Questions:

- What kinds of systems are we going to model?
- What did you model last year?
- What is the connection between EAST-ADL model and UPPAAL model?
- What are the relationships among four abstract levels in EAST-ADL?

- How to make a group?

What is a “model”?

- A representation of something. It captures not all attributes of the represented thing, but rather only those seeming relevant. The model is created for a certain purpose and stakeholders.

- A model is built with a model language with an abstract syntax that covers the basic concepts of MBSE like requirements, structural system elements and functions.

What is a "model"?

- Typically the users focus is on the diagram and notation, i.e., called concrete syntax -- Much more important is the world behind the concrete syntax.

- Abstract syntax and semantic of the model elements, e.g., the data structure and the meaning.

- The abstract syntax is the enabler to do computer-based analysis, simulation, traceability, etc., -- That's the difference between a model and a drawing.

Architecture

“Computer architecture is a set of rules and methods that describe the functionality, organization, and implementation of computer systems. Some definitions of architecture define it as describing the capabilities and programming model of a computer but not a particular implementation.” – Wikipedia

“Computer architecture, like other architecture, is the art of determining the needs of the user of a structure and then designing to meet those needs as effectively as possible within economic and technological constraints.” -- Frederic P. Brooks, Jr. Ch2 of a book (planning a computer system)

Architecture

- Visualizing, specifying, constructing, and documenting a SW intensive system demands that the system be viewed from a number of perspectives*.

- Each looks at that system in a different ways at different times over the project's life.

- An important artifact that can be used to manage theses different viewpoints and control incremental development of a system throughout its life cycles.

*Different perspectives: Different stakeholders, end-users, analysis, developer, designer, tester, system integrator, etc.

Architecture is the set of decisions about:

- The organization of a SW system;
- The selection of the structural elements and their interfaces by which the system is composed;
- Their behaviors, as specified in the collaborations among those elements;
- The composition of these structural and behavioral elements into progressively larger subsystems;
- The guidance of organization: the static and dynamic elements, their interfaces, their collaborations and compositions.

Model-Driven Architecture (MDA)

- Model-based approach for engineering complex software systems.
- Decoupling the SWA from the complicated platform, while allowing both to be co-designed.
- Designing embedded systems (ES) because models can be easily evolved as HW and SW requirements evolve.

What is a “component” ?

- An individual **software component** is the basic unit of functionality := a module that encapsulates a set of related function (or data).

- OO programming: an approach to designing modular reusable software systems. Function object: an object with a single method (in C++, this method would be the function operator, "operator()") that acts much like a function (like a C/C++ pointer to a function).

- System processes are placed into separate components: all of the data and functions inside each component are semantically related.
  ➔ Components are modular and cohesive.

What is a "component" ?

- Modular programming: a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules -- each contains everything necessary to execute only one aspect of the desired functionality.

- CBSE (component-based SW Engineering): A reuse-based approach to defining, implementing and composing loosely coupled independent components into systems

What is a “component” ?

- A software object, meant to interact with other components, encapsulating certain functionality or a set of functionalities.

- Has a clearly defined interface and conforms to a prescribed behavior common to all components within an architecture. Multiple components may be composed to build other components.

- Expected to exhibit certain behaviors and characteristics that let them participate in the component structure and interact with its environment and other components.

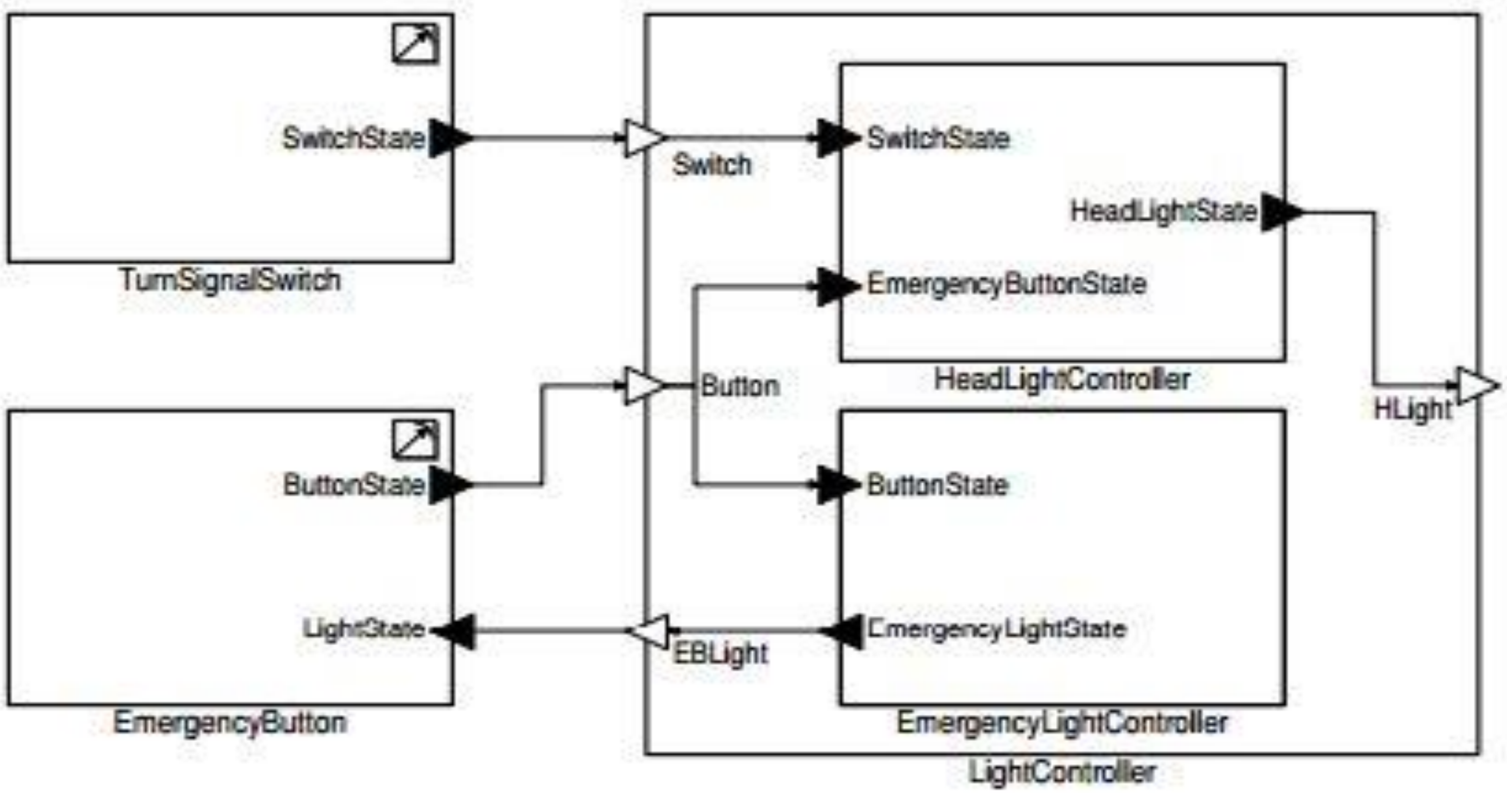


Figure 1: Example of componentized software architecture.

- Components
- Ports
- Connectors
- Hierarchy of SWC interconnected through Ports.
- Components communicate through ports connected by connectors

- Atomic component
- Composition component
- Components can be aggregated using composition components, e.g. LightController aggregates HeadLightController & EmergencyLightController

Software architecture (SWA) & Components

- All SW systems have an architecture that can be viewed in terms of the decomposition of the system into components, connectors, and attachments representing units of system functionality and their potential run-time interactions.

- Placing constraints on their interactions permits the assembly of groups of component and connector types into families of systems designated architectural styles. Patterns of interaction can support reasoning w.r.t certain system-related quality attributes such as modifiability, reliability, and confidentiality.

- SWA is focused on in the early design phase when the overall structure of the system is designed to satisfy functional and non-functional requirements.

Architecture Description Languages (ADL)

- Modeling notation to support Architecture-Based Development;
- Used to define and model system architecture prior to detailed design and implementation;
- Cover SW features such as processes, thread, data and subprograms as well as HW components such as processors, devices, buses, etc.;
- Related to OO Modeling language and Module interconnection languages

Architecture Analysis & Design Language (AADL)

- AADL was first developed in the field of avionics, known formerly as the Avionics Architecture Description Language

- Intended for design both the HW and the SW of an embedded real-time system.

- Define physical port-to-port connections as well as logical flows through chains of ports.

- Component definitions are divided into:

1. *Component types* define the features visible to other components;
2. *Component implementations* define the inner parts of the component

Unified Modeling Language (UML)

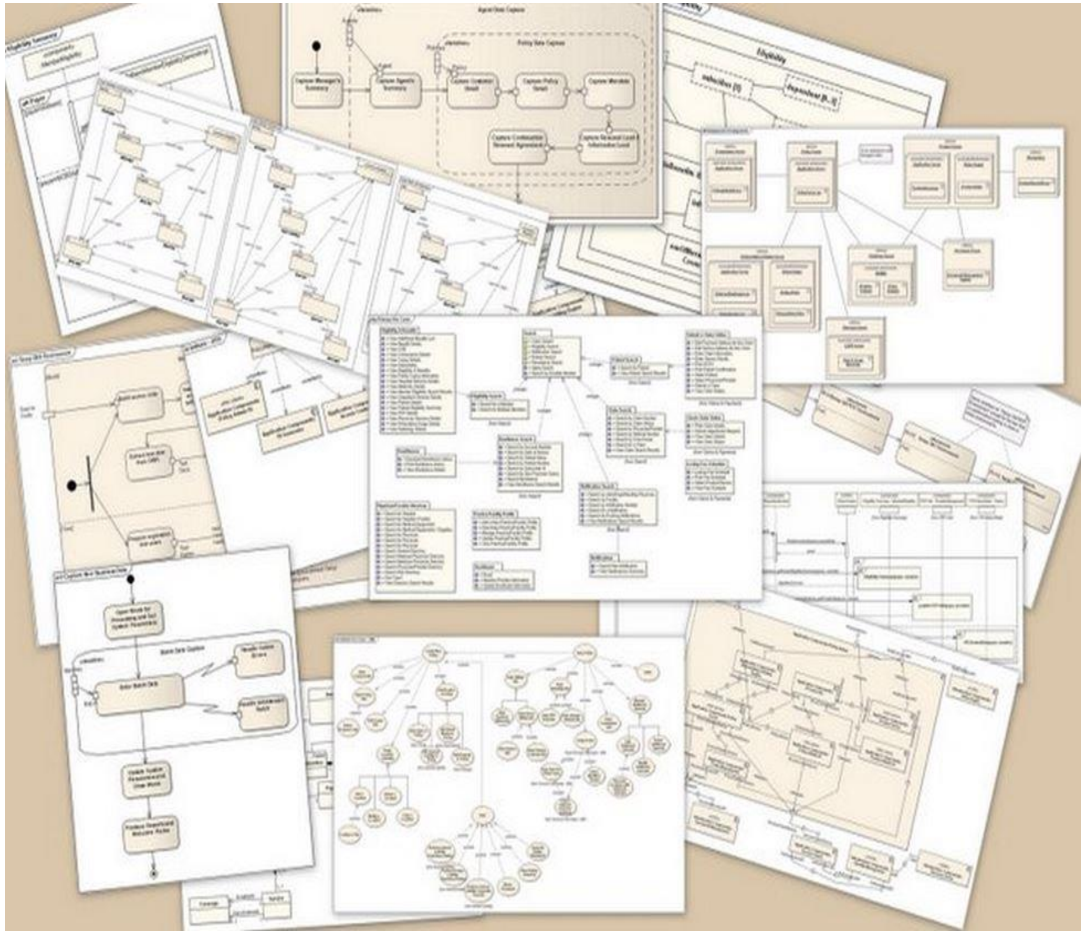
- A general-purpose modeling language in the field of SE -- designed to provide a standard way to visualize the design of a system.

- Developed by James Rumbaugh, Grady Booch, and Ivar Jacobson at Rational Software in the nineties.

- UML is not an ADL and was not intended to be. However, it is rather suitable to model a system.

- Version 2.0 of UML was released in 2004. It included the Object Constraint Language (OCL) -- a declarative language intended to describe the model in plain text

Unified Modeling Language (UML)



- Offers a way to visualize a system's architectural blueprints in a diagram including elements such as:[
- Any activities (jobs)
- Individual components of the system and how they can interact with other software components
- How the system will run
- How entities interact with others (components and interfaces)
- External user interface

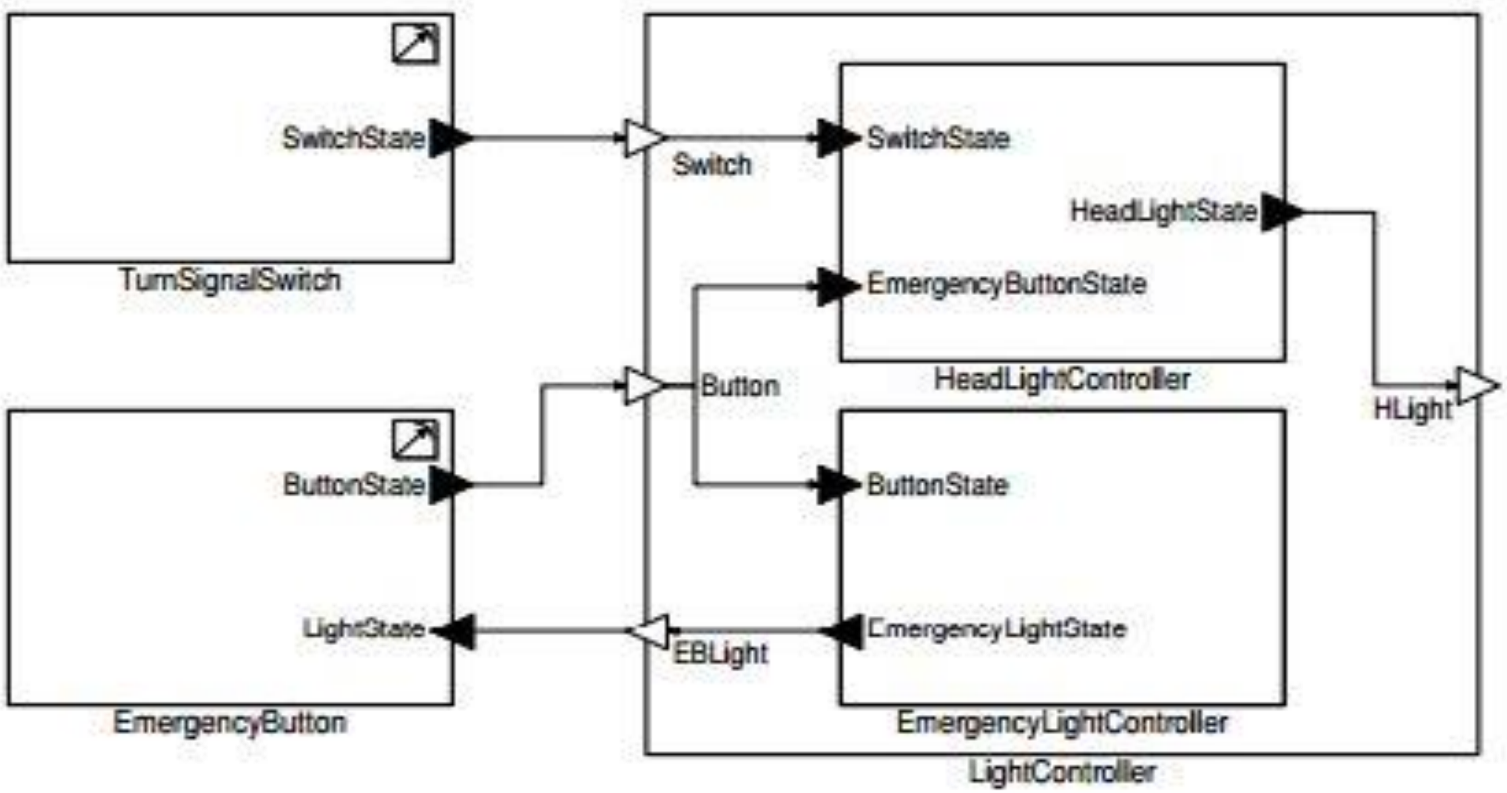


Figure 1: Example of componentized software architecture.

- Components
- Ports
- Connectors
- Hierarchy of SWC interconnected through

Ports.

- Components communicate through ports connected by connectors

- Atomic component
- Composition component
- Components can be aggregated using composition components, e.g. LightController aggregates HeadLightController & EmergencyLightController

EAST-ADL

- **EAST-ADL** is an Architecture Description Language (ADL) for automotive embedded systems, developed in several European research projects. It is designed to complement AUTOSAR with descriptions at higher level of abstractions. Aspects covered by EAST-ADL include vehicle features, functions, requirements, variability, software components, hardware components and communication.

- www.est-adl.info/ www.atesst.org/ www.maenad.eu

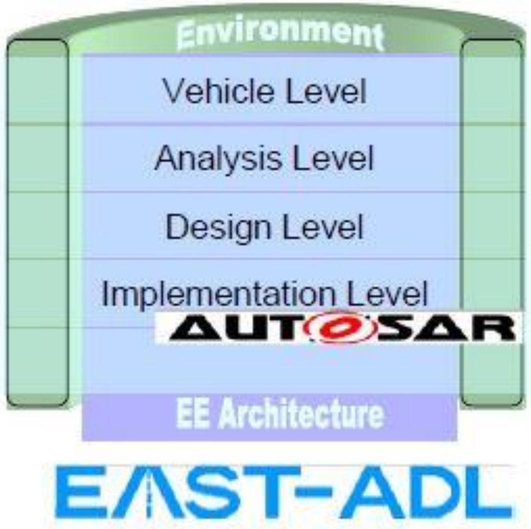
- **AUTOSAR** (AUTomotive Open System ARchitecture) -- an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers, in order to provide a basic infrastructure to assist with developing vehicle software, user interfaces and management for all application domains.

EAST-ADL Complements AUTOSAR

- An information structure including aspects beyond the Software Architecture
  - Requirements, traceability, feature content, variability definition, safety, etc.

- Provides means to define what the software does
  - An AUTOSAR specification defines the software architecture and information required for SW integration - but is neutral to its functionality

- Provides means to model strategic properties
  - Key vehicle aspects are captured independently of the software architecture



- Supports modelling of error behavior and the representation of safety-related information and requirements

https://github.com/kangeu/DVS2016

- Check the news.
- All the course instruction, news and references will be announced on this repository.