



# ***DEVS Training Course***

## **Introduction to DEVS**

Prof. Dr. Valdemar Vicente Graciano Neto

# Short Bio




## Dr. Valdemar Vicente Graciano Neto

- Double degree PhD in Computer Science Brazil/France;
- Experience with DEVS since 2016: research, training and supervision;
- Contributor to the Modeling and Simulation Body of Knowledge (MSBoK), to be published by Springer;
- Adoption of DEVS in his PhD to model and simulate software-intensive systems, including Flood Monitoring System and the Brazilian Space System;
- Tutorial using DEVS in ECSA 2021 and CBSOFT 2021;
- Assistant Professor permanent position at the Federal University of Goiás, in Brazil, since 2014.
- **More:** [https://dblp.org/pers/hd/n/Neto:Valdemar\\_Vicente\\_Graciano](https://dblp.org/pers/hd/n/Neto:Valdemar_Vicente_Graciano)




# M&S

- 
- Modeling and Simulation (M&S);
  - Definition 1 (Banks 1999):

*“A simulation is an approximate imitation of the operation of a process or system”.*

# M&S

- 
- Modeling and Simulation (M&S);
  - Definition 2 (MSBOK 2021):

*“Simulation is providing experience under controlled conditions for training, i.e., for gaining/enhancing competence in one of the three types of skills:*


- (1) motor skills (virtual simulation, or use of simulators),*
- (2) decision and/or communication skills (constructive simulation such as business games, war games, or peace games; aka serious games), and*
- (3) operational skills (live simulation).*



# DEVS

## (Discrete-Events Systems Specification)

# DEVS

- 
- A formal notation based on:
    - Atomic models (Parts)
    - Coupled models (a whole composed of other parts)
  
  - There are several DEVS specifications:
    - Python;
    - Java;
    - ...

# DEVS



- MS4ME
  - Based on Eclipse and Java
  - Supported by DEVSNL (DEVS Natural Language)
  - Animation
  - It allows the specification of sequence diagrams

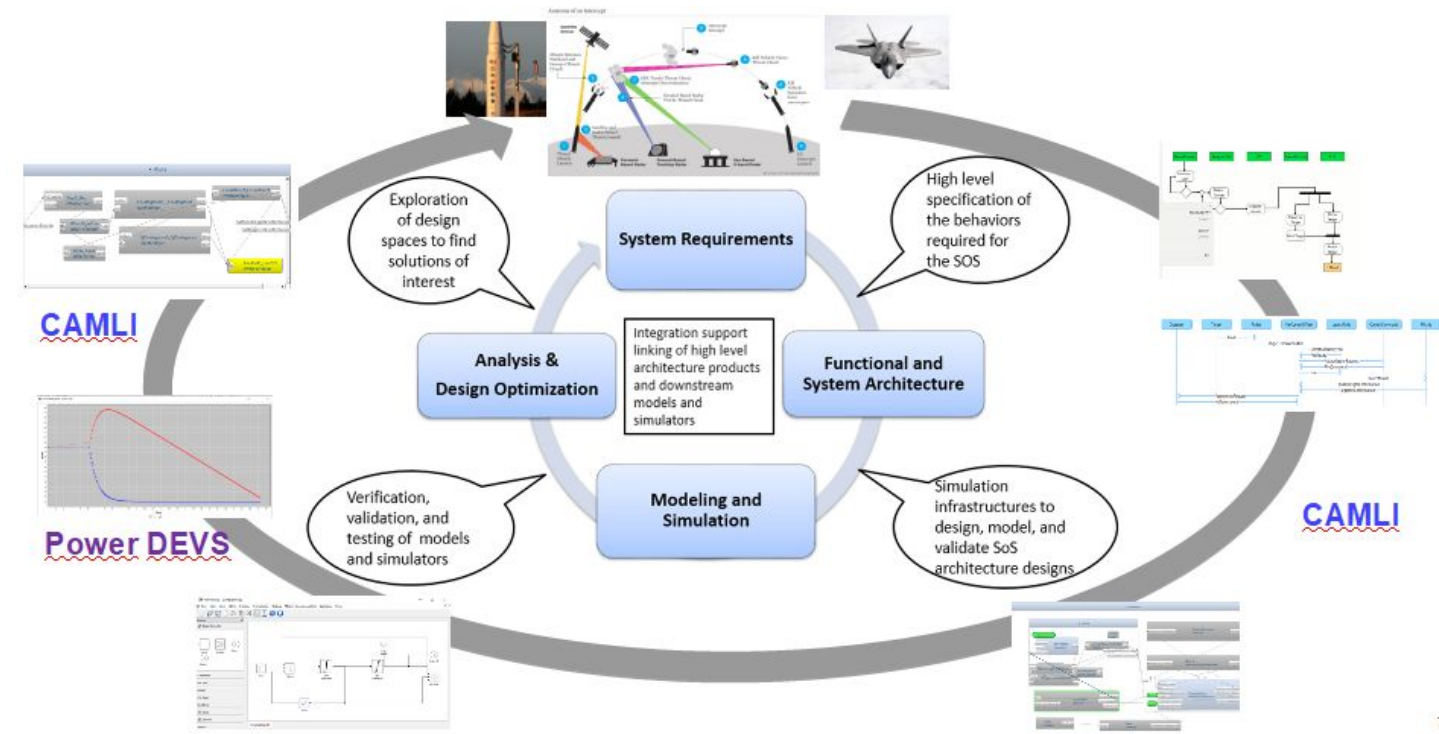
# DEVS

- 
- Why you need DEVS for complex SoS M&S and how DEVS can provide better solutions than other commercial tools?



# DEVS

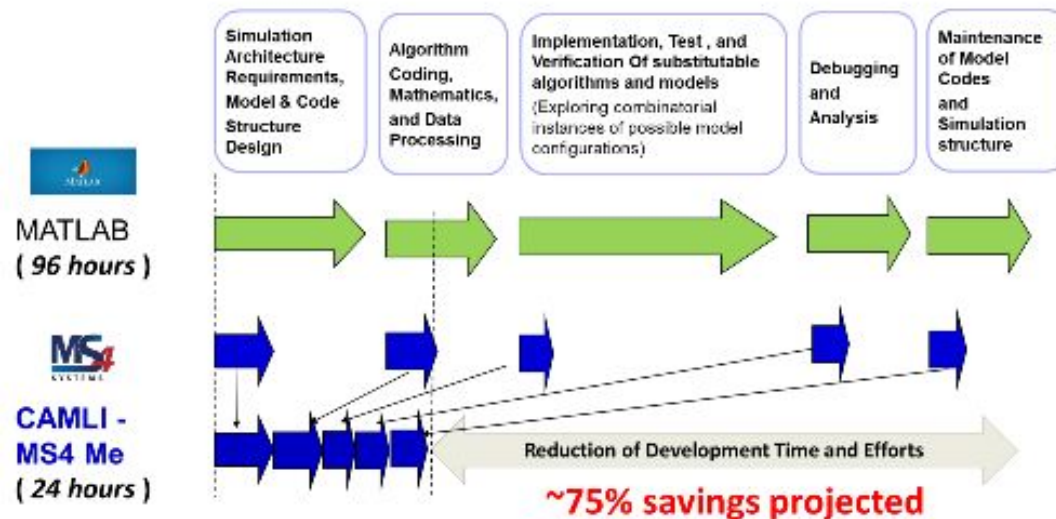
- Why you need DEVS for complex SoS M&S and how DEVS can provide better solutions than other commercial tools?



# DEVS

- Why you need DEVS for complex SoS M&S and how DEVS can provide better solutions than other commercial tools?

Comparison of Development Time and Efforts - BMDSSim Example



# DEVS

- Why you need DEVS for complex SoS M&S and how DEVS can provide better solutions than other commercial tools?

## Comparison of MBSE full life cycle support

Example: **No Magic** MBSE projects call for these separate independent tools requiring significant upfront investment, personnel training, and organizational burden



**No Magic** MBSE Solutions need different tools:


- **Modeling:** MagicDraw + SysML
- **Simulation:** Cameo Simulation Toolkit
- **Integration:** Phoenix Model Center
- **Analytical Models:** STK and MATLAB




CAML1 –  
MS4 Me  
software

DEVS provides a  
universal Integrated  
Solution to close this  
major Gap

# DEVS


- 
- MS4ME
    - Atomic models are composed of:
      - Data types
      - Events
      - Input and Output Ports
      - A state machine to specify behaviors (Mealey Machine - IOST)

# DEVS

- 
- MS4ME
    - Atomic models are composed of:
      - Data types
      - Events
      - Input and Output Ports
      - A state machine to specify behaviors (Mealey Machine - IOST)

**It can have less than all that, ok?!**

# DEVS


- 
- Introduction to MS4ME
    - Open the simulation environment
    - Create or use a workspace (don't forget to save it)
    - Create a DEVS Modeling Project
    - Create a .SES (System Entity Structure – Coupled Model) file
    - Create specific Atomic models

# DEVS



- MS4ME
  - Hello World DEVS – two systems exchanging messages;
  - First:
    - Building the simulation using Activity Diagram;

# DEVS

- 
- MS4ME instructions
    - Create a new DEVS Modeling Project with name Hello;
    - In MS4Me Launch Page, select “Sequence Diagram” > Create New;
    - Specify the model;
    - Save it;
    - Create > Model;
    - Prune SES into PES;
    - Run PES in SimViewer.





# A brief look into Atomic Models

# DEVS

- General Rule for Specification of an Atomic Model State Machine:
  - Communication between two systems is only possible if one system is in the sending state and the other is in the receiving state.
  - “Pattern” for States specification:  
DEVS Input:  
passivate in << fromState > >!  
when in << fromState > > and receive <<dataReceived > > go to << toState > >!  
//external transition!  
  
DEVS Output:  
hold in << fromState > > for time 1!  
after << fromState > > output << dataType > >!  
from << fromState > > go to << toState > >! //internal transition!  
<<hold or passivate again to>> <<toState>>!

# DEVS



- Example DEVSNL
- HelloWorld.ses
- SES code:

From the top perspective, sequenceDiagram is made of Someone and Another!

From the top perspective, Someone sends Message to Another!

# DEVS



- Exemplo DEVSNL
- Someone.dnl

//State Machine!  
to start hold in s0 for time 2!  
after s0 output Message!  
from s0 go to s1!  
passivate in s1!

# DEVS

- 
- Example DEVSNL
  - Another.dnl

to start passivate in s0!  
when in s0 and receive Message go to s1!  
passivate in s1!

# Exercise



- 1) Modify the code so that Another also sends messages to Someone and they exchange messages continuously.



# ***DEVS Training Course***

## **Introduction to DEVS**

Prof. Dr. Valdemar Vicente Graciano Neto