

Component oriented modeling

Spoken Tutorial Project
<http://spoken-tutorial.org>

National Mission on Education through ICT
<http://sakshat.ac.in>

Bhargava Nemmaru
FOSSEE, IIT Bombay



22 June 2015



Learning Objectives

In this tutorial, we are going to learn how to:

Learning Objectives

In this tutorial, we are going to learn how to:

- ▶ **instantiate a model**

Learning Objectives

In this tutorial, we are going to learn how to:

- ▶ **instantiate a model**
- ▶ **define connector class**

Learning Objectives

In this tutorial, we are going to learn how to:

- ▶ instantiate a model**
- ▶ define connector class**
- ▶ develop model of a simple electric circuit using component models**



System Requirements

- ▶ **OpenModelica 1.9.2**

System Requirements

- ▶ **OpenModelica 1.9.2**
- ▶ **Ubuntu OS 14.04**

System Requirements

- ▶ **OpenModelica 1.9.2**
- ▶ **Ubuntu OS 14.04**
- ▶ **Any OS: Linux, Windows, Mac OS X or FOSSEE OS on ARM**

Prerequisites

- ▶ **Knowledge of class definition in Modelica**

Prerequisites

- ▶ **Knowledge of class definition in Modelica**
- ▶ **Knowledge of package definition in Modelica**

Prerequisites

- ▶ **Knowledge of class definition in Modelica**
- ▶ **Knowledge of package definition in Modelica**
- ▶ **Knowledge of icon and diagram annotations in Modelica**

Prerequisites

- ▶ Knowledge of class definition in Modelica
- ▶ Knowledge of package definition in Modelica
- ▶ Knowledge of icon and diagram annotations in Modelica
- ▶ Prerequisite tutorials are mentioned on our website



www.spoken-tutorial.org



Class Instantiation

- ▶ **Modelica classes can be instantiated**

Class Instantiation

- ▶ **Modelica classes can be instantiated**
- ▶ **Eg: An individual can be considered an instance of human being class**

Class Instantiation

- ▶ **Modelica classes can be instantiated**
- ▶ **Eg: An individual can be considered an instance of human being class**
- ▶ **Instance of a class has the same variables and equations as the class**

Syntax

<name-of-class><name-of-instance>;

Example: Human Sam(age = 40);

Component orientation

- ▶ **component models represent single physical phenomenon**

Component orientation

- ▶ **component models represent single physical phenomenon**
- ▶ **component models can be instantiated and connected to produce desired effect**

Component orientation

- ▶ **component models represent single physical phenomenon**
- ▶ **component models can be instantiated and connected to produce desired effect**
- ▶ **eg: development of RLC circuit from resistor, inductor and capacitor models**

Acausal Connectors

- ▶ **serve as interface between component instances**

Acausal Connectors

- ▶ serve as interface between component instances
- ▶ defined using **connector** class

Acausal Connectors

- ▶ serve as interface between component instances
- ▶ defined using **connector** class
- ▶ eg: pin for electrical components

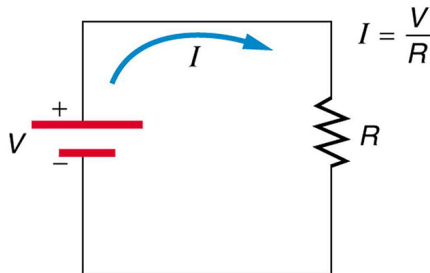
Acausal Connectors

- ▶ serve as interface between component instances
- ▶ defined using **connector** class
- ▶ eg: pin for electrical components
- ▶ contain across and **flow** variables

Acausal Connectors

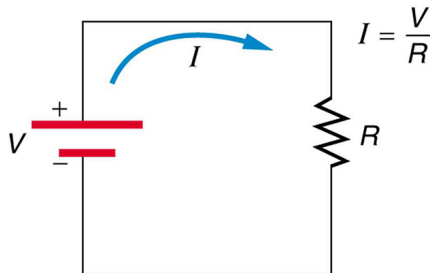
- ▶ serve as interface between component instances
- ▶ defined using **connector** class
- ▶ eg: pin for electrical components
- ▶ contain across and **flow** variables
- ▶ cannot contain equations

Electric Circuit Diagram



► **Voltage(V) = $V_o \cdot \sin(2 \cdot \pi \cdot f \cdot t)$**

Electric Circuit Diagram



- ▶ Voltage(V) = $V_o * \sin(2 * \pi * f * t)$
- ▶ V_o - 10 V, f - 1 Hz, Resistance(R) - 5

foss
be
educ



Solution Methodology

- ▶ **Resistor and Voltage Source have two pins: Positive and Negative**



Solution Methodology

- ▶ **Resistor and Voltage Source have two pins: Positive and Negative**
- ▶ **Define a connector named 'pin'**

Solution Methodology

- ▶ **Resistor and Voltage Source have two pins: Positive and Negative**
- ▶ **Define a connector named 'pin'**
- ▶ **Define a class named 'Ground' with one instance of 'pin' connector**

Solution Methodology

- ▶ **Resistor and Voltage Source have two pins: Positive and Negative**
- ▶ **Define a connector named 'pin'**
- ▶ **Define a class named 'Ground' with one instance of 'pin' connector**
- ▶ **Define a class named 'Resistor'**



Solution Methodology

- ▶ Resistor and Voltage Source have two pins: Positive and Negative
- ▶ Define a connector named 'pin'
- ▶ Define a class named 'Ground' with one instance of 'pin' connector
- ▶ Define a class named 'Resistor'
- ▶ 'Resistor' class should have two instances of 'pin': Positive and Negative

Solution Methodology

- ▶ Resistor and Voltage Source have two pins: Positive and Negative
- ▶ Define a connector named 'pin'
- ▶ Define a class named 'Ground' with one instance of 'pin' connector
- ▶ Define a class named 'Resistor'
- ▶ 'Resistor' class should have two instances of 'pin': Positive and Negative

Solution Methodology (contd.)

- ▶ Define a class named 'VoltageSource' with two instances of 'pin' connector

Solution Methodology (contd.)

- ▶ Define a class named 'VoltageSource' with two instances of 'pin' connector
- ▶ Define a class named 'simpleCircuit'



Solution Methodology (contd.)

- ▶ Define a class named 'VoltageSource' with two instances of 'pin' connector
- ▶ Define a class named 'simpleCircuit'
- ▶ 'simpleCircuit' should have instances of 'Resistor' and 'VoltageSource'



Solution Methodology (contd.)

- ▶ Define a class named 'VoltageSource' with two instances of 'pin' connector
- ▶ Define a class named 'simpleCircuit'
- ▶ 'simpleCircuit' should have instances of 'Resistor' and 'VoltageSource'
- ▶ Connect the respective pins of 'Resistor' and 'VoltageSource'



Assignment

Construct an electric circuit with two resistors in series with one voltage source. Use the component models for resistor and voltage source.

About the Spoken Tutorial Project

- ▶ Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarises the Spoken Tutorial project

About the Spoken Tutorial Project

- ▶ Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarises the Spoken Tutorial project
- ▶ If you do not have good bandwidth, you can download and watch it

Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- ▶ Conducts workshops using spoken tutorials
- ▶ Gives certificates to those who pass an online test
- ▶ For more details, please write to contact@spoken-tutorial.org

Forum to answer questions

- ▶ Do you have questions in **THIS Spoken Tutorial?**
- ▶ Choose the minute and second where you have the question.
- ▶ Explain your question briefly.
- ▶ Someone from the **FOSSEE** team will answer them.

Please visit <http://forums.spoken-tutorial.org/>



Textbook Companion Project

- ▶ **The FOSSEE team coordinates coding of solved examples of popular books**
- ▶ **We give honorarium and certificate to those who do this**

For more details, please visit this site:

<http://OM.fossee.in/Textbook-Companion-Project>



Lab Migration Project

- ▶ **The FOSSEE team helps migrate commercial simulator labs to OpenModelica**
- ▶ **We give honorarium and certificates to those who do this**

For more details, please visit this site:

<http://OM.fossee.in/lab-migration-project>



Acknowledgements

- ▶ Spoken Tutorial Project is a part of the Talk to a Teacher project
- ▶ It is supported by the National Mission on Education through ICT, MHRD, Government of India
- ▶ More information on this Mission is available at <http://spoken-tutorial.org/NMEICT-Intro>

Thanks!

<http://openmodelica.org>