an Internet-based collaboration system that incorporates:

- Speech communication
- Video conferencing
- Email
- File sharing
- A shared whiteboard that collaborators (such as software engineers) on a virtual team can use for brainstorming and sketching their ideas (for instance, drawing UML diagrams for software design).

The system should allow collaborators to use a variety of input devices, such as wireless pen tablets or touch screens, and it should be capable of sending output to different devices such as display stations or smartphones. It must be highly responsive, particularly for sketching or drawing, speech and video conferencing. Additionally, the system should recover from unstable network connections, track progress, and troubleshoot any failures at runtime.

---

1. Business Goals and Engineering Objectives

Business Goals	**Engineering Objectives**
Enable seamless collaboration for teams	Provide low-latency and high-performance communication systems
Support cross-platform usability	Ensure compatibility with multiple devices and platforms
Guarantee real-time interactions	Minimize delays in speech, video, and whiteboard collaborations
Allow secure data and communication	Ensure data privacy through strong encryption methods.

2. Quality Attributes

Quality Attribute	**Description**
Performance	The system should maintain high responsiveness, particularly during real-time activities such as sketching and video conferencing.

Quality Attribute	Description
Availability	The system should be capable of recovering from network disruptions and remain operational under unstable conditions.
Usability	The interface should be intuitive and user-friendly, particularly when interacting with various input devices.
Security	All data transmitted through the system should be encrypted to guarantee privacy and security.
Scalability	The system should support numerous concurrent users without noticeable performance degradation.

3. Quality Attribute Scenarios

Quality Attribute	Scenario
Performance	The system should facilitate real-time video conferencing with latency under 200 milliseconds.
Availability	The system should restore functionality within 5 seconds of a network disconnection, with no loss of data.
Usability	Users should be able to initiate a collaboration session (including video and whiteboard) in less than 10 seconds.
Security	All communications between users must be encrypted end-to-end, protecting sensitive project information.
Scalability	The system should support up to 500 simultaneous users with real-time video and whiteboard interactions.

4. Constraints

Business Constraints

Constraint	Description
Budget limitations	The project budget should not exceed $500,000.
Development timeframe	The system must be ready for launch within one year.
Compliance with regulations	The system must comply with data protection regulations like GDPR.

Technical Constraints

Constraint	Description

Constraint	Description
Hardware compatibility	The system must run on devices with minimal technical specifications.
Technology stack	The technology stack should prioritize open-source solutions.
Internet bandwidth limitations	The system should function effectively even in areas with limited bandwidth.
Cross-device compatibility	The system must work seamlessly across tablets, smartphones, and PCs.
Resource management	Optimize CPU and memory usage to avoid system crashes on low-end devices.
Open standards	The system should use open protocols for communication and file sharing.

5. Functionalities

1. Real-time video conferencing with multiple users
2. Speech communication between team members
3. Shared whiteboard for collaborative drawing (e.g., UML diagrams)
4. File sharing with team members in real-time
5. Email integration for sending project updates
6. Text-based chat for non-verbal communication
7. Ability to record and store video/audio meetings
8. Document version control to manage file history
9. Notification system for team activity and task updates
10. Integration with project management tools (e.g., Jira, Trello)

6. Architectural Drivers

Architectural Driver	Impact on Design
Real-time interaction requirements	The design should prioritize fast communication with minimal delays.
Security and privacy concerns	All data transfers between users and the system should be secured.
Multi-device compatibility	Design should support various input/output devices such as tablets and smartphones.
Scalability needs	The system should be designed to handle many users at the same time without performance issues.
Network reliability	Mechanisms for fault tolerance and recovery should be implemented to address network instability.

7. ADD Process Iterations

Iteration 1: Initial Module View

Step 1: Review Inputs

- **Drivers:** Real-time collaboration, multi-device compatibility.
- **Constraints:** Budget, development timeframe.
- **Quality Attributes:** Performance, usability.

Step 2: Establish Iteration Goal by Selecting Drivers

- **Goal:** Define the fundamental structure to handle communication and file sharing.

Step 3: Choose Elements to Refine

- **Elements:** Communication module, Storage module, UI module.

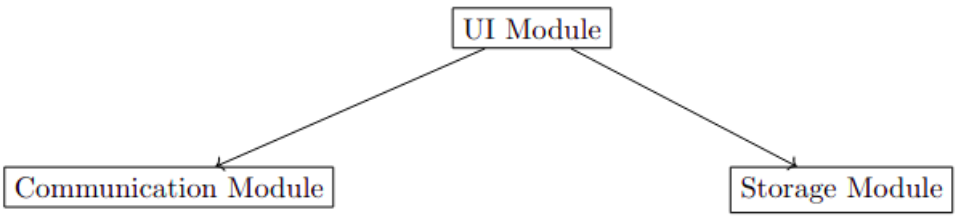
Step 4: Choose Design Concepts to Satisfy Drivers

- **Concept:** Modular architecture to isolate communication, data storage, and user interface interactions.

Step 5: Instantiate Elements and Define Interfaces

- **Communication Handler:** Manages communication between users in real time.
- **Data Manager:** Handles file uploads and downloads.
- **UI Controller:** Manages user interactions.

Step 6: Sketch Views and Record Design Decisions



Step 7: Review Iteration Goal and Achievement of Purpose

- **Achieved:** Base structure for real-time communication and file sharing.

Iteration 2: Adding Real-time Communication and Whiteboard

Step 1: Review Inputs

- **Drivers:** Real-time interaction, multi-device compatibility.
- **Quality Attributes:** Performance, usability.

Step 2: Establish Iteration Goal by Selecting Drivers

- **Goal:** Implement real-time video and collaborative whiteboard functionality.

Step 3: Choose Elements to Refine

- **Elements:** Communication Module.

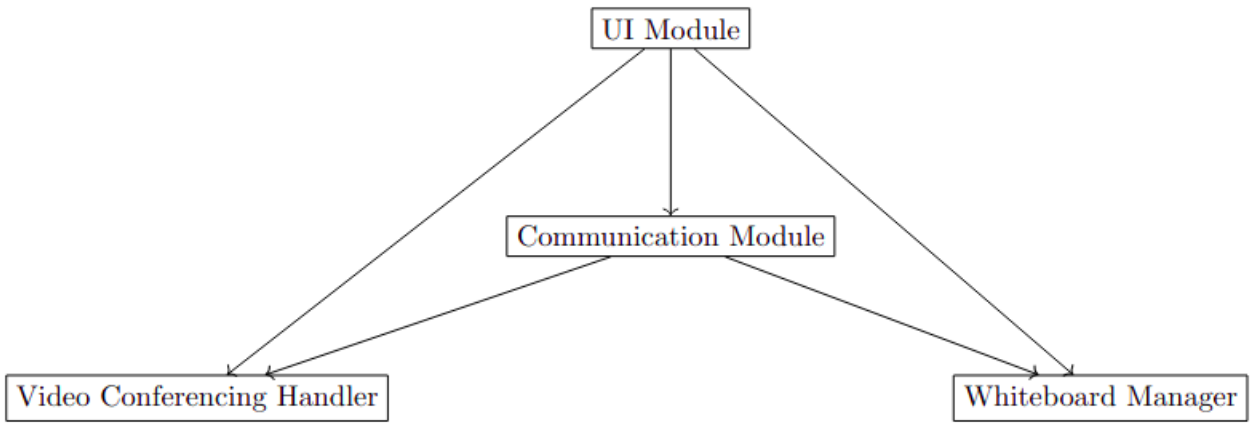
Step 4: Choose Design Concepts to Satisfy Drivers

- **Concept:** Integrate WebRTC for video conferencing and real-time whiteboard.

Step 5: Instantiate Elements and Define Interfaces

- **Video Conferencing Handler:** Manages video streams.
- **Whiteboard Manager:** Real-time drawing canvas shared across users.

Step 6: Sketch Views and Record Design Decisions



Step 7: Review Iteration Goal and Achievement of Purpose

- **Achieved:** Video conferencing and whiteboard functionalities integrated.

Iteration 3: Security Enhancements

Step 1: Review Inputs

- **Drivers:** Data privacy, GDPR compliance.
- **Constraints:** Legal regulations, performance.

Step 2: Establish Iteration Goal by Selecting Drivers

- **Goal:** Secure communication and storage.

Step 3: Choose Elements to Refine

- **Elements:** Communication Module, Storage Module.

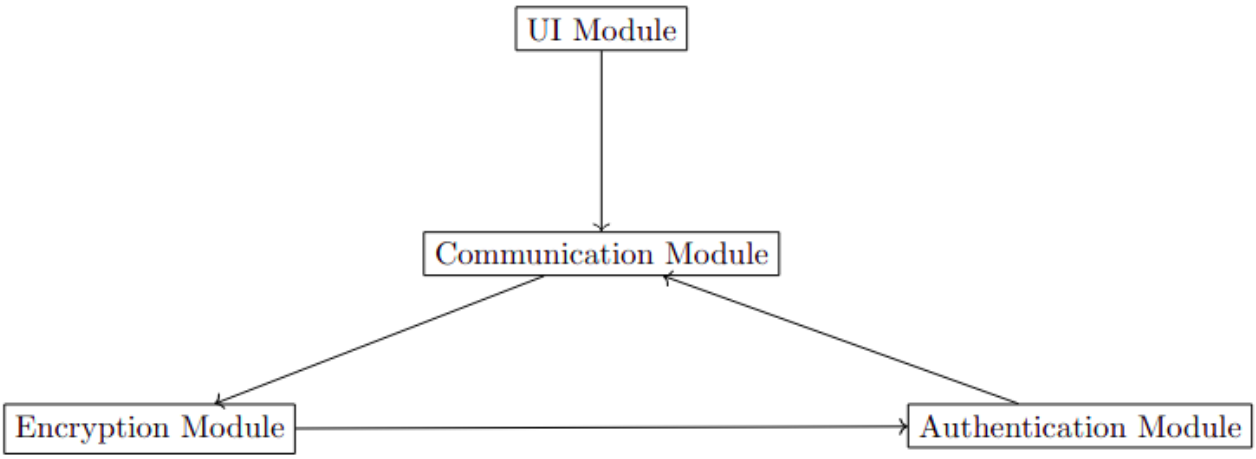
Step 4: Choose Design Concepts to Satisfy Drivers

- **Concept:** Implement end-to-end encryption, OAuth 2.0 for authentication.

Step 5: Instantiate Elements and Define Interfaces

- **Encryption Module:** Provides secure communication for data-in-transit.
- **Authentication Module:** Manages user access using OAuth 2.0.

Step 6: Sketch Views and Record Design Decisions



Step 7: Review Iteration Goal and Achievement of Purpose

- **Achieved:** Communication secured with encryption, access control implemented.

Iteration 4: Scalability Considerations

Step 1: Review Inputs

- **Drivers:** Scalability, high performance under load.
- **Constraints:** Infrastructure cost, system architecture.

Step 2: Establish Iteration Goal by Selecting Drivers

- **Goal:** Ensure system scalability to support 500+ concurrent users.

Step 3: Choose Elements to Refine

- **Elements:** Communication Module, Storage Module.

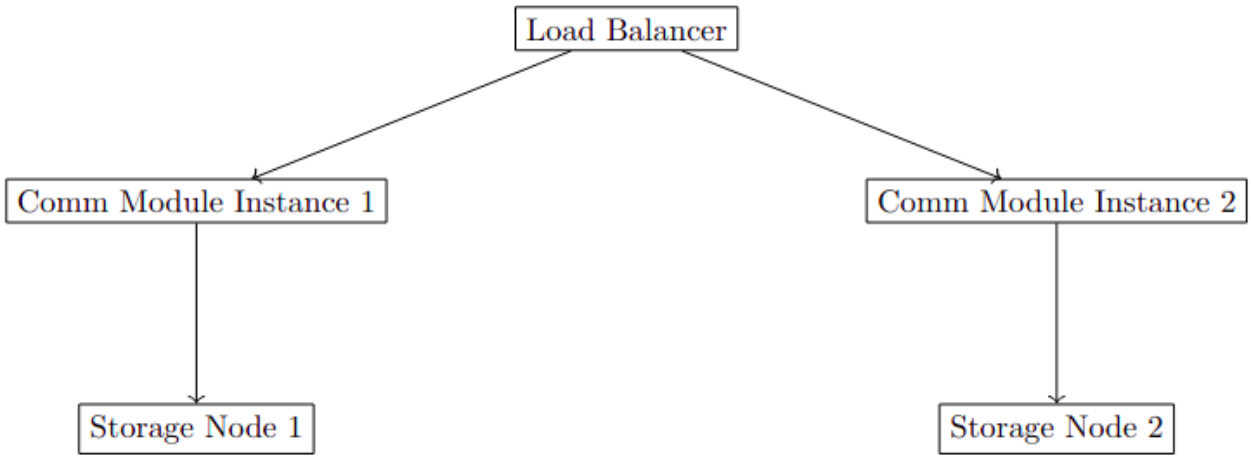
Step 4: Choose Design Concepts to Satisfy Drivers

- **Concept:** Use load balancing, distributed storage, and stateless services for scalability.

Step 5: Instantiate Elements and Define Interfaces

- **Load Balancer:** Distributes communication across multiple servers.
- **Distributed Storage Nodes:** Ensures scalability for file sharing.

Step 6: Sketch Views and Record Design Decisions



Step 7: Review Iteration Goal and Achievement of Purpose

- **Achieved:** System scalability addressed using load balancing and distributed storage.

Iteration 5: Network Fault Tolerance

Step 1: Review Inputs

- **Drivers:** System availability, network reliability.
- **Constraints:** Minimize downtime during network failures.

Step 2: Establish Iteration Goal by Selecting Drivers

- **Goal:** Improve system fault tolerance to ensure minimal service interruptions.

Step 3: Choose Elements to Refine

- **Elements:** Communication Module, Storage Module.

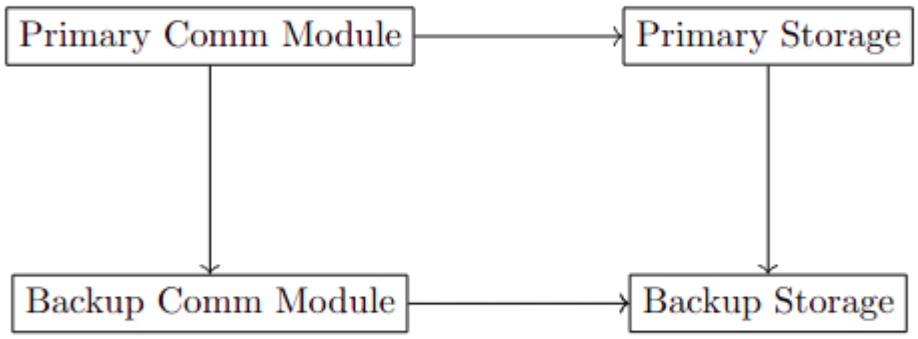
Step 4: Choose Design Concepts to Satisfy Drivers

- **Concept:** Implement server redundancy and failover protocols.

Step 5: Instantiate Elements and Define Interfaces

- **Failover Protocol:** Switches to a backup server if the primary one fails.
- **Backup Storage Nodes:** Replicate data across servers for redundancy.

Step 6: Sketch Views and Record Design Decisions



Step 7: Review Iteration Goal and Achievement of Purpose

- **Achieved:** The system can handle server failures and continue functioning.

---

8. ATAM Analysis

Attribute	Decision Impact
Performance	Use of a low-latency communication protocol ensures fast video conferencing and drawing.
Availability	Backup systems should be in place to help the system recover if the network goes down
Usability	The design is user-friendly, with simple interfaces for multiple device inputs.
Security	Strong encryption protocols ensure that data privacy is maintained throughout the system.
Scalability	Distributed architecture allows for horizontal scaling to support a large number of users.