

Seminário

A ideia do seminário é estudar um assunto relacionado à disciplina, porém não apresentado durante as aulas, e apresentar o assunto em sala de aula.

Os assuntos do seminário estão contidos no livro “Concurrency – State Models & Java Programs”, Jeff Magee & Jeff Kramer, ou nas páginas relacionadas ao livro (Primeira Edição).

Os autores do livro fornecem uma ferramenta baseada em sistema de transição rotulada, contendo uma linguagem de descrição e um visualizador, não estudados em sala de aula. Isso é proposital, vocês vão pesquisar o assunto para poder apresentar.

O link apontado na página da disciplina contém o software da ferramenta, exemplos e anotações de aula (PowerPoint). Nós podemos emprestar o livro para a elaboração do seminário (temos uma cópia).

Os problemas também foram escolhidos de forma proposital. Eles saem um pouco do estilo dos exemplos didáticos e requerem certo conhecimento do problema para se interpretar os resultados da ferramenta. Os problemas, modelos em sistema de transição e avaliações estão descritos no livro (Primeira Edição, a Segunda Edição pode ter outra disposição dos assuntos).

Na apresentação, use a ferramenta para demonstrar a modelagem e as verificações possíveis. Quase todos os exemplos já tem os fontes do modelo disponíveis para download.

Todos devem estar preparados para começar no dia da 1ª apresentação...

Alisson Cavalcante
Anny Caroline
João Luiz Vieira da Costa
Rosana Landi
Thalita Cannavezes

Alexandre 1. Introduction

- 1.1 Concurrent Programs.¶
- 1.2 The Modeling Approach.
- 1.3 Practice.
- 2. Processes and Threads.
- 2.1 Modeling Processes.
- 2.2 Implementing Processes.
- 3. Concurrent Execution.
- 3.1 Modeling Concurrency.
- 3.2 Multi-Threaded Programs. .
- 4. Shared Objects and Mutual Exclusion.
- 4.1 Interference.
- 4.2 Mutual Exclusion in Java.
- 4.3 Modeling Mutual Exclusion. .

Alisson Cavalcante 5. Monitors and Condition Synchronization.

- 5.1 Condition Synchronization.
- 5.2 Semaphores.
- 5.3 Bounded Buffers.
- 5.4 Nested Monitors.
- 5.5 Monitor Invariants. .

João Luiz 6. Deadlock.

- 6.1 Deadlock Analysis.
- 6.2 Dining Philosophers Problem. .

Anny Caroline 7. Safety and Liveness Properties.

- 7.1 Safety.
- 7.2 Single-Lane Bridge Problem.
- 7.3 Liveness.
- 7.4 Liveness of the Single-Lane Bridge.
- 7.5 Readers-Writers Problem. .

????? - 8. Model-Based Design.

- 8.1 From Requirements to Models.
- 8.2 From Models to Implementation.

Rosana 9. Dynamic Systems.

- 9.1 Golf Club Program.
- 9.2 Golf Club Model.
- 9.3 Fair Allocation.
- 9.4 Revised Golf Ball Allocator.
- 9.5 Bounded Overtaking.
- 9.6 Bounded Overtaking Golf Ball Allocator.
- 9.7 Master-Slave Program.
- 9.8 Master-Slave Model. .

Thalita 10. Message Passing.

- 10.1 Synchronous Message Passing.
- 10.2 Asynchronous Message Passing.
- 10.3 Rendezvous. .

???? 11. Concurrent Architectures.

- 11.1 Filter Pipeline.
- 11.2 Supervisor-Worker.
- 11.3 Announcer-Listener. .

???? 12. Timed Systems.

- 12.1 Modeling Timed Systems.
- 12.2 Implementing Timed Systems.
- 12.3 Parcel Router Problem.
- 12.4 Space Invaders. .

?????? 13. Program Verification.

- 13.1 Sequential Processes.
- 13.2 Modeling Condition Synchronization.
- 13.3 Modeling Variables and Synchronized Methods.
- 13.4 Bounded Buffer Example.
- 13.5 Readers-Writers Example. .

?????? 14. Logical Properties.

- 14.1 Fluent Propositions.
- 14.2 Temporal Propositions.
- 14.3 Fluent Linear Temporal Logic (FLTL).