Software Quality Engineering: UML and System Bottlenecks

Comprehensive Review and Preparation

Adam Bouafia

11 June 2024

University of L'Aquila

Supervised by: Prof. Vittorio Cortellessa

Table of contents

1. Introduction to UML
2. UML Diagrams
3. Models and Meta-Models
4. Bottlenecks and Performance
5. Reliability and Availability
6. Suggested Questions and Answers
7. Project Documentation Review
8. Conclusion

Introduction to UML

Unified Modeling Language (UML)

- Standardized modeling language for software systems
- Specifies, visualizes, constructs, and documents artifacts
- Key for object-oriented software-intensive systems

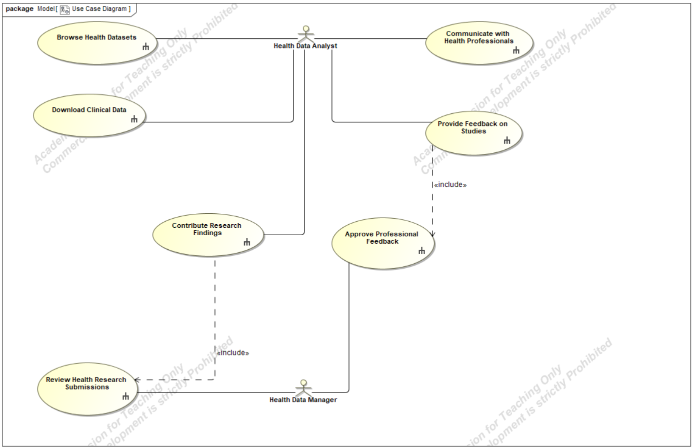
Why Use UML?

- Facilitates communication among stakeholders
- Helps in visualizing and analyzing system architecture
- Aids in documenting and maintaining system design
- Provides a standard way to understand system behavior

UML Diagrams

Use Case Diagram

- Represents system functionality through actors and use cases
- Level: M1
- Components: Actors, Use Cases, Associations, Include/Extend Relationships

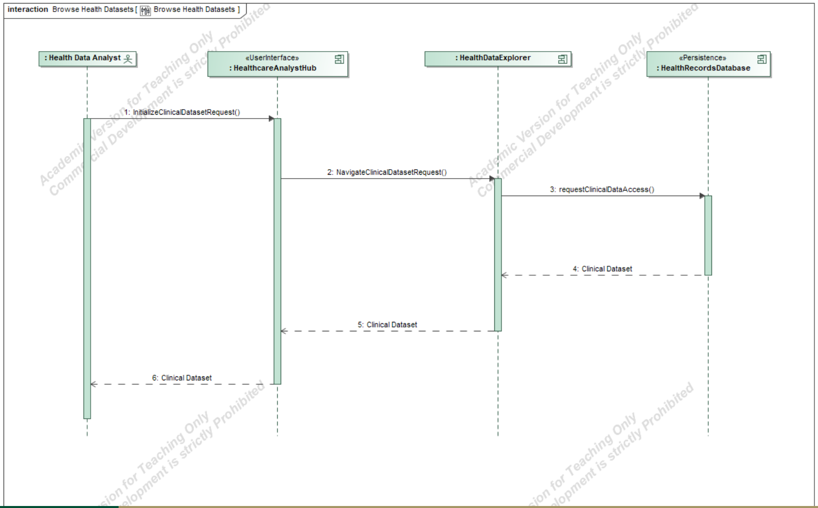


Components of Use Case Diagram

- **Actors**: Entities that interact with the system (users or other systems).
- **Use Cases**: Specific functionalities or services provided by the system.
- **Associations**: Lines connecting actors to use cases, indicating interaction.
- **Include/Extend Relationships**: Define how use cases include or extend other use cases.

Sequence Diagram

- Illustrates object interactions in a time sequence
- Level: M1
- Components: Lifelines, Messages, Activation Bars, Fragments

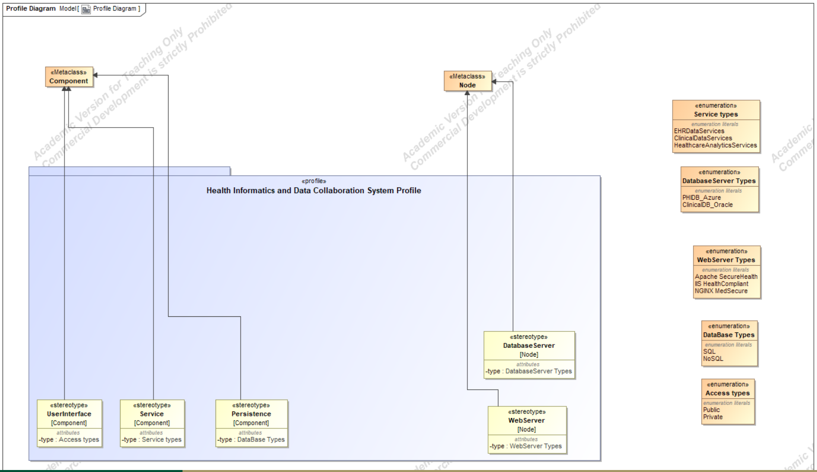


Components of Sequence Diagram

- **Lifelines**: Represent objects or participants in the interaction.
- **Messages**: Arrows representing communication between lifelines.
- **Activation Bars**: Indicate the period an object is active during the interaction.
- **Fragments**: Represent conditional or looped interactions.

Profile Diagram

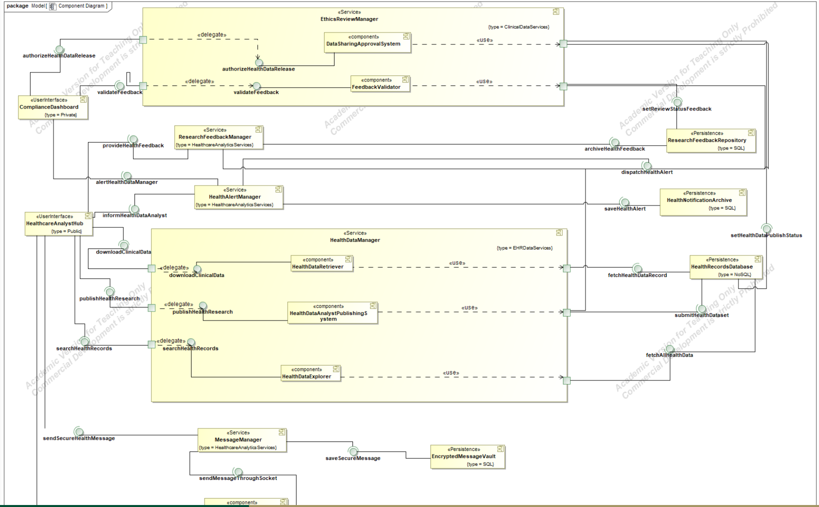
- Extends UML meta-model with custom stereotypes
- Level: M2
- Components: Stereotypes, Tagged Values, Constraints, Classes, Attributes, Operations, Relationships, Nodes, Components



Components of Profile Diagram

- **Stereotypes**: Custom elements that extend UML metamodel.
- **Tagged Values**: Additional properties for stereotypes.
- **Constraints**: Rules applied to stereotypes.
- **Classes, Attributes, Operations, Relationships**: Standard UML elements extended by stereotypes.
- **Nodes, Components**: Physical elements that can be extended.

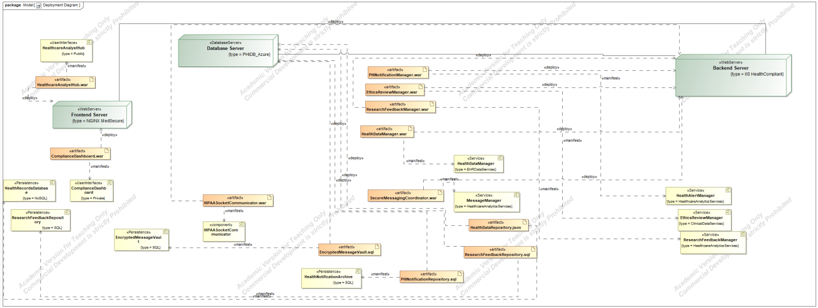
- Visualizes organization and relationships among components
- Level: M1
- Components: Components, Interfaces, Ports, Relationships



Components of Component Diagram

- **Components**: Modular parts of a system with well-defined interfaces.
- **Interfaces**: Points of interaction between components.
- **Ports**: Specific interaction points on components.
- **Relationships**: Dependencies and associations between components.

- Models physical deployment of artifacts on nodes
- Level: M1
- Components: Nodes, Artifacts, Communication Paths



Components of Deployment Diagram

- **Nodes**: Physical devices or execution environments.
- **Artifacts**: Software components deployed on nodes.
- **Communication Paths**: Paths showing how nodes communicate with each other.

Models and Meta-Models

Model, Meta-Model, Meta-Meta-Model

- Model (M1): System being designed (e.g., Use Case Diagram, Sequence Diagram, Component Diagram, Deployment Diagram)
- Meta-Model (M2): Language for expressing a model (UML meta-model, e.g., Profile Diagram)
- Meta-Meta-Model (M3): Language for expressing a meta-model (MOF - Meta-Object Facility)

Bottlenecks and Performance

Bottlenecks in Software Systems

- Definition: A point where the performance is limited due to insufficient capacity or resources
- Identified bottlenecks: High disk demand, network latency, CPU usage
- Resolutions: Increase server capacity, optimize network resources, enhance CPU performance

Bottlenecks and Resolutions

6. Bottleneck Analysis and Resolution Post-Simulation:

The execution of the simulation has led to the identification of three primary bottlenecks within the system. They are as follows:

1. **Contribute Resource Finding Bottleneck:**
   The operation "insertResource" presented a significant delay, manifesting as the first bottleneck. This was attributed to an elevated Disk demand value of 13, which when computed using our service time equation, resulted in a prolonged processing time of 0.65 seconds for each individual request. The optimal strategy employed to mitigate this issue was to increase the number of servers at the HealthAnalyticsServer_Disk node to three, which effectively alleviated the congestion.
2. **Provide Feedback on Studies Bottleneck:**
   Similarly, the "archiveHealthNotification" operation emerged as a second bottleneck, stemming from an identical complication as the first, with high disk demand causing delays. Implementing the same resolution as for the first bottleneck, which involved the enhancement of server capacity at the respective node, proved successful in resolving this bottleneck.
3. **Communicate with Health Professionals Bottleneck:**
   The third bottleneck was observed within the "Communicate with Health Professionals" functionality. However, upon a detailed review, it was decided to categorize this as a non-critical issue and thus, it was not addressed in the same manner as the others. The rationale behind this decision is rooted in the system design, particularly the functionality for offline users. Considering that only 25% of users are expected to be online at any given time, it was determined that the notification delivery to offline Health Professionals—which only occurs upon their return online—need not be factored into the immediate demand. This led to a strategic decision to prioritize system resources and focus exclusively on the online user interactions. Consequently, this targeted approach resolved the bottleneck without necessitating changes to the system's infrastructure for the offline component.

Worst-Case Analysis

- Evaluates the maximum load and stress the system can handle
- Determines the upper limits of performance and resource usage
- Helps in planning for peak loads and ensuring system stability
- Example: Simulating the maximum number of simultaneous users

Worst-Case Analysis Calculation

- **Browse Health Dataset:**
- CPU: 9 (resource demand) x 0.01 (maximum service time) = 0.09 sec
- Disk: 4 (resource demand) x 0.05 (maximum service time) = 0.20 sec
- Network: 12 (resource demand) x 0.0125 (maximum service time) = 0.15 sec
- **Total: 0.44 sec**
- Number of seconds needed to complete 10 requests: 0.44 x 10 = 4.4 sec

Worst-Case Analysis Calculation

4. Worst-Case Analysis :

When conducting the worst-case standalone analysis, we chose the following maximum service times (in seconds):

CPU	DISK	NET
0.01	0.05	0.0125

4.1 Browse Health Dataset :

The Total Resource Demanded in the "Browse Health Dataset" as it displayed in the Execution Graph is:

CPU	DISK	NET
9	4	12

- ✓ **Response Time Requirement**: The system should respond within 5 seconds under a maximum workload of 10 requests/second.

Worst-Case Analysis:

- CPU: 9 x 0.01 = 0.9 sec
- DISK: 4 x 0.05 = 0.2 sec
- NET: 12 x 0.0125 = 0.35 sec

- ✓ TOTAL (sum): 0.44 sec
- ✓ Number of seconds needed to complete 10 requests: 0.44 x 10 = 4.4 sec

- ❖ In the Worst-Case scenario, the system meets the response time requirement

Queueing Network Model

- Uses mathematical models to represent the flow of data and resources
- Helps in analyzing the system’s behavior under different loads
- Identifies potential delays and points of congestion
- Example: Modeling the request-response time for database queries

What-If Analysis

- Simulates different scenarios to predict system behavior
- Helps in decision-making and planning for future upgrades
- Analyzes the impact of changes in load, resources, and configurations
- Example: Testing the effect of doubling the number of users on system performance

3. **What if we quadruple the server count?**

If the current server capacity is increased fourfold, what impact would this have on the system's overall throughput and response time, particularly during peak usage periods?

✓ **Solution**:

Based on the JMT simulation results, it appears that increasing the number of EHRDatabase_CPU servers beyond two does not significantly improve the system's response time.

Therefore, for the current workload and system configuration, two servers are sufficient to handle the tasks efficiently. Adding more servers beyond this does not yield a proportional benefit.

For System Response Time: There's a slight decrease in the response time when the number of servers increases from one to two. Beyond two servers, the response time does not significantly improve, which indicates that two servers are currently optimal for our system in terms of response time.

For System Power: The system's power usage increases as more servers are added, suggesting that having more than two servers doesn't contribute to efficiency and may lead to unnecessary energy consumption without tangible performance benefits.

Reliability and Availability

Reliability and Availability

- Reliability: The probability that a system will perform without failure over a specific period
- Availability: The proportion of time a system is operational and accessible when required
- Importance: Critical for systems where downtime can lead to significant losses or harm
- Strategies: Redundancy, fault tolerance, regular maintenance, robust testing

Suggested Questions and Answers

- **What is UML and why is it important in software engineering?**
  Answer: UML provides a standardized way to visualize system design, crucial for communication and documentation in software engineering.
- **Explain the difference between a model, meta-model, and meta-meta-model in UML.**
  Answer: Models represent the actual system (M1), meta-models define the language for expressing models (M2), and meta-meta-models define the language for meta-models (M3).

Suggested Questions and Answers

- **Describe the purpose and key components of a profile diagram.**
  Answer: Profile diagrams extend UML meta-models with custom stereotypes, nodes, and components.
- **What are the common bottlenecks in software systems and how can they be resolved?**
  Answer: Bottlenecks can be due to CPU, disk, and network resource constraints; resolved by optimizing resources and increasing server capacity.

Suggested Questions and Answers

- **What strategies can be employed to ensure system reliability and availability?**
  Answer: Strategies for reliability and availability include redundancy, fault tolerance, regular maintenance, and robust testing.
- **What are the main objectives of the HIDC system?**
  Answer: The primary objectives include fostering collaboration, providing a secure and compliant platform, and integrating health data analytics tools.
- **How does the HIDC system ensure compliance with regulations?**
  Answer: The system adheres to GDPR and HIPAA regulations through secure data handling and compliance dashboards.

Suggested Questions and Answers

- **What is the role of Health Data Managers in the HIDC system?**
  Answer: Health Data Managers manage datasets, approve access, and handle feedback, ensuring data integrity and regulatory compliance.
- **How does the HIDC system facilitate collaboration among health data analysts?**
  Answer: The system offers features like secure messaging, data sharing, and compliance tools to support collaborative research and analysis.

Project Documentation Review

HIDC System Overview

- Centralized platform for health data analysis and collaboration
- Integration with SoBigData++ for enhanced data analytics
- Key features: secure communication, compliance dashboards, and data publication systems

Identified Bottlenecks and Resolutions

- Insert Resource: High disk demand resolved by increasing server capacity.
- Archive Health Notification: Similar resolution as the first bottleneck.
- Communicate with Health Professionals: Issues resolved by optimizing network resources.

Conclusion

Summary

- Reviewed UML diagrams and their purposes
- Explained models, meta-models, and meta-meta-models
- Discussed bottlenecks and performance optimization
- Reviewed project documentation for HIDC system

Questions?