Sunday

	Registration and Opening (8-9)		
Jörg Desel	Introduction + Modeling Behavior of Distributed Systems (9-11)		
Wolfgang Reisig	The Essence of Petri Nets (11-13)		
Jetty Kleijn	From Nets to Behaviour: Petri Nets and their Semantics (14-16)		
Maciej Koutny	From Behaviour to Nets: Petri Net Synthesis (16-18)		

Monday

Javier Esparza	Verification of Distributed Systems and Protocols (10-13)	Dirk Fahland Xixi Lu	Process Mining 1: Model Discovery and Event Log Pre-Processing (9-13)
Alex Yakovlev	Workcraft – Application of Petri Nets to Asynchronous Circuits Design (14-18)	Dirk Fahland Xixi Lu	Process Mining 1: Model Discovery and Event Log Pre-Processing (14-18)

Tuesday

Sławomir Lasota	The Reachability Problem for Petri Nets (10-13)	Laure Petrucci	Model Checking Timed and Strategic Properties (10-13)
Alex Yakovlev	Workcraft – Application of Petri Nets to Asynchronous Circuits Design (14-18)	Lars Kristensen	Coloured Petri Nets for Concurrent Software Systems Engineering (14-18)

Wednesday

Wolfgang Reisig	Heraklit (10-11)		
	Posters (11-13)		
Natalia Sidorova	Process Mining 2 – Conformance Checking (14-18)	Lars Kristensen	Coloured Petri Nets for Concurrent Software Systems Engineering (14-18)

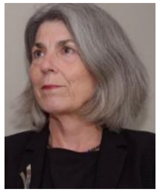
Thursday

Natalia Sidorova	Process Mining 2 – Conformance Checking (9-13)	Łukasz Mikulski	Analysis of Concurrent Systems: Traces and Causal Structures (10-13)
Karsten Wolf	Efficient Verification of Petri Net Models (14-18)	Elvio Amparore Susanna Donatelli	Performance Evaluation and (Stochastic) Verification of (Stochastic) Petri Nets (14-18)

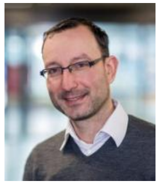
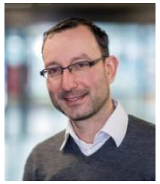
Friday

Kamila Barylska	Analysis and Synthesis of Some Subclasses of Petri Nets (10-13)	Marco Montali	Data-Aware Processes: Modelling and Verification (10-13)
Karsten Wolf	Efficient Verification of Petri Net Models (14-18)	Elvio Amparore Susanna Donatelli	Performance Evaluation and (Stochastic) Verification of (Stochastic) Petri Nets (14-18)

Sunday, September 3rd

 **Jörg Desel**	**Introduction + Modeling Behavior of Distributed Systems (9-11)** Introduction: - welcome, organisational matters, goal of the Advanced Course, brief overview of the lectures, what we offer the participants and what we expect from them Modeling Behavior of Distributed Systems (fundamental concepts related to distributed systems, the roles of system models and behavioral models, how Petri nets are suitable for capturing various aspects of behavioral models of distributed systems): - distributed systems and their behavior - causality, independence, concurrency - models of behavior - observation of executions vs. specification of processes - creation of models - choices, interfaces, policies -the role of time
 **Wolfgang Reisig**	**The Essence of Petri Nets (11-13)** Starting with C.A. Petri's seminal thesis in 1962, Petri nets challenge some assumptions about fundamentals of discrete behavior modeling in informatics: - Petri nests emphasize that cause and effect of an event is locally confined. This yields a lot of insight into the nature of single behaviors of a system, and allows for specific classes of nets and analysis techniques; - Petri nets base the notion of state on concepts of propositional and first order logic, and can be conceived as a dynamization of logic; - Petri nets can be composed in a way that avoids the state explosion problem of automata, process algebras and many other modeling techniques.
 **Jetty Kleijn**	**From Nets to Behaviour: Petri Nets and their Semantics (14-16)** Petri Nets - local dynamics, occurrence of single transitions sequential semantics: - firing sequences - (labelled) transition systems - behavioural equivalences Elementary Net Systems - relations between transition occurrences: causality, conflict, concurrency, and independence partial order semantics: - processes - partial orders - traces (equivalence classes of firing sequences) Other net models - more relations between transition occurrences generalising partial orders
 **Maciej Koutny**	**From Behaviour to Nets: Petri Net Synthesis (16-18)** - Elementary Net Synthesis - Different Forms of the Synthesis Problem - General Theory of Net Synthesis - Synthesis of P/T-Nets - Synthesis of Nets with the Step Firing Rule

Monday, September 4th

 **Javier Esparza**	**Verification of Distributed Systems and Protocols (10-13)** Verifying Liveness Properties of Replicated Systems: - verification of qualitative liveness properties of replicated systems under stochastic scheduling, - a finite-state program, executed by an unknown number of indistinguishable agents, - a Presburger stage graph, - Presburger-definable sets of configurations, - complexity of the verification problem, - an incomplete procedure for the construction of Presburger stage graphs, - the theory of well-quasi-orders, - the structural theory of Petri nets and vector addition systems, - a set of benchmarks (population protocols).	 **Dirk Fahland**  **Xixi Lu**	**Process Mining 1: Model Discovery and Event Log Pre-Processing (9-13)** Process Discovery: - the automatic learning of process models using event data - well-known discovery algorithms (directly-follows-graph miner, inductive miner) - the event-data preprocessing techniques (creating views, filtering event logs, event abstraction, and label refinements) - quality and accuracy of the discovered models Practical sessions: - popular process mining tools and libraries. - hands-on exercises with real-life event data (preprocessing event data and discovering process models effectively).
 **Alex Yakovlev**	**Workcraft – Application of Petri Nets to Asynchronous Circuits Design (14-18)** The aim of this course is to give students an understanding of an asynchronous design methodology based on Petri nets and Signal Transition Graphs and to introduce them to asynchronous design flow using some tools under the Workcraft framework, illustrating this flow with an example of power electronics controller. Day 1: Theory and Tools in Workcraft Slots 1 and 2: Fundamental principles of asynchronous design, models and tools Slot 3: Introductory practical in Workcraft (Petri net models, Signal Transition Graphs - STGs, Design of C-element) Slot 4: Logic Synthesis from STGs (Design of basic VME bus controller) Day 2: Elements of design flow in Workcraft and case study Slot 1: State encoding Slot 2: Logic decomposition Slot 3: Hierarchical design and verification Slot 4: Case study: Design of DC/DC buck controller	 **Dirk Fahland**  **Xixi Lu**	**Process Mining 1: Model Discovery and Event Log Pre-Processing (14-18)** Process Discovery: - the automatic learning of process models using event data - well-known discovery algorithms (directly-follows-graph miner, inductive miner) - the event-data preprocessing techniques (creating views, filtering event logs, event abstraction, and label refinements) - quality and accuracy of the discovered models Practical sessions: - popular process mining tools and libraries. - hands-on exercises with real-life event data (preprocessing event data and discovering process models effectively).

Tuesday, September 5th

Sławomir Lasota

The Reachability Problem for Petri Nets (10-13)

- Reachability and coverability; relevance and brief history of the problems
- Vector addition systems and counter programs
- Decidability of reachability
- ACKERMANN lower bound for reachability

Laure Petrucci

Model Checking Timed and Strategic Properties (10-13)

Part 1: Timed models
- Timed Automata
- Time(d) Petri Nets syntax and concrete semantics
- region graph (symbolic semantics)

Part 2: Temporal Logics for Timed Systems
- LTL and CTL
- Adding time: TCTL
- Adding Strategies: TATL, STCTL

Part 3: Model checking algorithms
- Exploration strategies:
- BFS, DFS, ID

Part 4: Conclusion and going further
- adding temporal parameters
- tools: Imitator, Romeo

Alex Yakovlev

Workcraft – Application of Petri Nets to Asynchronous Circuits Design (14-18)

The aim of this course is to give students an understanding of an asynchronous design methodology based on Petri nets and Signal Transition Graphs and to introduce them to asynchronous design flow using some tools under the Workcraft framework, illustrating this flow with an example of power electronics controller.

Day 1: Theory and Tools in Workcraft
Slots 1 and 2: Fundamental principles of asynchronous design, models and tools
Slot 3: Introductory practical in Workcraft (Petri net models, Signal Transition Graphs - STGs, Design of C-element)
Slot 4: Logic Synthesis from STGs (Design of basic VME bus controller)

Day 2: Elements of design flow in Workcraft and case study
Slot 1: State encoding
Slot 2: Logic decomposition
Slot 3: Hierarchical design and verification
Slot 4: Case study: Design of DC/DC buck controller

Lars Kristensen

Coloured Petri Nets for Concurrent Software Systems Engineering (14-18)

Coloured Petri Nets for Concurrent Software Systems Engineering (introduction):
- Coloured Petri Nets (combining Petri Nets with a functional programming language)
- formal foundation (concurrency, synchronization, communication, and resource sharing)
- sequential computations on data

The theory-tool module:
- syntax and semantics of the CPN modelling language
- editing, simulation, and validation of CPN models
- hierarchical CPNs (support for scalability, abstraction, and maintainability)
- examples of CPN models from recent projects in the domain of smart software systems
- the case-study (constructing and simulating CPN models using CPN Tools)

Wednesday, September 6th

<table>
<tr><td>
Wolfgang Reisig</td><td colspan="3">**Heraklit (10-11)**

HERAKLIT combines a research program and a development project, aiming at an infrastructure to represent (i.e., to model), communicate and analyze computer-integrated systems.</td></tr>
<tr><td></td><td colspan="3">**Posters (11-13)**

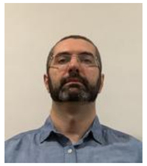
- short presentations
- poster session</td></tr>
<tr><td>
Natalia Sidorova</td><td>**Process Mining 2 – Conformance Checking (14-18)**

- Introduction to Conformance Checking: importance, goals, and challenges.
- Types of conformance checking: replay-based, alignment-based, and model-based techniques.
- Evaluation metrics for conformance checking.
- Multi-perspective conformance checking: resource, data, and control-flow perspectives.
- Interpreting deviations in control-flow, resource and data perspectives.
- Hands-on exercises using ProM.
- Conformance checking on real-world event and discussion and analysis of the results obtained.
- Open research challenges in conformance checking.</td><td>
Lars Kristensen</td><td>**Coloured Petri Nets for Concurrent Software Systems Engineering (14-18)**

Coloured Petri Nets for Concurrent Software Systems Engineering (introduction):
- Coloured Petri Nets (combining Petri Nets with a functional programming language)
- formal foundation (concurrency, synchronization, communication, and resource sharing)
- sequential computations on data

The theory-tool module:
- syntax and semantics of the CPN modelling language
- editing, simulation, and validation of CPN models
- hierarchical CPNs (support for scalability, abstraction, and maintainability)
- examples of CPN models from recent projects in the domain of smart software systems
- the case-study (constructing and simulating CPN models using CPN Tools)</td></tr>
</table>

Thursday, September 7th

 **Natalia Sidorova**	**Process Mining 2 – Conformance Checking (9-13)** - Introduction to Conformance Checking: importance, goals, and challenges. - Types of conformance checking: replay-based, alignment-based, and model-based techniques. - Evaluation metrics for conformance checking. - Multi-perspective conformance checking: resource, data, and control-flow perspectives. - Interpreting deviations in control-flow, resource and data perspectives. - Hands-on exercises using ProM. - Conformance checking on real-world event and discussion and analysis of the results obtained. - Open research challenges in conformance checking.	 **Łukasz Mikulski**	**Analysis of Concurrent Systems: Traces and Causal Structures (10-13)** Concurrency Paradigms: - histories and observations Systems with step sequence semantics: - elementary net systems with inhibitor and mutex arcs - step traces, subclasses of step traces - general order structures, maximal and closed relations Introduction to systems with interval order semantics: - elementary net systems with inhibitors - interval traces and structures, relationship with Mazurkiewicz traces
 **Karsten Wolf**	**Efficient Verification of Petri Net Models (14-18)** The verification of Petri nets in a hands-on session: - Petri net models and queries from the Model Checking Contest - the performance of several verification techniques - interplay of general state space reduction methods and Petri net theory - stubborn set methods, - symmetries, - the state equation and Petri net invariants, - siphons and traps, - net reduction, - formula rewriting.	 **Elvio Amparore**  **Susanna Donatelli**	**Performance Evaluation and (Stochastic) Verification of (Stochastic) Petri Nets (14-18)** Model-based validation and evaluation of systems using Stochastic Petri nets (SPN): - SPN as a stochastic extension of Petri nets with priorities and inhibitor arcs - model-based evaluation (model construction; qualitative properties; quantitative properties) - compositional techniques - coloured Petri nets (specification of systems where multiple components have similar behaviour) - structural analysis (P- and T-semiflows, bounds, siphons and traps) - model checking (LTL, CTL and CTL* properties) - decision diagrams - Continuous Time Markov Chains (CTMC) - stochastic logic CSLTA - non-exponential delays for transitions - experimentation using the tool GreatSPN

Friday, September 8th

Kamila Barylska

Analysis and Synthesis of Some Subclasses of Petri Nets (10-13)

Choice-free nets:
- Implementation
- Pre-synthesis
- Proper synthesis
- Simultaneous proper synthesis

(Weighted) Marked Graphs

Divide and Conquer:
- Products and Sums
- Articulations
- Mixed Decomposition

Marco Montali

Data-Aware Processes: Modelling and Verification (10-13)

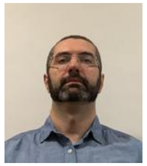
The need of combining static (i.e., data-related) and dynamic (i.e., process-related) aspects has been increasingly recognized as a key milestone towards the design, verification, and understanding of business and work processes.

Models for data-aware processes:
- formal modeling and analysis
- the usage of bounded Petri nets to account for the process control-flow
- possibility to handle verification tasks such as soundness and temporal model checking

Karsten Wolf

Efficient Verification of Petri Net Models (14-18)

The verification of Petri nets in a hands-on session:
- Petri net models and queries from the Model Checking Contest
- the performance of several verification techniques
- interplay of general state space reduction methods and Petri net theory
- stubborn set methods,
- symmetries,
- the state equation and Petri net invariants,
- siphons and traps,
- net reduction,
- formula rewriting.



Elvio Amparore

Susanna Donatelli

Performance Evaluation and (Stochastic) Verification of (Stochastic) Petri Nets (14-18)

Model-based validation and evaluation of systems using Stochastic Petri nets (SPN):
- SPN as a stochastic extension of Petri nets with priorities and inhibitor arcs
- model-based evaluation (model construction; qualitative properties; quantitative properties)
- compositional techniques
- coloured Petri nets (specification of systems where multiple components have similar behaviour)
- structural analysis (P- and T-semiflows, bounds, siphons and traps)
- model checking (LTL, CTL and CTL* properties)
- decision diagrams
- Continuous Time Markov Chains (CTMC)
- stochastic logic CSLTA
- non-exponential delays for transitions
- experimentation using the tool GreatSPN