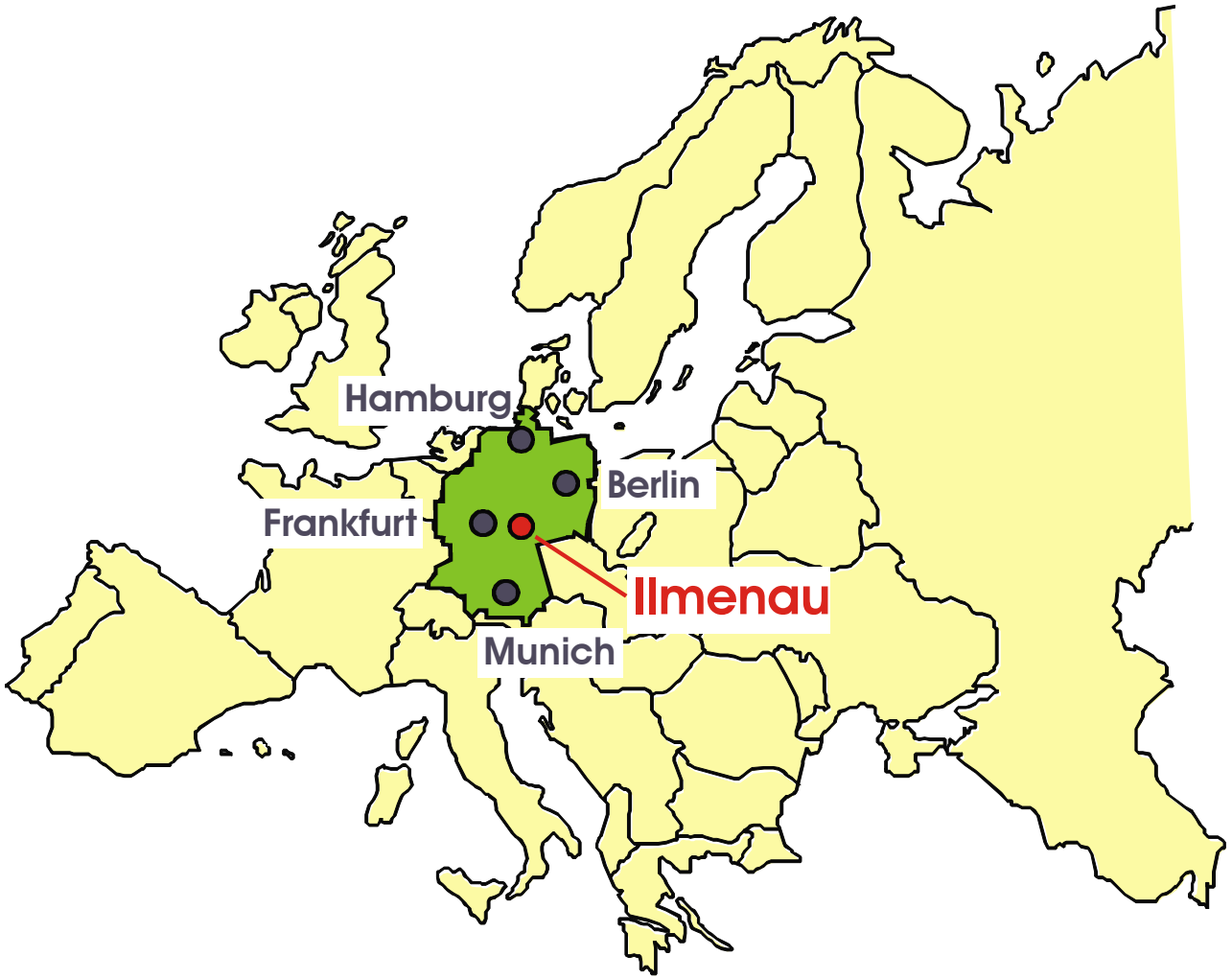
Modeling and Simulation of Operating System Behavior

Bernd Däne, Wolfgang Fengler, Falk Berger

Ilmenau Technical University, Germany

Where is Ilmenau?



Topics

1. Introduction and Overview
2. Modeling Methodology
3. Simulation and Evaluation
4. Conclusion

Parts of this work were and are supported by the Thuringian Ministry of Science, Research and Art (FKZ B509-00002) and by the German Research Council (SFB 622).

1. Introduction

- Model the behavior of Real Time Operating Systems (RTOS)
- Goal: Quantitative Estimation of Properties
- Methodology to model the system behavior in discrete event manner

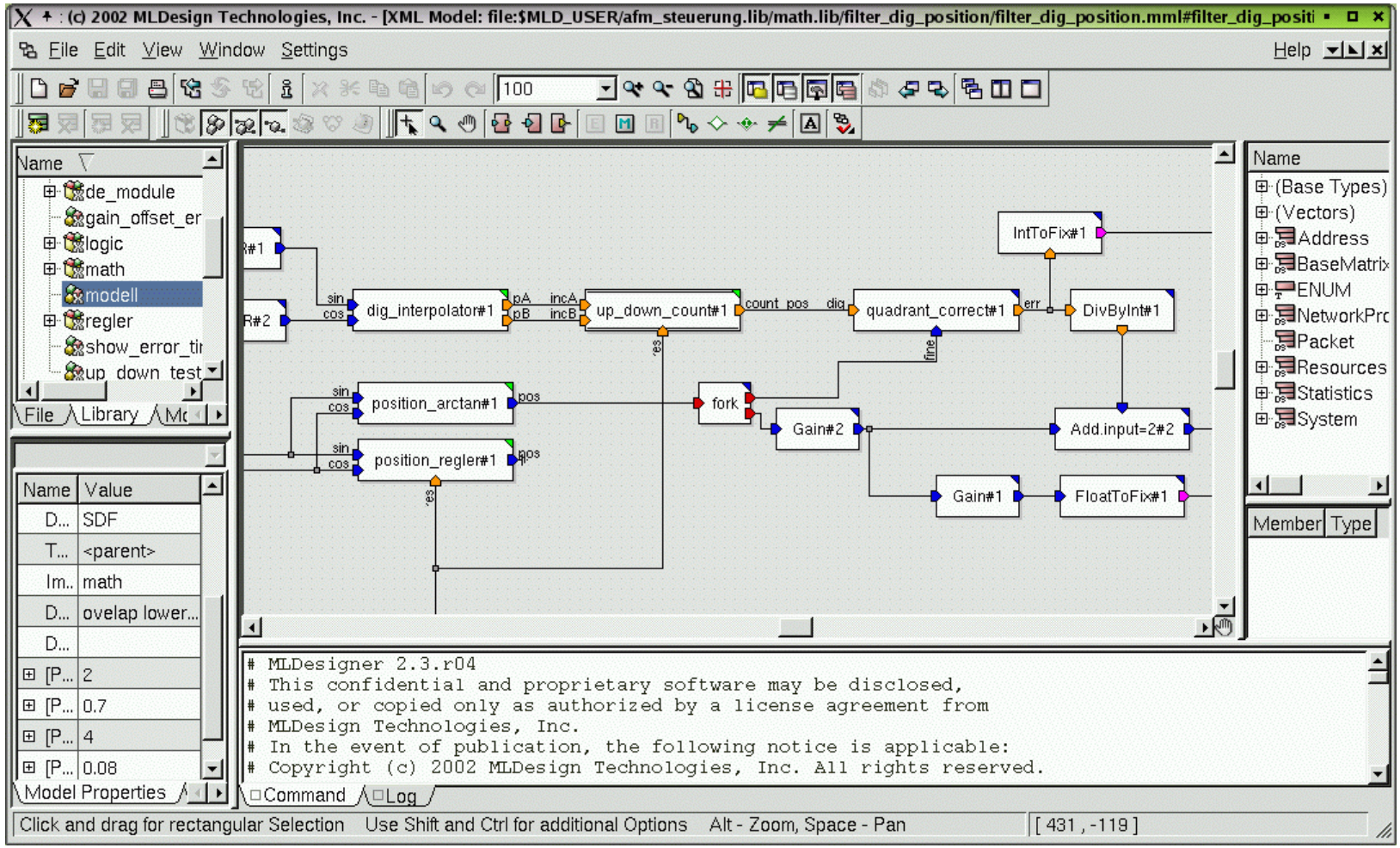
Features of the RTOS

- Multiple scheduling strategies
  - Cooperative task's
  - Non-cooperative task's
- Resource management
- Circular memory buffer

MLDesigner Basics

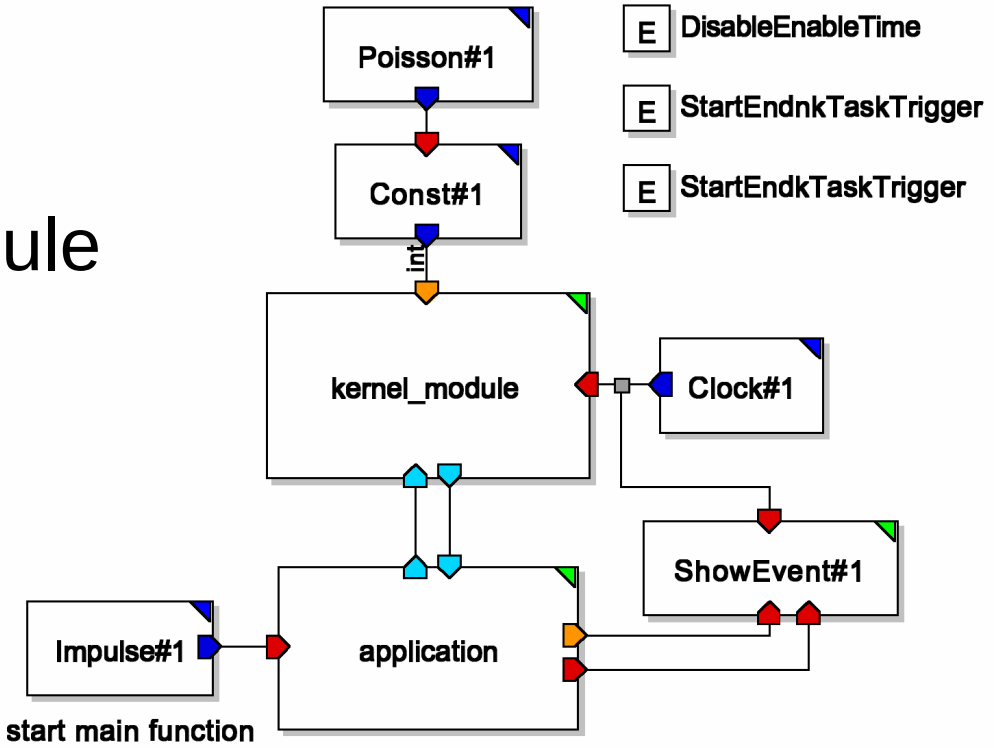
- Hierarchical multi domain modeling framework
- Covers module, system and strategy („mission“) levels
- Combines numerous modeling domains (discrete and continuous paradigms)
- Capabilities for simulation, design check, code generation, export
- Derived from well-known Ptolemy tool (University of Berkeley)

MLDesigner Sample Workspace



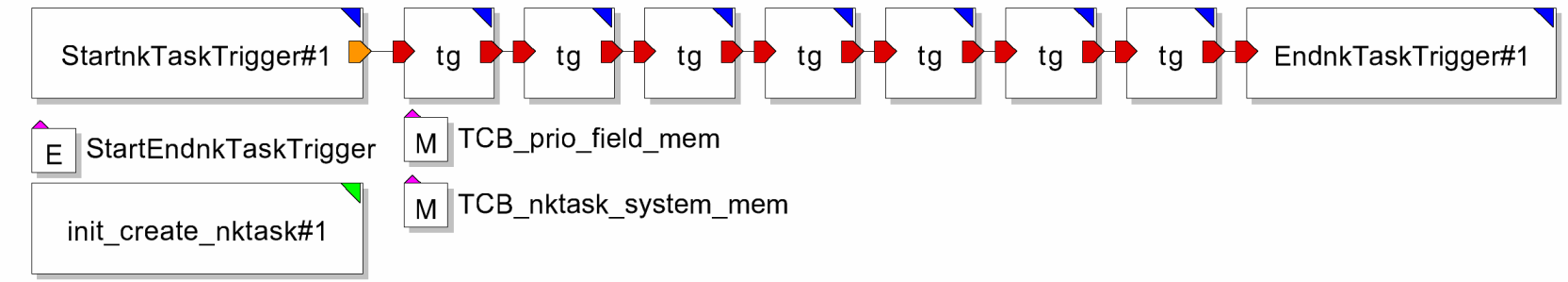
2. Modeling Methodology

- Discrete Event Domain of MLDesigner
- Main Parts
  - Kernel Module
  - Application Module



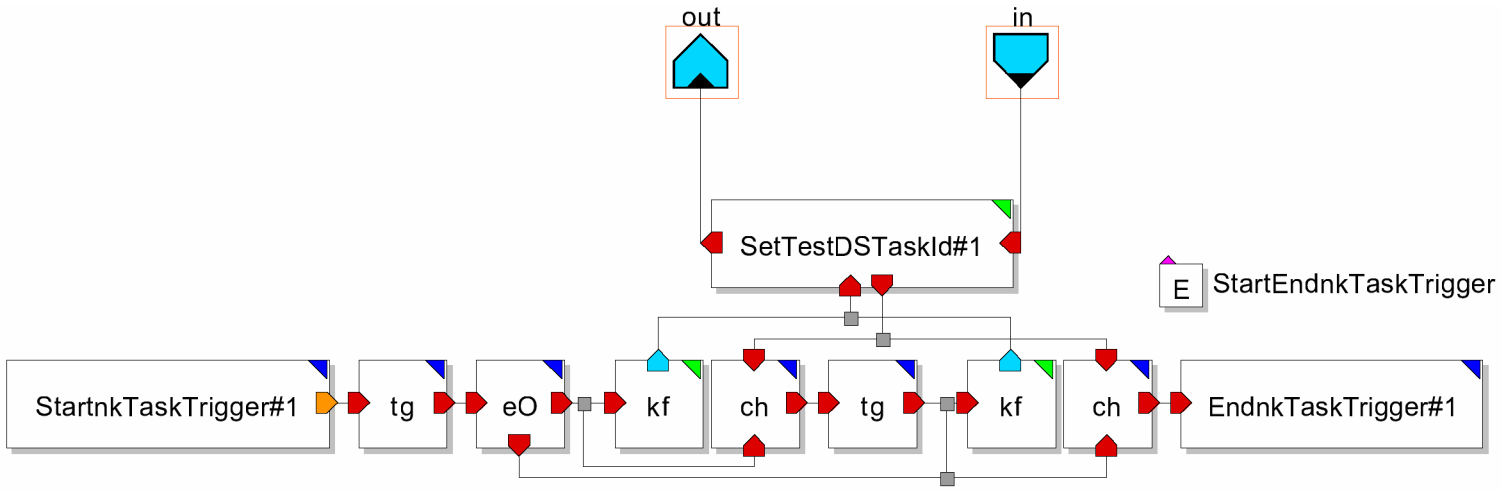
Application Module

- Includes one block for each application task
- Basic elements are atomic blocks (*tg*)
- Task switch on block boundaries



Task Model with Kernel Invocation

- Managed by special interface block
- Each call is initiated by block *kf*
- Block *ch* holds task's control flow

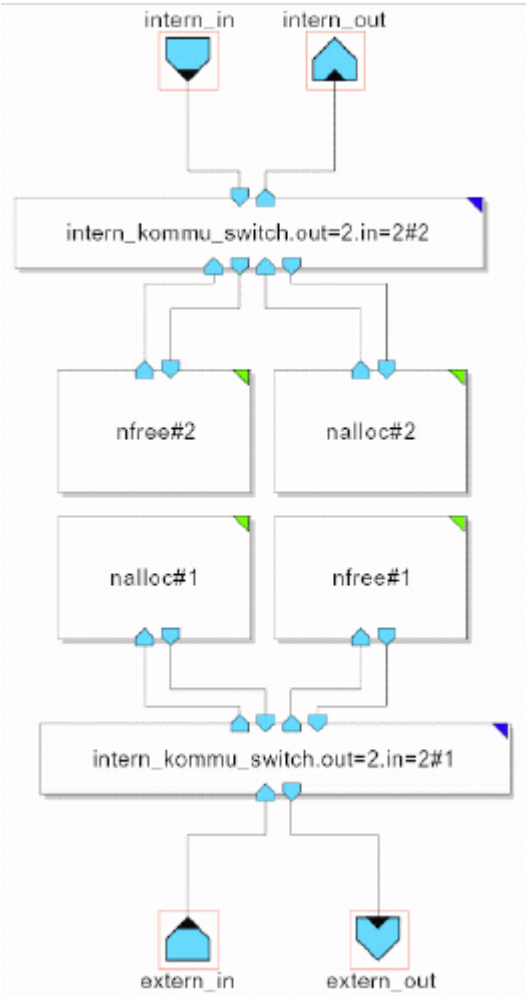


Kernel Module

- Block for task scheduling
- Blocks for system services (messaging, device and memory management)
  - Refer to certain groups of kernel functions
  - Hold specific information (resource occupation)

Memory Management Module

- Simplified structure
- 4 blocks encapsulating basic memory management
  - nalloc (modified malloc())
  - nfree (modified free())
- Blocks share resources

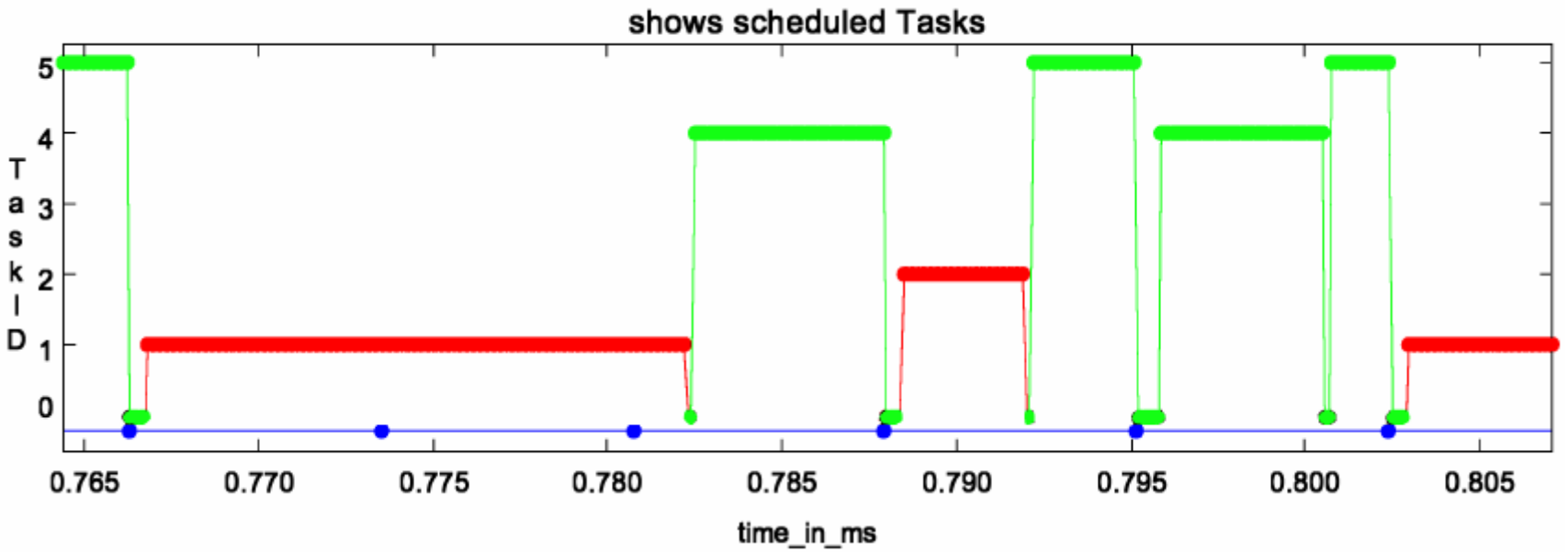


3. Simulation and Evaluation

- Simulation with MLDesigner tool
  - Collects information about timing, task status and resource status
- Different views for evaluation

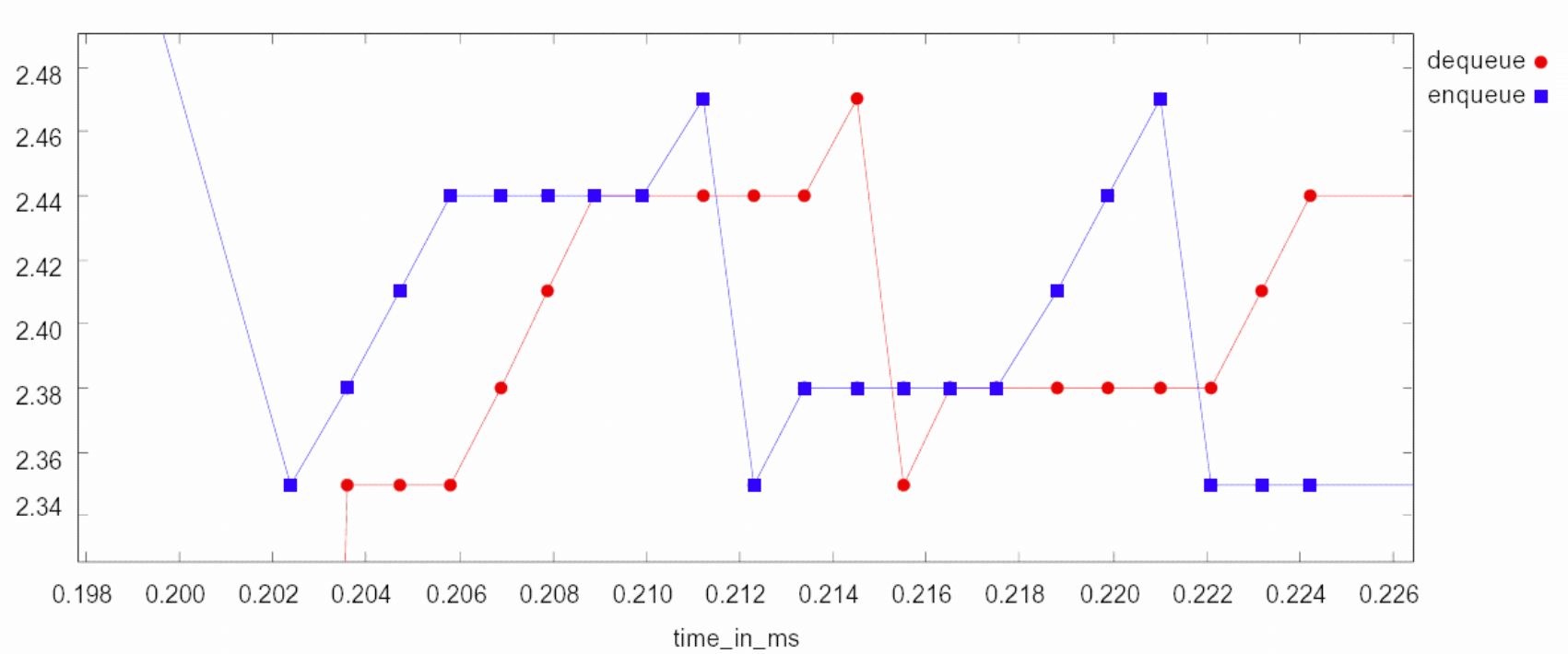
Task Switch Diagram

- Task‘s identified by task-ID
- Shows with task is active at any time



Memory Occupation View

- Activity of a task, that manages memory as a circular buffer



Device Occupation View

- First column shows absolute time
- Second the actual occupation
- Last the queued requests

```
0.2356: dev : owner | wait[prio]
0.2356: 13 -1 -1 |
0.2455552169: dev : owner | wait[prio]
0.2455552169: 13 1 -1 |
0.2611419639: dev : owner | wait[prio]
0.2611419639: 13 1 2 |
0.2920497108: dev : owner | wait[prio]
0.2920497108: 13 1 2 | 6[2]
0.3272022048: dev : owner | wait[prio]
0.3272022048: 13 1 2 | 6[2] 3[99]
0.9865274578: dev : owner | wait[prio]
0.9865274578: 13 1 3 | 6[2]
1.195372084: dev : owner | wait[prio]
1.195372084: 13 1 6 |
1.199432084: dev : owner | wait[prio]
1.199432084: 13 1 6 |
1.225018831: dev : owner | wait[prio]
1.225018831: 13 1 6 |
1.23871059: dev : owner | wait[prio]
1.23871059: 13 1 6 |
1.246435494: dev : owner | wait[prio]
1.246435494: 13 1 6 |
1.26116341: dev : owner | wait[prio]
1.26116341: 13 1 6 |
```

Conclusion

- Can replace some testing
- Cannot replace formal analysis

- Further work:
  - Improvement of the methodology
  - More elements for control flow (loops)
  - Implement software directly from model