VISUAL MODELLING ENVIRONMENT FOR CBD'S

PART 1: READING ASSIGNMENT

MICHAEL DECKERS
20145715

UNIVERITY OF ANTWERP, 2014-2015

INTRODUCTION

Sources:
- Van Mierlo, S., Van Tendeloo, Y., Barroca, B., Mustafiz, S., Vangheluwe, H.,2014. Explicit modelling of a parallel devs experimentation environment.
- Vangheluwe, H., Denil, J., Mustafiz, S., Riegelhaupt, D., Van Mierlo, S., 2014. Explicit modelling of a cbd experimentation environment. In: Proceedings of the Symposium on Theory of Modeling & Simulation – DEVS Integrative. DEVS '14. Society for Computer Simulation International, San Diego, CA, USA, pp. 13:1–13:8. URL http://dl.acm.org/citation.cfm?id=2665008.2665021

- CAUSAL BLOCK DIAGRAMS
  - VISUAL MODELLING LANGUAGE
  - MODELS MATHEMATICAL AND BOOLEAN EXPRESSIONS
- NO MODERN VISUAL EDITOR
  - PREVIOUS VERSION IN ATOM3
    - DEPRECATED
    - NO WORKING DEBUGGING
- GOAL = CREATE COMPLETE MODELLING ENVIRONMENT IN ATOMPM
  - DESIGN
  - SIMULATION
  - DEBUGGING

CBD DEBUGGING

- Two ways
    - Breakpoints
    - Step by step execution
- CBDs: two loops
    - Outer loop: one simulation iteration
        - Breakpoints: automatically stop executing after every or a particular iteration
        - Step by step execution: manually control the execution, halting simulation after every step (Big step)
    - Inner loop: One block calculation
        - Breakpoints: somewhere in the dependency graph
        - Step by step execution: manually control the execution, halting simulation after every block calculation (small step)

ATOMPM MODEL DEBUGGING

- Linking front end to back end
  - Front end = AToMPM
    - Easy to use
    - Design view
    - Simulation view
    - "pretty but dumb"
  - Back end = Python CBD simulation script
    - Difficult to design
    - "smart but ugly"
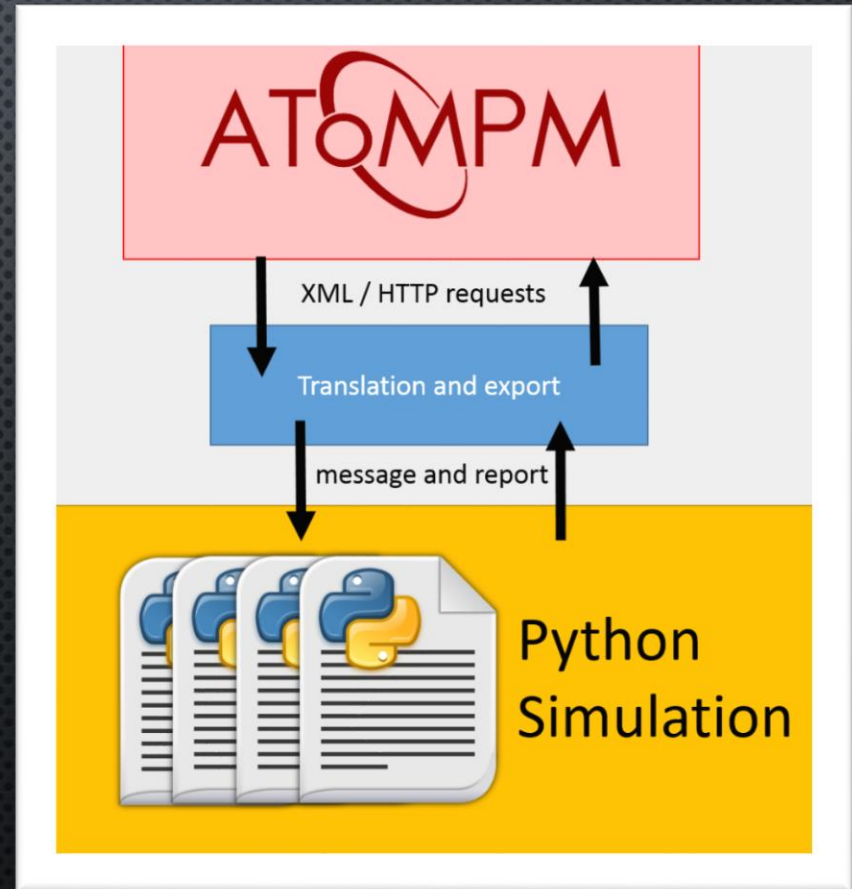
ATOMPM CBD DEBUGGING

- Breakpoints
  - Introduce new "Breakpoint" or "Halting" block
  - Put inbetween two existing blocks to enforce correct order
- Step by step executiong
  - No modification to diagram
  - Only in simulation

CONCLUSION AND FUTURE

- DEBUGGING
  - HALTING EXECUTION OR TRACE EXAMINING
- HALTING EXECUTION
  - BREAKPOINTS OR STEP BY STEP EXECUTION
- BREAKPOINTS
  - INTRODUCE NEW BLOCK TO CBD ONLY FOR THE CAUSE OF DEBUGGING
- LINKING CBD SHELL TO PYTHON SIMULATION ENVIRONMENT

- IMPLEMENTATION WORK
  - START WITH CREATING CBD FORMALISM
  - ADD SIMULATION (LINK WITH PYTHON BACK END)
  - IMPLEMENT DEBUGGING