



K. J. Somaiya College of Engineering, Mumbai-77

Batch: C2 Roll No.: 16010121110

Experiment / assignment / tutorial No

TITLE Study of Architectural Description Language.

AIM: To understand need of architectural descriptive language .

Expected OUTCOME of Experiment:

Need of ADL and use of the same in representing software architecture

CO 1. Design the architecture of software systems in various architectural styles.

Books/ Journals/ Websites referred:

- <https://web.archive.org/web/20150909170715/http://www.mrtc.mdh.se/han/FoP/lan/ass2-bjornander.pdf> accessed on July 5, 2024
- <https://www.sciencedirect.com/science/article/abs/pii/B9780123742872500045> Accessed on July 5, 2024
- <https://www.opengroup.org/archimate-forum/archimate-overview> accessed on July 5, 2024
- <https://www.bing.com/videos/riverview/relatedvideo?q=learning+archimate+software+youtube&mid=BF3F92F1CA1BEF8CD91BBF3F92F1CA1BEF8CD91B&FORM=VIRE> accessed on July 5, 2024
- <https://www.youtube.com/watch?v=PXr5gVGlbNU&list=PLUxDCM4ujDqXD12P2Vm2Ruo6VfQVJRvv> accessed on July 5, 2024

Background Theory:

Architecture description, defines an architecture description language as "any form of expression for use in architecture descriptions"

An architecture description language (ADL) is used in software engineering to create a description of software architecture. It is necessary to develop such a language to model complex processors at a higher level of abstraction and enable automatic analysis and generation of efficient tools and prototypes.



K. J. Somaiya College of Engineering, Mumbai-77

Software ADLs are used for representing and analysing software architectures where they capture the behavioural specifications of the components and their interactions that comprise the software architecture.[2] The hardware ADLs capture the structure (hardware components and their connectivity) and the behaviour (instruction set) of processor architectures and these ADLs have been successfully used as a specification language for processor development [2] Many executable models, including simulators, compilers, and hardware implementations, are created using the ADL specification. These models can be used to perform a variety of design automation activities, including synthesis, exploration, simulation, compilation, testing, and validation.

Programming languages connect all architectural abstractions to particular point solutions; in contrast, ADLs purposefully reduce or alter this binding. This is how ADLs differ from programming languages.

ADLs are distinct from modeling languages like UML in that the former focus on the depiction of components, while the latter are more interested in the behaviors of the whole.

The ADLs can be divided into three groups: mixed, behavioral, and structural, depending on the type of information. However, modern ADLs can be divided into four groups according to their goals: ADLs that are compilation-oriented, simulation-oriented, synthesis-oriented, and validation-oriented.

An ADL is a computer language used to describe software architectures.

Few examples of ADL

- Acme (developed by CMU)
- AADL (standardized by the SAE)
- C2 (developed by UCI)
- SBC-ADL (developed by National Sun Yat-Sen University)
- Darwin (developed by Imperial College London)
- Wright (developed by CMU).

With little difference

- ArchiMate (now a standard of The Open Group)
- ABACUS (developed by the University of Technology, Sydney).

An architecture's transferable abstraction, early design decisions, and mutual communication are all facilitated by the use of architecture standard notation (ADL) for architecture representation. In the past, box-and-line drawings that were annotated with details about the component's nature, attributes, connection semantics, and overall system behavior were used to depict architectures. As a result, ADLs overcome the limitations of the linguistic approach to the formal representation of architectures. Furthermore significant, advanced ADLs provide early research and testing of the viability of architectural design choices.

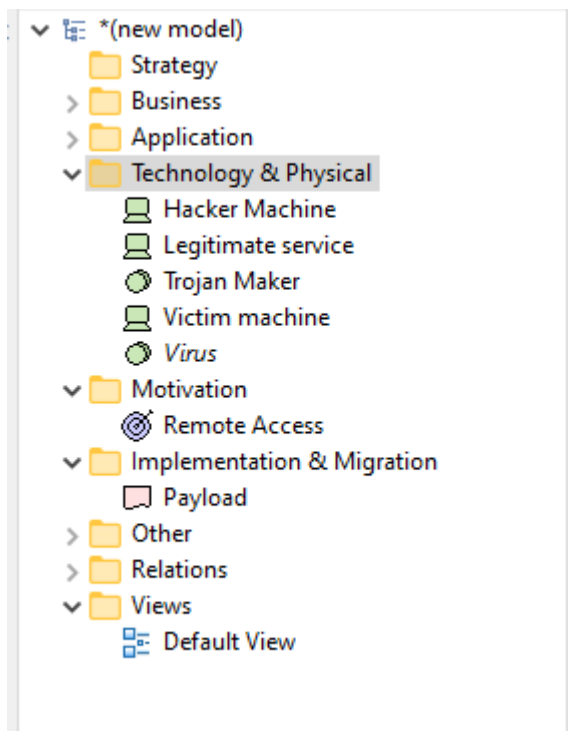
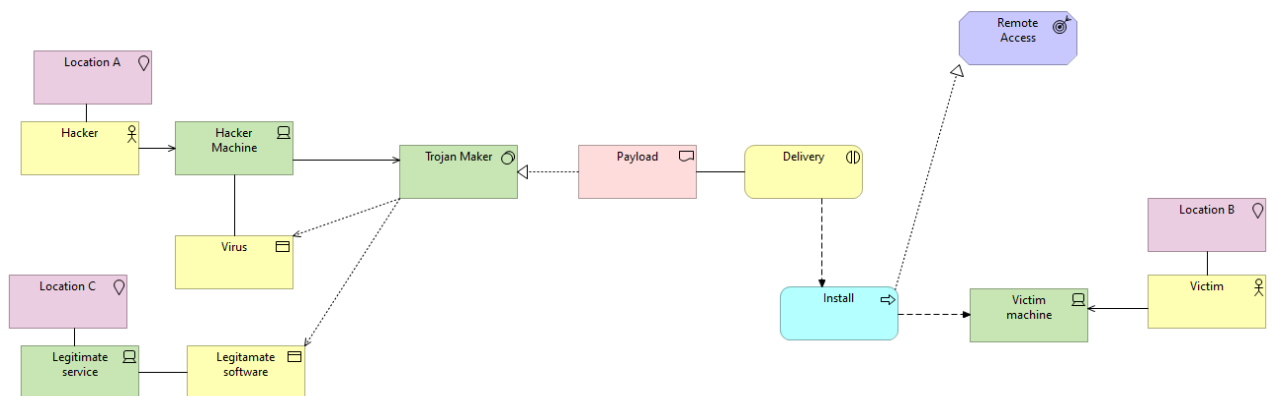
Laboratory Work:

1. Students are required to study characteristics of *any* ADL.

K. J. Somaiya College of Engineering, Mumbai-77

2. Install ArchiMate
3. Learn basic concepts such as Application Layer, Technology Layer etc.
4. Draw architecture diagram of any one from following
 - a. Food Delivery Application
 - b. Any B2B application

Architecture Diagram for Trojan Maker software.





K. J. Somaiya College of Engineering, Mumbai-77

Post Laboratory questions:

1. Need of ADL

Architecture Description Languages (ADLs) are essential tools in software and systems engineering for capturing and communicating architectural designs. They provide a structured way to depict system components, their interactions, and their behaviors, facilitating better understanding and management of complex systems. ADLs help in early-stage design decisions, ensuring architectural integrity and consistency throughout development.

2. Merits and Limitations of ADL

ADLs enhance collaboration among stakeholders by providing a common language and visualization of system architecture. They also support automated analysis and simulation, improving reliability and performance prediction. Yet, ADLs can be complex to learn and use, requiring specialized knowledge and tools. Additionally, they may not fully capture dynamic aspects of systems, such as runtime behavior and emergent properties, which limits their applicability in certain contexts.