Technical University of Moldova
Inginerical department S.A.

Report

№: 9

AMOO

Subject: Deployment Diagrams. Data Flow Diagrams. OntoUML.

Author: Terman Emil FAF161
Prof: M. Gavrilita

Chisinau 2018

Theory

Deployment diagrams

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

OntoUML

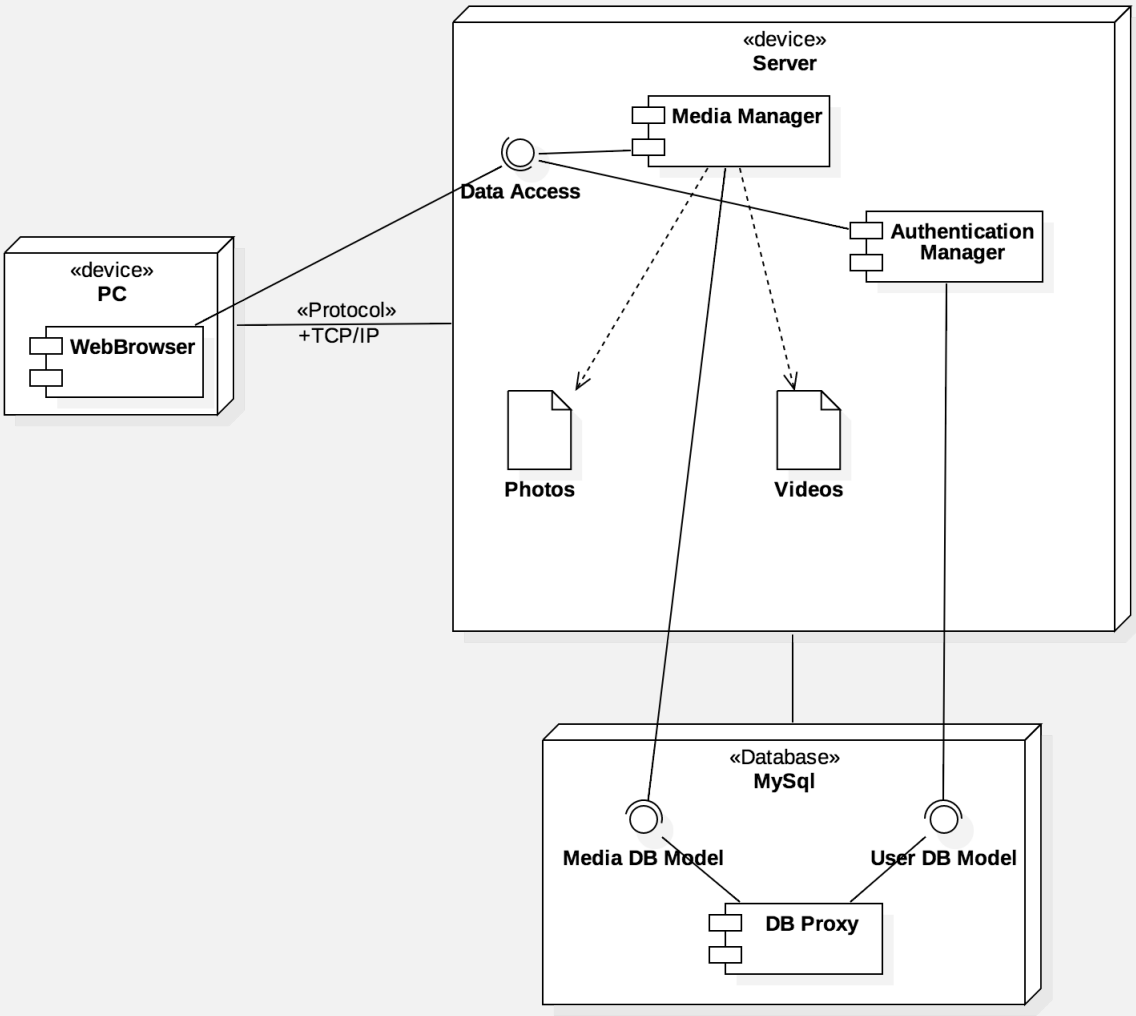
OntoUML is a ontologically well-founded language for Ontology-driven Conceptual Modeling. OntoUML is built as a UML extension based on the Unified Foundational Ontology (UFO). OntoUML has been adopted by many academic, corporate and governmental institutions worldwide for the development of conceptual models in a variety of domains. It has also been considered as a candidate for addressing the OMG SIMF (Semantic Information Model Federation) Request for Proposal, as is explicitly recognized as the foundations for the Data Modeling Guide (DMG) For An Enterprise Logical Data Model (ELDM) initiative. Finally, some of the foundational theories underlying OntoUML have also influenced other popular conceptual modeling languages such as ORM 2.0.

Deployment diagrams

Figure 1 represents a general Deployment Diagram conerning the *Media Manager*. The *Photos* and the *Videos* are saved on the server and only the paths are stored in the Database. The Database has Proxy which comunicates with our program. The client accesses the web app through a Web Browser. The Browser then communicates through a *Data Access* ”interface”, which represents the actual requests with its headers.

Conclusion

With **Deployment diagrams** it’s easier to create an even more general view on the project. In this laboratory work, I have learned what hardware oriented diagrams there are. Despite the previous UNL schemes, in this one, it is clearly shown that the User uses his **Web Browser** to interact with the server. Another important thing is that this type of diagram visualizes where the *Photos* and *Videos* are located, which wasn’t quite possible in the previous diagrams.



Img 1: Deployment Diagram