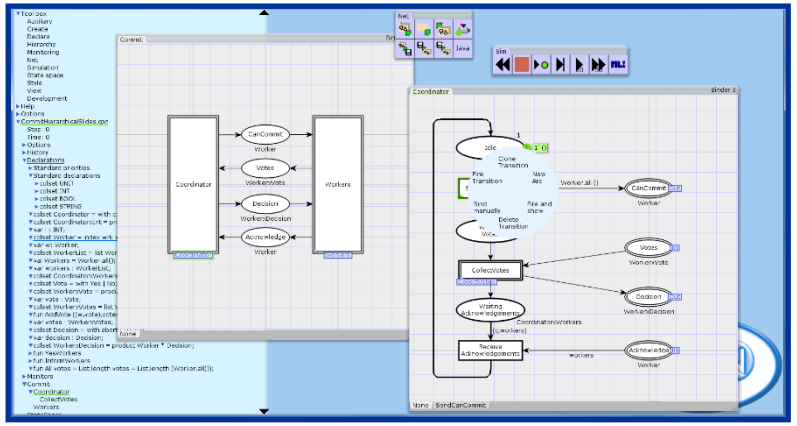
Case Study Parts

Case studies and hands-on experience with CPN Tools



Lars Michael Kristensen
Department of Computer Science, Electrical Engineering, and Mathematical Sciences
Western Norway University of Applied Sciences
Email: lmkr@hvl.no

Outline

- **DAY 1: Theory-Tool**
  - Motivation and overview of the Coloured Petri Nets modelling language
  - Background on Coloured Petri Nets: From Place/Transition Nets to Coloured Petri Nets
  - Syntax and semantics of the basic constructs of Coloured Petri Nets
  - Modules for hierarchical structuring of large Coloured Petri Nets models
- **DAY 2: Case Study**
  - Two examples on the application of Coloured Petri nets within smart software systems
  - Installation and getting started with CPN Tools
  - Hands-on experiments with simulation of CPN models in CPN Tools
  - Hands-on experiments with editing and construction of CPN models in CPN Tools

You are also most welcome to work on your own case study!

More challenging hands-on …

- **CPN modelling and simulation of the Paxos protocol for consensus in distributed systems**

Tutorial Summary: Paxos Explained from Scratch

Hein Meling and Leander Jehl

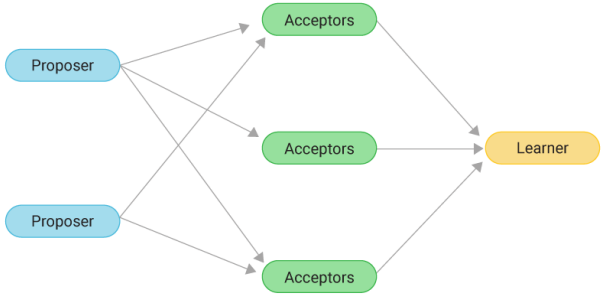
University of Stavanger, Norway

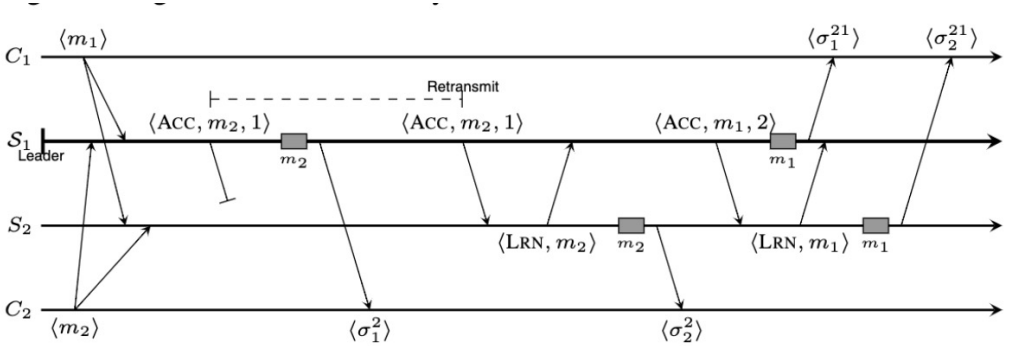
Abstract. Paxos is a flexible and fault tolerant protocol for solving the consensus problem, where participants in a distributed system need to agree on a common value. However, Paxos is reputed for being difficult to understand. This tutorial aims to address this difficulty by visualizing Paxos in a completely new way. Starting from a naive solution and strong assumptions, Paxos is derived in a step-wise fashion. In each step, minimal changes are made to the solution and assumptions, aimed at understanding why the solution fails. In this manner, a correct solution that corresponds to Paxos is eventually reached.

1 Introduction

Paxos is a flexible and fault tolerant consensus protocol that can be used in applications that need to agree on a common value among distributed participants. Paxos was proposed by Lamport in his seminal paper [1] and later gave a simplified description in [2]. Paxos can be used to solve the atomic commit problem in distributed transactions, or to order client requests sent to a replicated state machine (RSM). An RSM provides fault tolerance and high availability, by implementing a service as a deterministic state machine and replicating it on different machines. Paxos is relevant because it is often used in production systems such as Chubby and ZooKeeper [3, 4] among many others. Understanding Paxos is important because it reveals the distinction between a strongly consistent RSM and a primary-backup system.

Both before and after its publication in [1], Paxos attracted much attention for its unorthodox exposition in the form of a fictional parliamentary system, supposedly used by legislators at the Greek island of Paxos. But the scientific contribution was also significant; it provided a new way to implement RSMs, and proved that the protocol guarantees that participants make consistent decisions, irrespective of the number of failures. Clearly Paxos cannot always make progress, e.g. during network partitions, as was shown in [5]. But perhaps most important, Paxos was described in a flexible and general way, ignoring many technical details. This made it an excellent foundation for further research into RSM-based protocols [6–9], aimed at supporting different failure models, wide-area networking, to improve latency, and so on. The fact that these protocols build on the Paxos foundation, which has been formally proven, makes it much easier to reason about their correctness through step-wise modifications of Paxos.

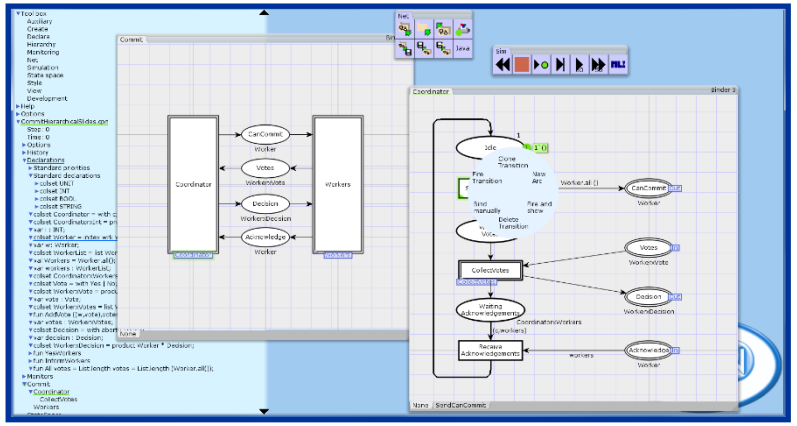




Petri Nets Course Assignment – Coloured Petri Nets Component:
https://github.com/lmkr/cpncourse/blob/master/assignment/assignment.md

Case Study | Part 6

CPN Tools Installation



Lars Michael Kristensen
Department of Computer Science, Electrical Engineering, and Mathematical Sciences
Western Norway University of Applied Sciences
Email: lmkr@hvl.no

CPN Tools installation

- **CPN Tools can be downloaded and installed via www.cpntools.org | https://cpntools.org/2018/01/15/windows/**

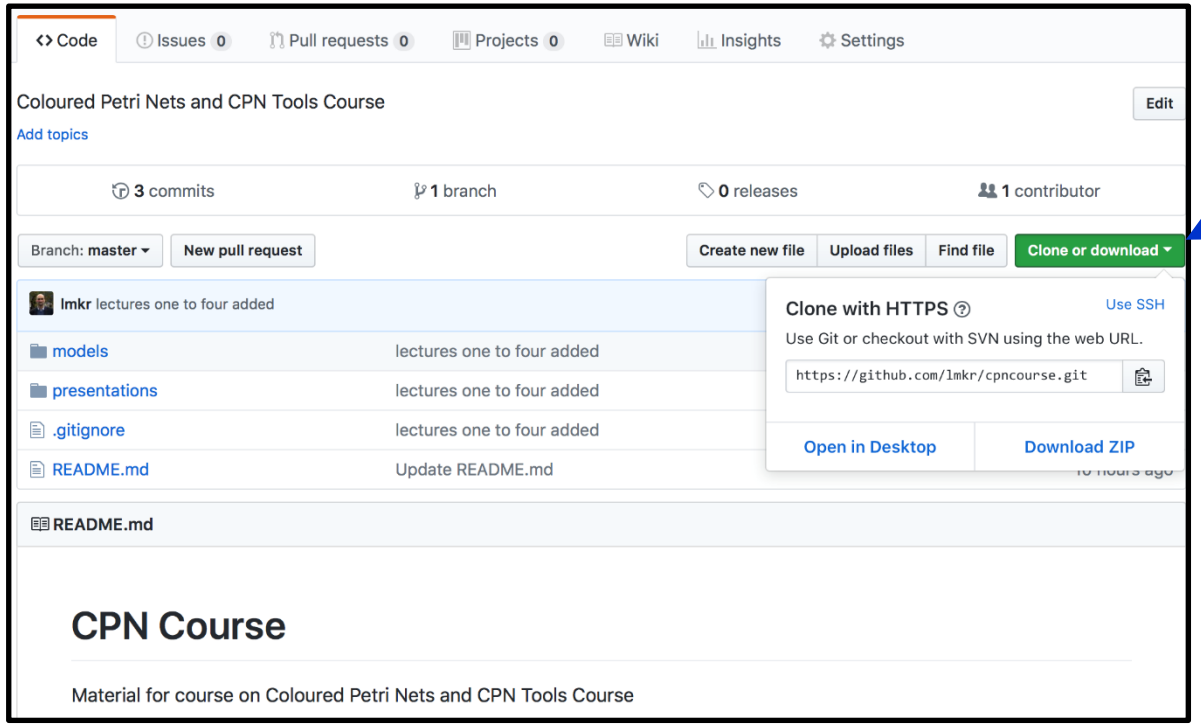
Running on Mac OS / Linux via a virtual machine or emulator

- **Some installations of Windows required the application to be run as administrator**

Material

- **Models and presentations are available via the github repository at**

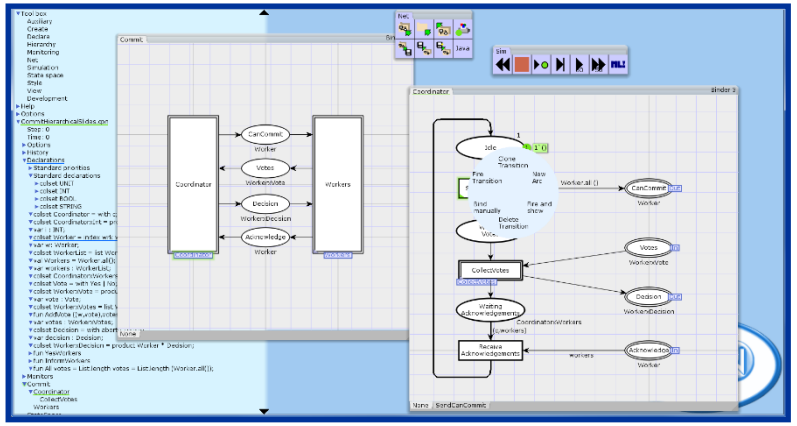
https://github.com/lmkr/cpncourse/acpn



Clone the git-repository or download as a zip-file

Case Study | Part 7

Navigation and simulation of CPN models



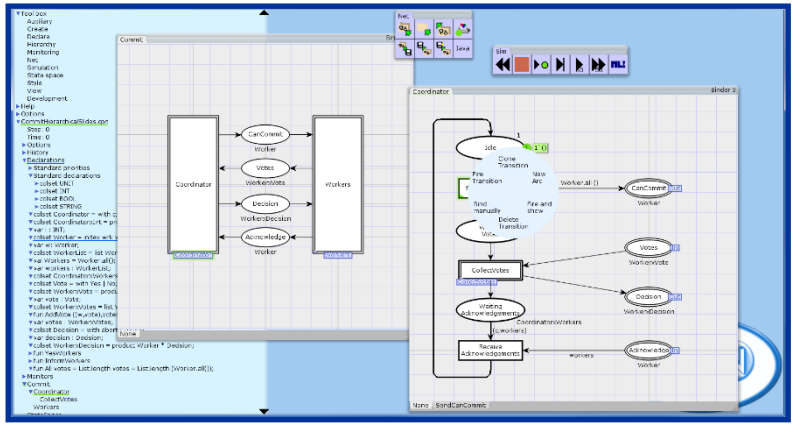
Lars Michael Kristensen
Department of Computer Science, Electrical Engineering, and Mathematical Sciences
Western Norway University of Applied Sciences
Email: lmkr@hvl.no

Hands-on: Simulation

- **The main aim is to become familiar with how to navigate and simulate CPN models**
  1. Open the two-phase-commit protocol CPN model located in the hands-on folder
  2. Use the tools in the Simulation tool palette to conduct interactive and automatic simulations – including manual binding selection
  3. Investigate scenarios where some workers vote No and a scenario where all workers vote Yes

Case Study | Part 8

Construction and editing of CPN models



Lars Michael Kristensen
Department of Computer Science, Electrical Engineering, and Mathematical Sciences
Western Norway University of Applied Sciences
Email: lmkr@hvl.no

Hands-on: Editing

- **The main aim is to become familiar with how to edit and modify CPN models**
- **Modify the two-phase commit protocol CPN model from hands-on 1 such that**
  1. The coordinator and the workers no longer return to their initial idle state upon completion of the protocol.
  2. The coordinator terminates in a place CoordinatorCompleted and the workers terminate in a place WorkerCompleted
  3. The coordinator puts an Abort-token or a Commit-token on a new place Result to indicate the result of the transaction
  4. Each worker records its vote locally by putting a tuple-token consisting of the worker and the vote in a new place Votes
- **Use simulation to validate the revised model**