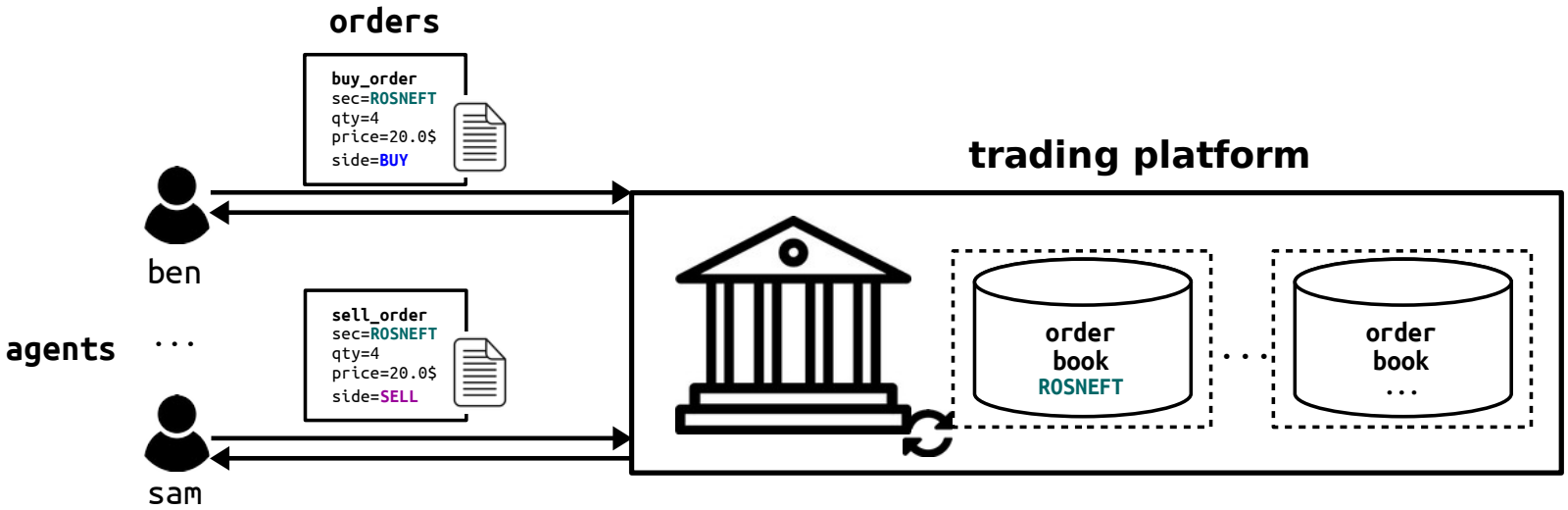
Formal Modeling and Validation of Stock Trading Systems Behavior: A Petri Net Approach

Julio C. Carrasquel
jcarrasquel@hse.ru

National Research University Higher School of Economics
Faculty of Computer Science

Trading Systems

A distributed system that consists of a software platform where agents submit *orders* to trade *securities*[1] (e.g., company shares).



domain experts seek for novel approaches for *validation* of trading systems, and that can detect *deviations*, i.e., differences between a real system and its specification.

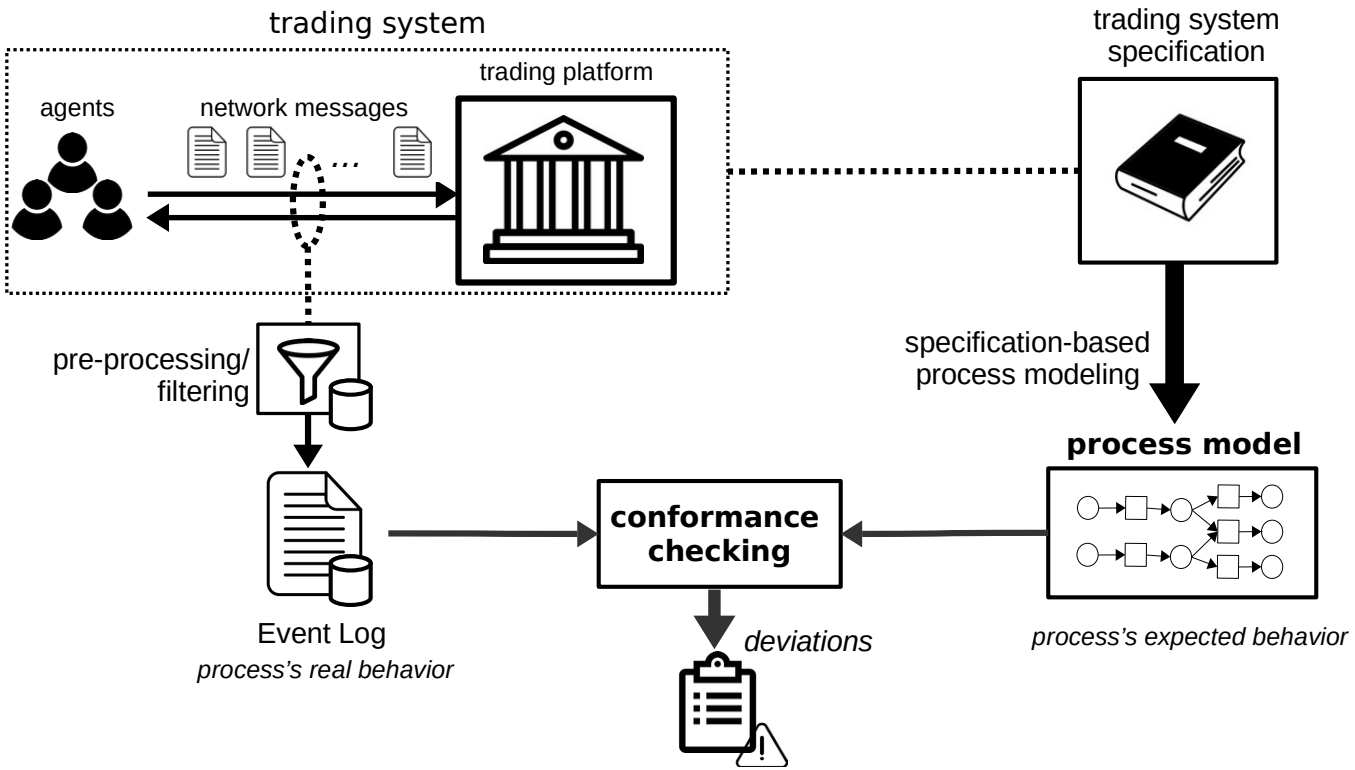
[1] Harris, L.: Trading and Exchanges: Market Microstructure for Practitioners. Oxford University Press (2003)

Validation encompasses different *model*-driven techniques for analysis of distributed systems.

- Simulation & Performance Analysis: run a system model and analyze key performance indicators (time, costs, etc).
- Verification (Model Checking)[1]: check whether a system model meets a given specification.
- Process Mining[2]: relate *real* behavior of a system (observed in event logs) with modeled behavior (built in process models).
- ...

[1] Baier, Christel, and Joost-Pieter Katoen. Principles of model checking. MIT press. (2008)

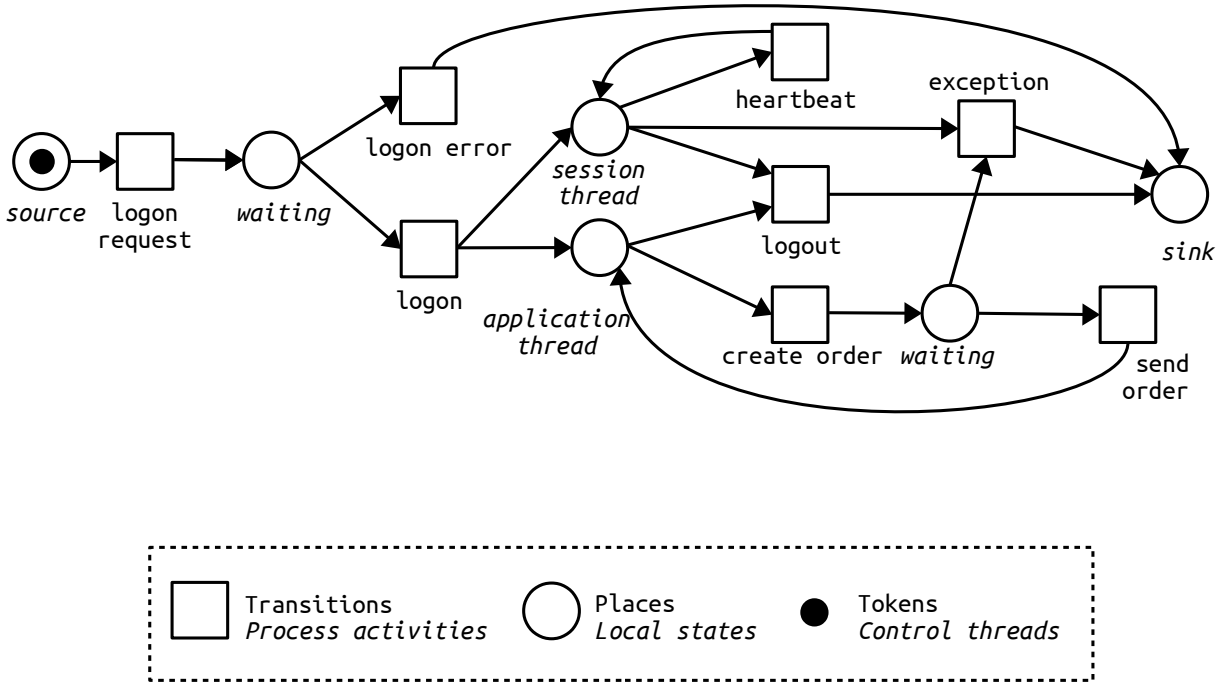
[2] van der Aalst, W.: Process Mining: Data Science in Action. Springer, 2nd edn. (2016)



Conformance checking[1] — process mining methods that compare *real behavior* in events logs with *expected behavior* described in a process model.

[1] Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking: Relating Processes and Models. Springer (2018)

Petri nets [1] are chosen for modeling processes in trading systems.



Focus on *control-flow* — activities and their causal dependence.

[1] Murata, T.: Petri nets: Properties, Analysis and Applications. Proceedings of the IEEE 77(4), 541580 (1989)

There are several *extensions* of Petri nets that come to enrinch the base formalisms with concepts such as data types, functions, modules, hierarchy, etc.

- Petri nets with Identifiers.[1]
- Colored Petri Nets [2]
- Nested Petri Nets [3]
- DB-nets [4]
- ...

[1] Hee, K., Sidorova, N., Voorhoeve, M., van der Werf, J.: Generation of Database Transactions with Petri Nets. Fundamenta Informaticae 93, 171184 (2009)

[2] Jensen, K., Kristensen, L.M. Coloured Petri Nets: Modelling and Validation of Concurrent Systems (2009)

[3] Lomazova, I.A.: Nested Petri Nets - a Formalism for Specification and Verification of Multi-Agent Distributed Systems. Fundamenta Informaticae 43, 195214 (2000)

[4] Montali, M., Rivkin, A.: DB-Nets: On the Marriage of Colored Petri Nets and Relational Databases. Transactions on Petri Nets and Other Models of Concurrency XII. LNCS, vol. 10470, pp. 91118. Springer (2017)

Research Problem

How to properly model processes in complex distributed systems such as trading systems using Petri nets, and how to employ models for validating real behavior in such systems.

Research Problem

How to properly model processes in complex distributed systems such as trading systems using Petri nets, and how to employ models for validating real behavior in such systems.

Research Aim

Define models based on Petri net extensions for properly modeling different aspects of processes in distributed systems, such as trading systems, and which can be compared against real behavior in such systems.

Main Results of This Thesis

1. Models based on *Petri net extensions* are defined for **formal modeling** certain aspects of processes in complex distributed systems such as trading systems. The considered aspects are: the interaction of individual objects, their autonomous behavior, and the transformation of their data attributes.

2. *New* conformance checking methods for **system validation** have been developed. The methods seek for concrete deviations in processes of distributed systems by comparing the proposed models with different event logs from the system.

3. Prototypical implementations of the conformance methods were developed, and their **experimental evaluation** for validating real systems are reported in this thesis. Experimental evaluations with artificial data are also reported.

A proposed model and a conformance method are chosen depending on the process and the aspects to validate.

Main Results of This Thesis

A Petri net extension model and a conformance method is chosen according to the process and the aspects to validate.

Example Process	**aspects to model & validate**	**Petri net extension** (Chapter 4)	**Event log** (Chapter 5)	**Conformance checking methods** (Chapter 6)
management of orders	interaction of *multiple* objects	Petri nets with identifiers (*PNI*s) (Section 4.1)	Event logs with object identifiers (Section 5.1)	Trace replay of an event log with object identifiers on a *PNI* model (Section 6.1)
	interaction of *multiple* objects and transformation of their *data* attributes	Colored Petri nets (CPNs) (Section 4.2)	Event logs with object data attributes (Section 5.2)	Trace replay of an event log with object data attributes on a CPN model (Section 6.2)
management of agents	interaction of *multiple* agents and their autonomous behavior	Nested Petri nets (NP-nets) (Section 4.3)	Event logs of multi-agent systems (Section 5.3)	Trace replay of an event log of a multi-agent system on a NP-net model (Section 6.3)

Related Work

1. *Passive analysis* make use of data science methods to analyze real behavior of distributed systems, such as trading systems, on the basis of logs. Yet, process mining has not been considered for trading systems.
   - Itkin, I., Yavorsky, R.. Overview of Applications of Passive Testing Techniques, MACSPro 2019, CEUR vol. 2478 (2019).
   - Itkin, I. et al. User-Assisted Log Analysis for Quality Control of Distributed Fintech Applications, IEEE AITest 2019, 45-51 (2019).

2. There a few conformance checking methods using extensions of Petri nets (e.g., Petri nets with data). However, there are no conformance checking methods for the chosen extensions in this work.
   - de Leoni, M., Munoz-Gama, J., Carmona, J., van der Aalst, W. Decomposing conformance checking on Petri nets with data. BPM center report BPM-14-06 (2014).

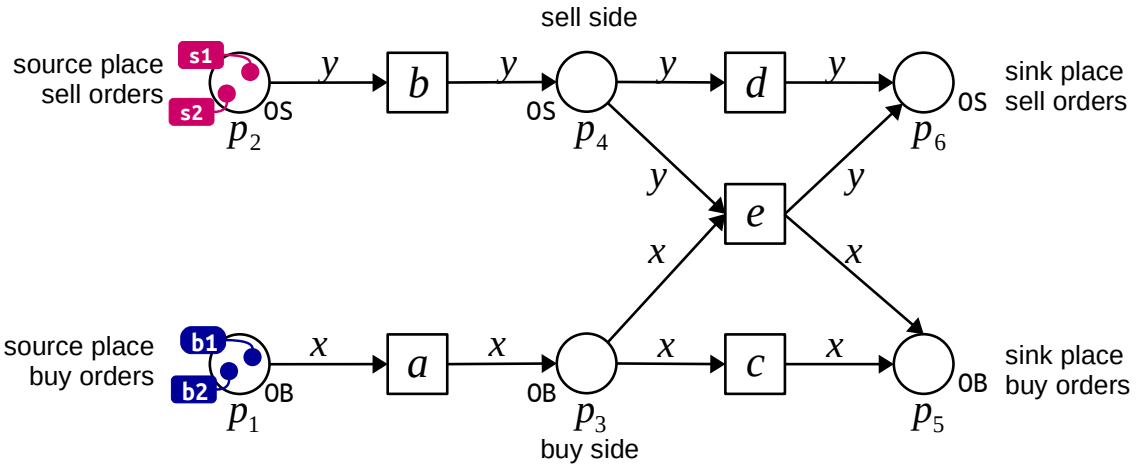
3. Use of object-centric process models is an active research direction in process mining.
   - van der Aalst, W. , Berti. A. Discovering object-centric Petri nets. Fundamenta Informaticae, vol. 175, no 1-4, p. 1-40. (2020).

A Petri net extension and a conformance method is chosen according to the process and the aspects to validate.

Example Process	**aspects to model & validate**	**Petri net extension** (Chapter 4)	**Event log** (Chapter 5)	**Conformance checking methods** (Chapter 6)
management of orders	interaction of *multiple* objects	Petri nets with identifiers (*PNI*s) (Section 4.1)	Event logs with object identifiers (Section 5.1)	Trace replay of an event log with object identifiers on a *PNI* model (Section 6.1)
	interaction of *multiple* objects and transformation of their *data* attributes	Colored Petri nets (CPNs) (Section 4.2)	Event logs with object data attributes (Section 5.2)	Trace replay of an event log with object data attributes on a CPN model (Section 6.2)
management of agents	interaction of *multiple* agents and their autonomous behavior	Nested Petri nets (NP-nets) (Section 4.3)	Event logs of multi-agent systems (Section 5.3)	Trace replay of an event log of a multi-agent system on a NP-net model (Section 6.3)

I. Formal Models
&
II. Validation Methods

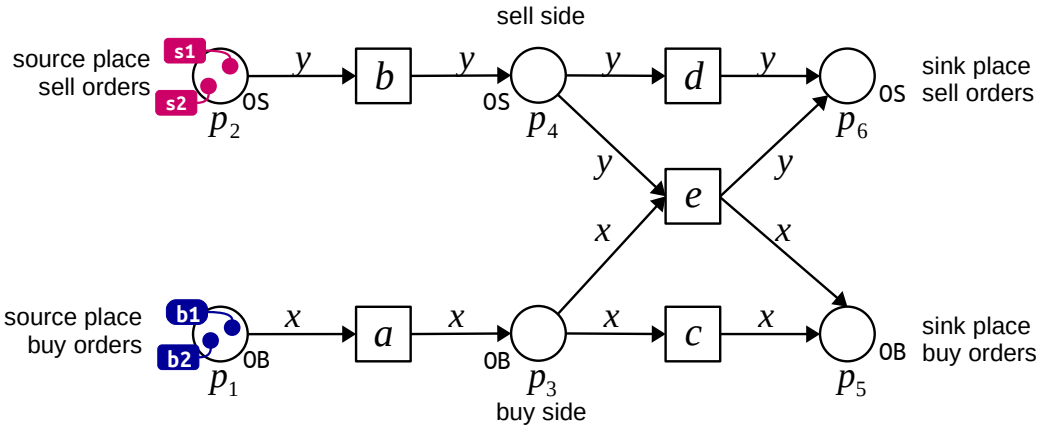
Petri nets with Identifiers (*PNI*s): tokens abstractly represent individual objects.



a	`new buy order`
b	`new sell order`
c	`cancel buy order`
d	`cancel sell order`
e	`trade`

Modeling interaction of orders

Petri nets with Identifiers (*PNI*s): tokens abstractly represent individual objects.



a	new buy order
b	new sell order
c	cancel buy order
d	cancel sell order
e	trade

Restrictions are imposed in the models; we call the models that comply such restrictions as **conservative workflow nets**.

Conservative: Tokens cannot duplicate or disappear.

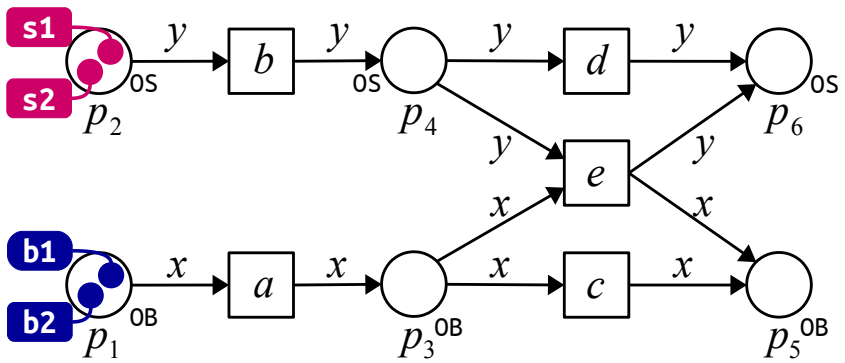
Workflow:
— For every type, there is a pair of distinguished source and sink places.
— For each source-sink pair (i, o), exists a (i, o)-path s.t. places in-between are of the same type.

A trace σ of an *event log with object identifiers* is **replayed** on a *PNI* model.

token jumps j = 0	token transfers k = 0

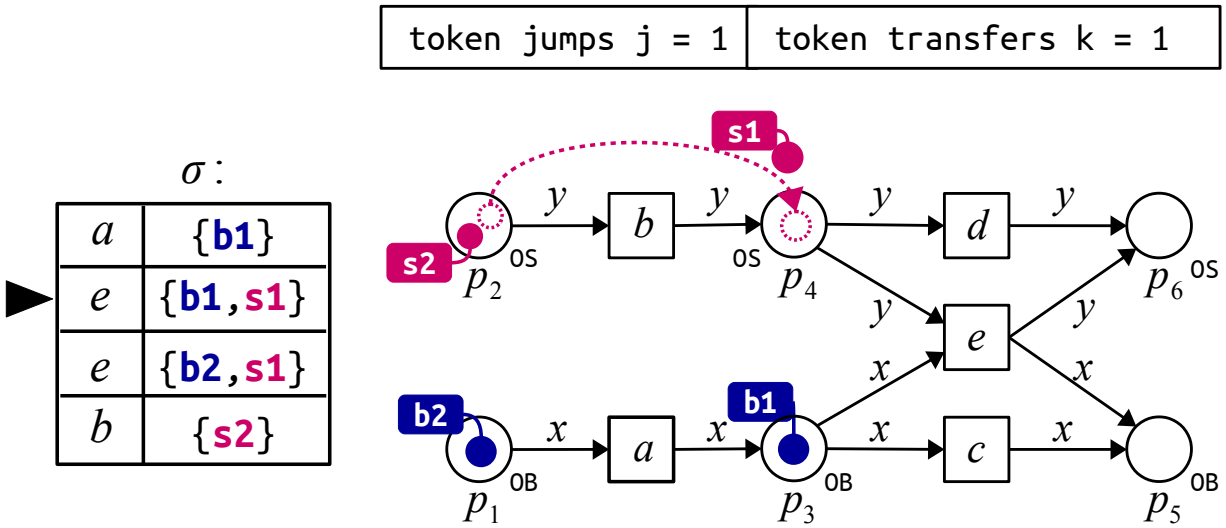
σ :

a	{b1}
e	{b1,s1}
e	{b2,s1}
b	{s2}



Replay — execute the model according to each event of the trace σ

A trace σ of an *event log with object identifiers* is **replayed** on a *PNI* model.



Replay — execute the model according to each event of the trace σ

A **deviation** is detected when the model cannot be executed as an event specifies.

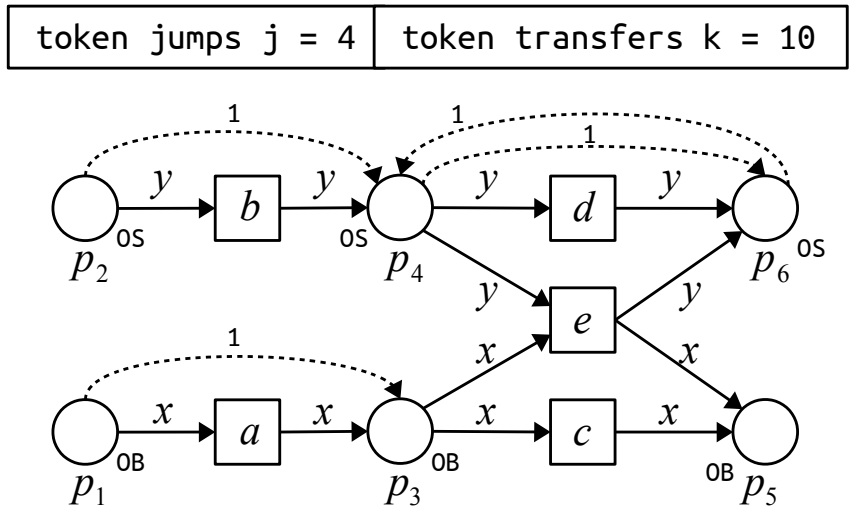
Token jumps are used to force the model execution.

A **fitness metric** is defined to measure compliance of a process (as observed in a trace σ) with a *PNI* model.

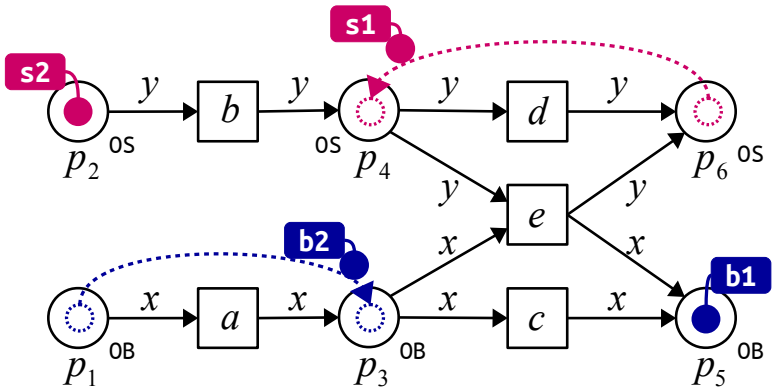
$$\texttt{fit}(\sigma, PNI) = 1 - \frac{\texttt{j}}{\texttt{k}} = 1 - \frac{4}{10} = 0.6$$

σ :

a	{**b1**}
e	{**b1**,**s1**}
e	{**b2**,**s1**}
b	{**s2**}

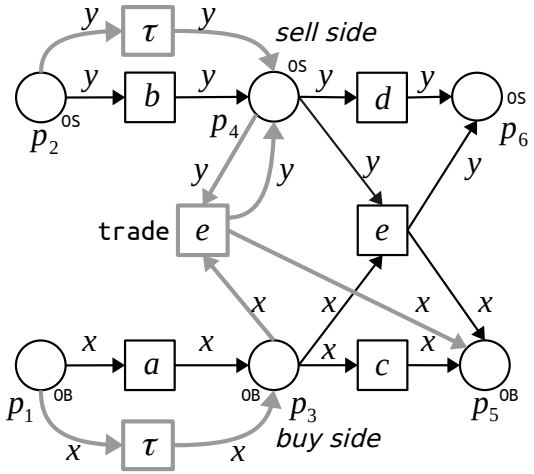


Token jumps can be seen as *desire lines*[*], i.e., actual paths taken by objects that violate the model.

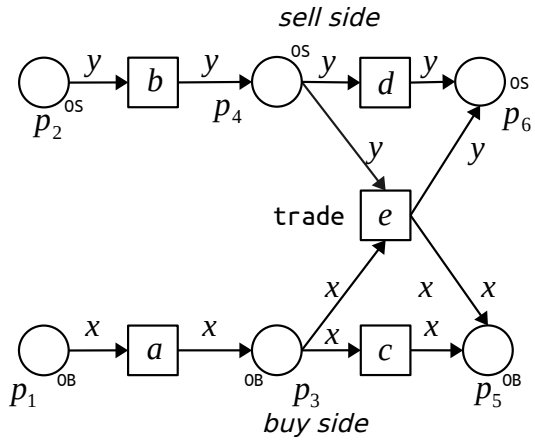


[*] W. van der Aalst, *Desire Lines in Big Data*, Encyclopedia of Social Network Analysis & Mining, pp. 351-364, 2014.

Local conformance measures are defined to diagnose deviations in specific components of a process.

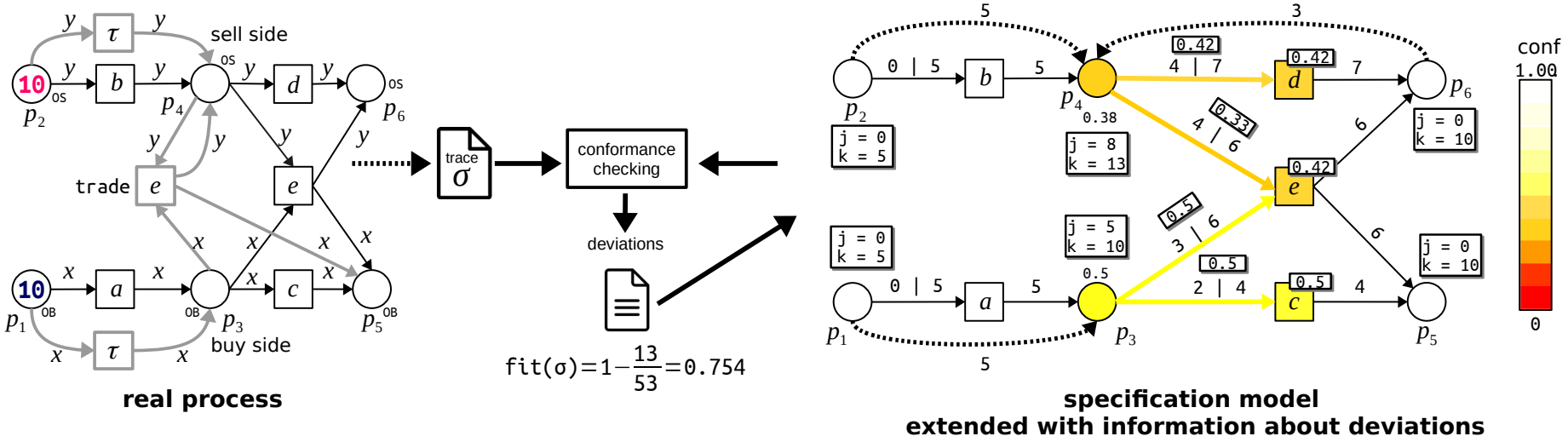


real process



specification model

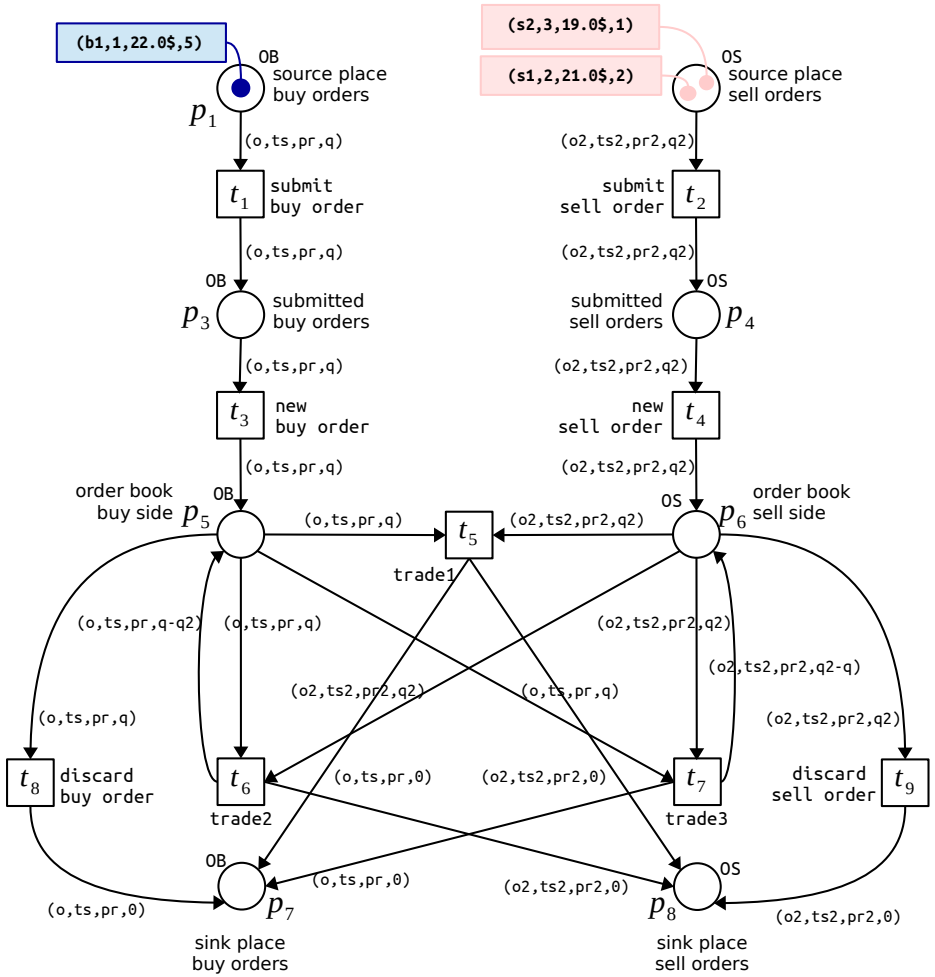
Local conformance measures are defined to diagnose deviations in specific components of a process.



A Petri net extension and a conformance method is chosen according to the process and the aspects to validate.

Example Process	**aspects to model & validate**	**Petri net extension** (Chapter 4)	**Event log** (Chapter 5)	**Conformance checking methods** (Chapter 6)
management of orders	interaction of *multiple* objects	Petri nets with identifiers (*PNI*s) (Section 4.1)	Event logs with object identifiers (Section 5.1)	Trace replay of an event log with object identifiers on a *PNI* model (Section 6.1)
	interaction of *multiple* objects and transformation of their *data* attributes	Colored Petri nets (CPNs) (Section 4.2)	Event logs with object data attributes (Section 5.2)	Trace replay of an event log with object data attributes on a CPN model (Section 6.2)
management of agents	interaction of *multiple* agents and their autonomous behavior	Nested Petri nets (NP-nets) (Section 4.3)	Event logs of multi-agent systems (Section 5.3)	Trace replay of an event log of a multi-agent system on a NP-net model (Section 6.3)

Colored Petri nets (CPNs) model interaction of objects and how their **data attributes** are transformed.

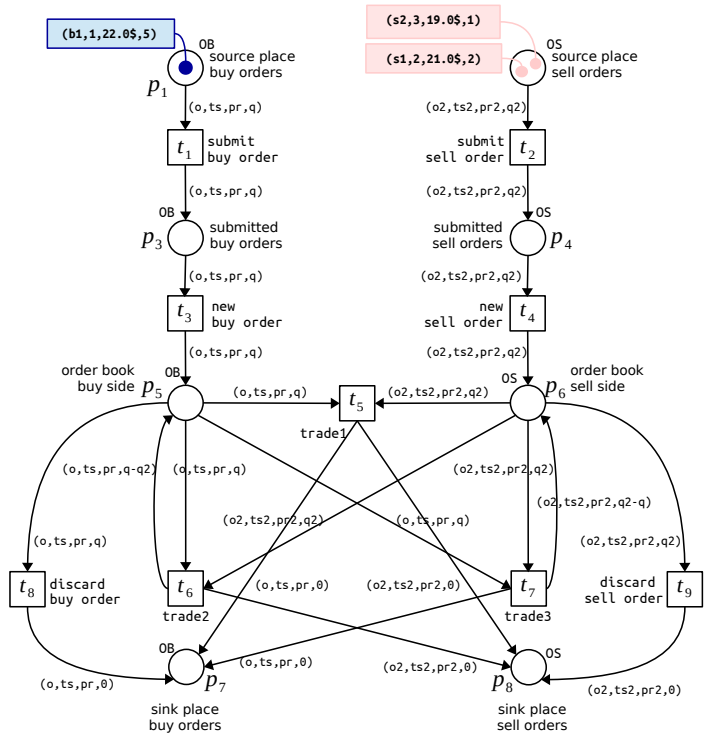


Jensen, K., Kristensen, L.M. Coloured Petri Nets: Modelling and Validation of Concurrent Systems (2009)

A trace σ of an *event log with object data attributes* is **replayed** on a *CPN* model.

event (e)	activity (a)	objects $(R(e))$
e_1	submit buy order	(b1, 1, 22.0, 5)
e_2	new buy order	(b1, 1, 22.0, 5)
e_3	submit sell order	(s1, 2, 21.0, 2)
e_4	new sell order	(s1, 2, 21.0, 2)
e_5	new sell order	(s2, 3, 19.0, 1)
e_6	trade2	(b1, 1, 22.0, 4), (s1, 2, 21.0, 0)

trace of an event log with object data attributes



CPN model

Different kinds of deviations can be detected.

A trace σ of an *event log with object data attributes* is **replayed** on a *CPN* model.

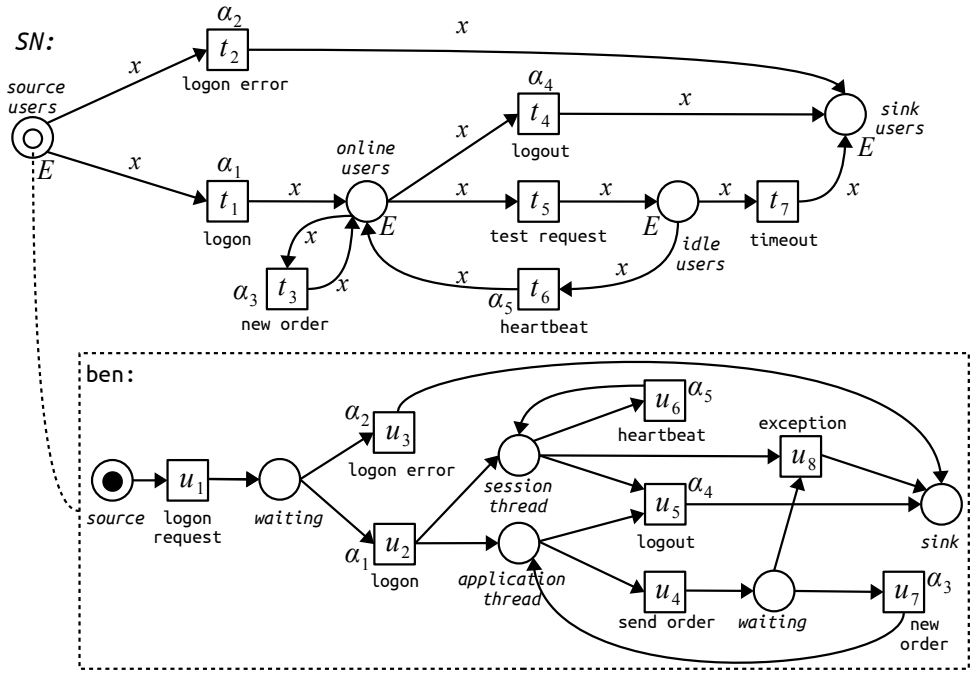
Deviations that can be detected during the replay:

1. **Control-Flow Deviation:** An activity is executed processing a given object, but illegally skipping activities that must have processed such an object before.
2. **Priority Rule Violation:** An object with less priority has been processed before an object with higher prority.
3. **Resource Corruption:** The real process (as observed in an event) has transformed attributes of an object in a different way than the model specifies.
4. **Non-proper termination:** Objects not fully processed by the process at the end of a run.

A Petri net extension and a conformance method is chosen according to the process and the aspects to validate.

Example Process	**aspects to model & validate**	**Petri net extension** (Chapter 4)	**Event log** (Chapter 5)	**Conformance checking methods** (Chapter 6)
management of orders	interaction of *multiple* objects	Petri nets with identifiers (*PNI*s) (Section 4.1)	Event logs with object identifiers (Section 5.1)	Trace replay of an event log with object identifiers on a *PNI* model (Section 6.1)
	interaction of *multiple* objects and transformation of their *data* attributes	Colored Petri nets (CPNs) (Section 4.2)	Event logs with object data attributes (Section 5.2)	Trace replay of an event log with object data attributes on a CPN model (Section 6.2)
management of agents	interaction of *multiple* agents and their autonomous behavior	Nested Petri nets (NP-nets) (Section 4.3)	Event logs of multi-agent systems (Section 5.3)	Trace replay of an event log of a multi-agent system on a NP-net model (Section 6.3)

Nested Petri nets (NP-nets) consists of a “system net" *SN* where “net tokens" (agents) have inner Petri nets that model individual agent behavior.



Modeling interaction of agents with a trading platform according to the session-layer of the FIX protocol.

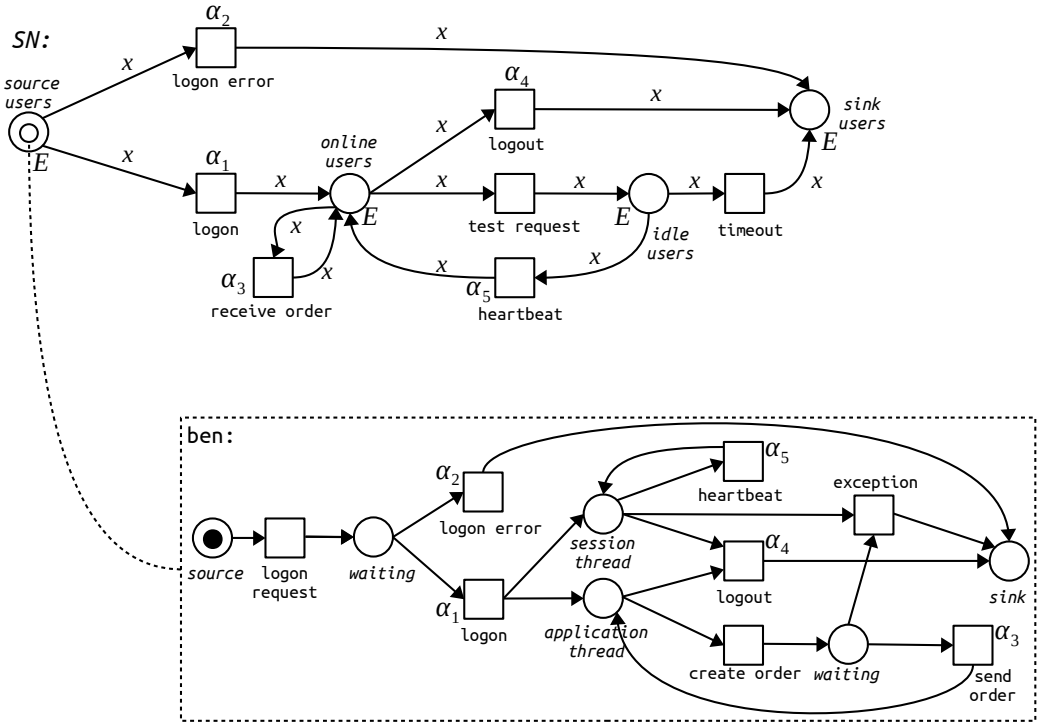
Lomazova, I.A.: Nested Petri Nets - a Formalism for Specification and Verification of Multi-Agent Distributed Systems. Fundamenta Informaticae 43, 195214 (2000)

A trace of an *event log of a multi-agent system* is **replayed** on a NP-net model.

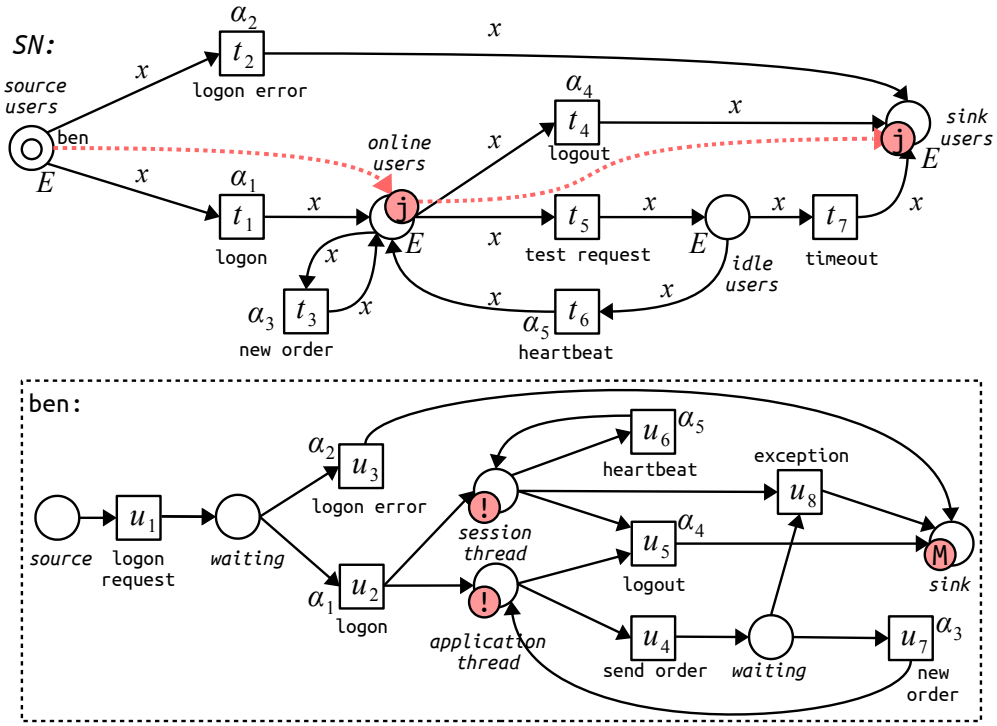
Trace of an event log of a multi-agent system

trace	event	
σ	e_1	(create order, ben)
	e_2	(receive order, {(send order, ben)})
	e_3	(test request, {ben})
	e_4	(heartbeat, {(heartbeat, ben)})

NP-net model



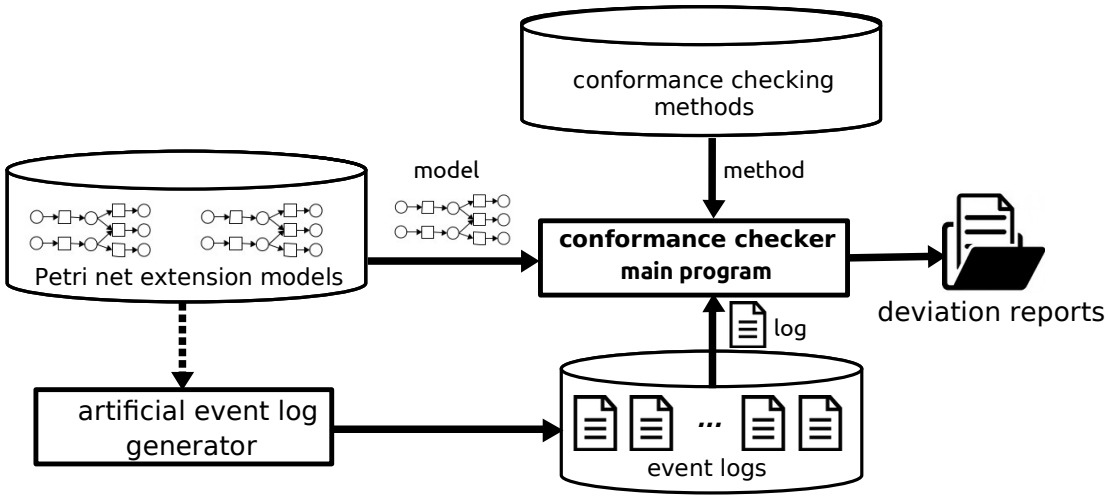
A trace of an *event log of a multi-agent system* is **replayed** on a NP-net model.



Deviations detected can be related to the system or to precise agents.

III. Experimental Evaluation

A set of command-line tools were developed in Python, and which implement the presented models and the validation methods.



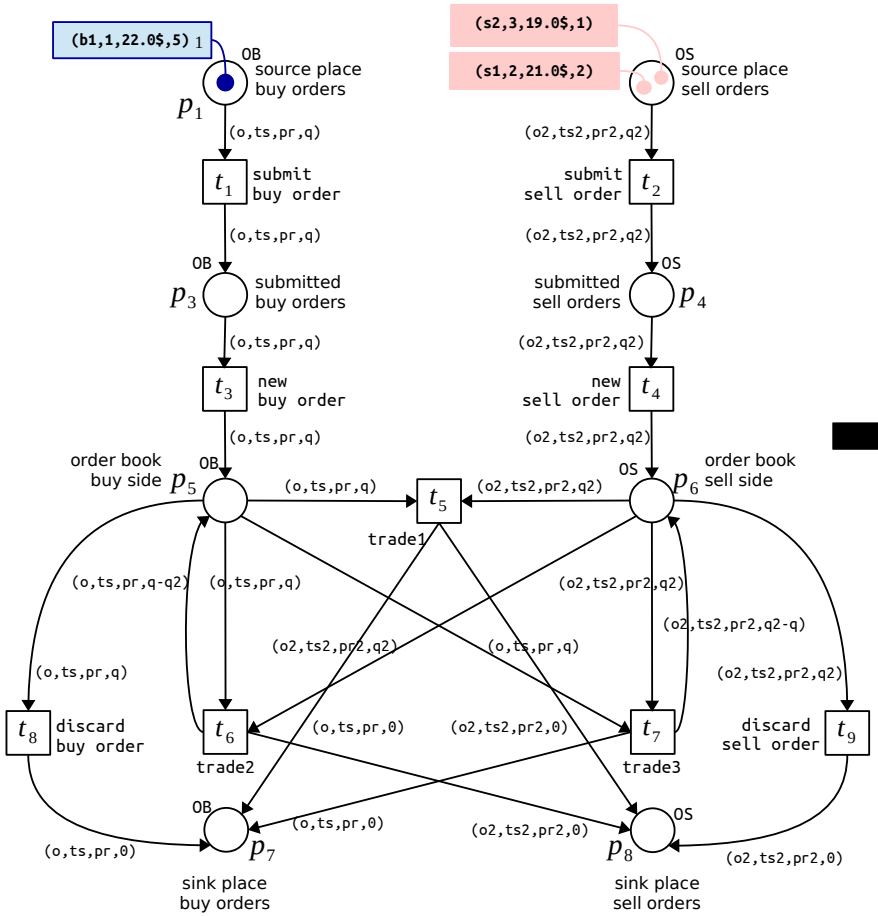
Organization of the project repository at:

https://github.com/jcarrasquel/hse-uamc-conformance-checking

command-line execution example:

```
python3 conformance_checker.py 1 model.py event_log.csv
```

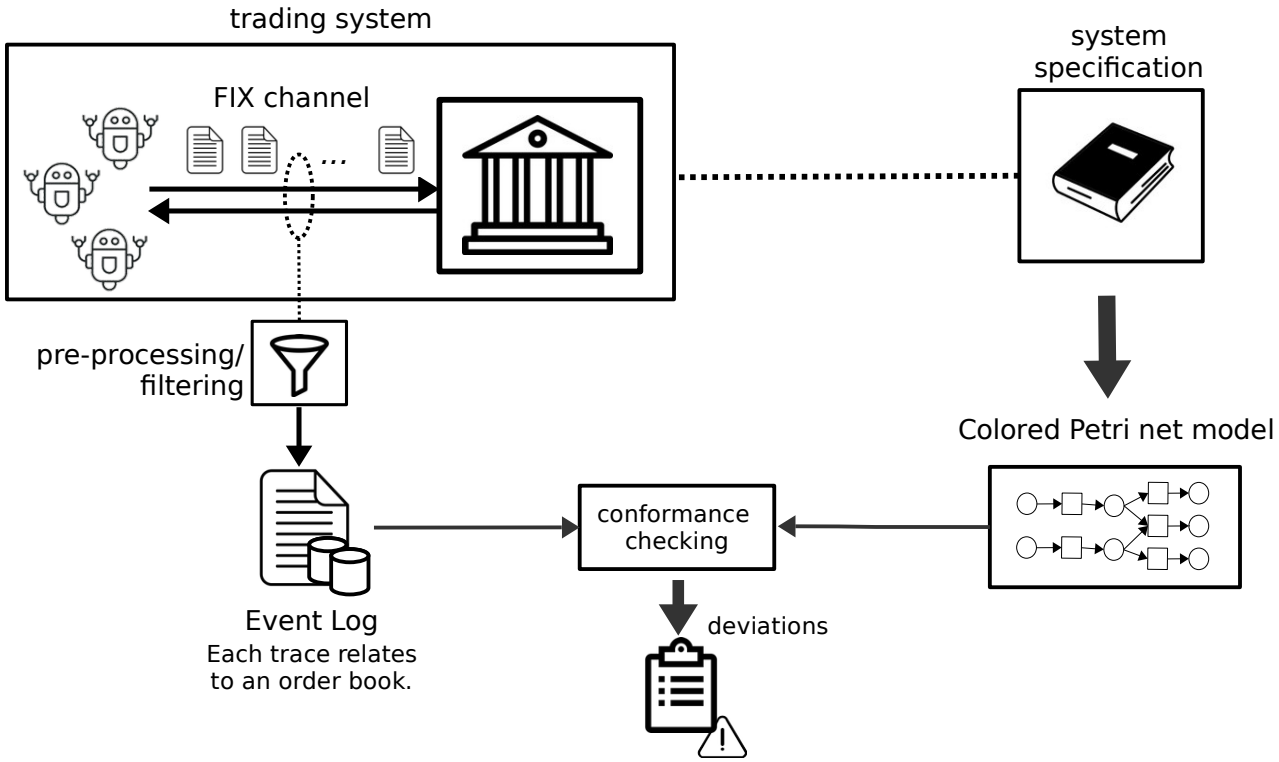
Models of Petri net extensions are built as Python scripts using SNAKES[1] — a Python-powered library for construction and simulation of high-level Petri nets.




```
def buildPetriNet():

    TYPE_A = TypeCheck(tTypeAId)
    TYPE_B = TypeCheck(tTypeBId)

    petriNetAttributes = {};
    petriNetAttributes["COLOR_TYPES"] = [TYPE_A, TYPE_B];

    petriNet = PetriNet('mais-1')

    if generateArtificialResources == True:
        generateResources()
        petriNet.add_place(Place("p1", resourcesClassA, TYPE_A))
        petriNet.add_place(Place("p2", resourcesClassB, TYPE_B))
    else:
        petriNet.add_place(Place("p1", [], TYPE_A))
        petriNet.add_place(Place("p2", [], TYPE_B))

    petriNetAttributes["INITIAL_PLACES"] = {TYPE_A: "p1", TYPE_B: "p2"}
    petriNetAttributes["FINAL_PLACES"] = {TYPE_A: "p5", TYPE_B : "p6"}

    petriNet.add_place(Place("p3", [], TYPE_A))
    petriNet.add_place(Place("p4", [], TYPE_B))
    petriNet.add_place(Place("p5", [], TYPE_A))
    petriNet.add_place(Place("p6", [], TYPE_B))

    activityLabels = ["a", "b", "c", "d", "e"]
    for i in range(5):
        t = Transition("t" + str(i+1))
        petriNet.add_transition(t)
        t.label(activity=activityLabels[i])

    petriNet.add_input("p1", "t1", Variable("x"))
    petriNet.add_output("p3", "t1", Variable("x"))

    petriNet.add_input("p2", "t2", Variable("y"))
    petriNet.add_output("p4", "t2", Variable("y"))
```

[1] https://snakes.ibisc.univ-evry.fr/

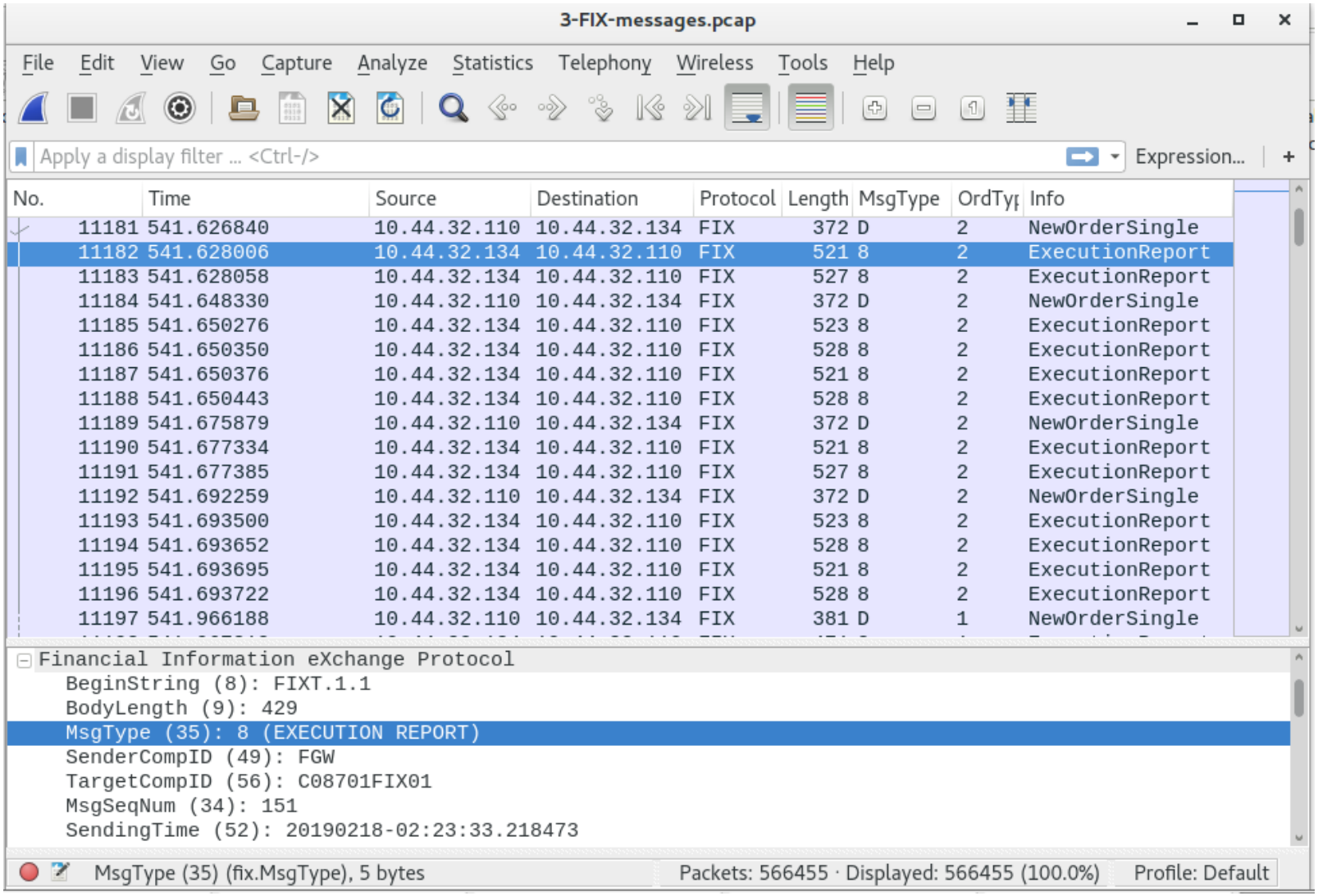
We validated the management of orders in several order books of a real trading platform.



FIX Messages = 552935
Selected Order books (Traces) = 73
Events = 2259
Avg. Events per Trace = 30.94

An *event log with object data attributes* is extracted from a set of Financial Information Exchange (FIX) protocol messages

FIX messages encapsulate activities executed by agents and the trading platform...



The event log is formated as a Comma-Separated-Values (CSV) file.

	A	B	C	D	E	F	G	H	I	J	K
1	CASE	TIMESTAMP	ACTIVITY	ID1	TSUB1	PRICE1	QTY1	ID2	TSUB2	PRICE2	QTY2
2	5169112	18-02-2019T03:00:23.804	submit buy order	00d0Phn2Svpv	18-02-2019T03:00:23.804000	105	100				
3	5169112	18-02-2019T03:00:23.805	new buy order	00d0Phn2Svpv	18-02-2019T03:00:23.804000	105	100				
4	5169112	18-02-2019T03:00:23.817	submit sell order	00d0Phn2Svpw	18-02-2019T03:00:23.817000	105	100				
5	5169112	18-02-2019T03:00:23.818	new sell order	00d0Phn2Svpw	18-02-2019T03:00:23.817000	105	100				
6	5169112	18-02-2019T03:01:24.280	trade1	00d0Phn2Svpw	18-02-2019T03:00:23.817000	105	0	00d0Phn2Svpv	18-02-2019T03:00:23.804000	105	0
7	5169113	18-02-2019T03:01:32.697	submit buy order	00d0Phn2SwNq	18-02-2019T03:01:32.697000	105	100				
8	5169113	18-02-2019T03:01:32.698	new buy order	00d0Phn2SwNq	18-02-2019T03:01:32.697000	105	100				
9	5169113	18-02-2019T03:01:32.708	submit sell order	00d0Phn2SwNr	18-02-2019T03:01:32.708000	105	100				
10	5169113	18-02-2019T03:01:32.709	new sell order	00d0Phn2SwNr	18-02-2019T03:01:32.708000	105	100				
11	5169113	18-02-2019T03:02:34.679	trade1	00d0Phn2SwNq	18-02-2019T03:01:32.697000	105	0	00d0Phn2SwNr	18-02-2019T03:01:32.708000	105	0
12	201701021	18-02-2019T05:05:44.541	submit buy order	00d0Phn2SzMt	18-02-2019T05:05:44.541000	12	10				
13	201701021	18-02-2019T05:05:44.544	new buy order	00d0Phn2SzMt	18-02-2019T05:05:44.541000	12	10				
14	201701021	18-02-2019T05:05:44.554	submit sell order	00d0Phn2SzMu	18-02-2019T05:05:44.554000	12	10				
15	201701021	18-02-2019T05:05:44.554	new sell order	00d0Phn2SzMu	18-02-2019T05:05:44.554000	12	10				
16	201701021	18-02-2019T05:05:44.555	trade1	00d0Phn2SzMu	18-02-2019T05:05:44.554000	12	0	00d0Phn2SzMt	18-02-2019T05:05:44.541000	12	0
17	201701021	18-02-2019T05:06:24.636	submit buy order	00d0Phn2SzPq	18-02-2019T05:06:24.636000	10	2000				
18	201701021	18-02-2019T05:06:24.642	new buy order	00d0Phn2SzPq	18-02-2019T05:06:24.636000	10	2000				
19	201701021	18-02-2019T05:06:24.650	submit buy order	00d0Phn2SzPr	18-02-2019T05:06:24.650000	10	2000				
20	201701021	18-02-2019T05:06:24.654	new buy order	00d0Phn2SzPr	18-02-2019T05:06:24.650000	10	2000				
21	201701021	18-02-2019T05:06:24.660	submit buy order	00d0Phn2SzPs	18-02-2019T05:06:24.660000	10	2000				
22	201701021	18-02-2019T05:06:24.663	new buy order	00d0Phn2SzPs	18-02-2019T05:06:24.660000	10	2000				

Each trace corresponds to events occurred in an individual *order book*.

Deviations Detected

TRACE	EVENT	TIMESTAMP	ACTIVITY	OBJECT	DEV.	DEVIATION DESCRIPTION
1488058	1781	05:52:58.18	trade2	bSovX	RC	resource has event-state: ('b00d0PhqYSovX' 1550491266 100.0 100) ,but model-state is: ('b00d0PhqYSovX' 1550491266 105.0 100)
1488058	1782	05:52:58.18	trade1	bSovX	RC	resource has event-state: ('b00d0PhqYSovX' 1550491266 101.0 0) ,but model-state is: ('b00d0PhqYSovX' 1550491266 100.0 0)
1488061	1792	05:53:23.38	trade1	sSowK	RV	resource with id: s00d0PhqYSowK did not have priority ,over other resources in the same place.
1488061	1792	05:53:23.38	trade1	sSowK	RC	resource has event-state: ('s00d0PhqYSowK' 1550490938 101.0 0) ,but model-state is: ('s00d0PhqYSowK' 1550490938 101.0 -100)
1488061	1792	05:53:23.38	trade1	bSowJ	RC	resource has event-state: ('b00d0PhqYSowJ' 1550490919 101.0 0) ,but model-state is: ('b00d0PhqYSowJ' 1550490919 105.0 100)
1488061	1793	05:53:23.38	trade2	bSowJ	CF	resource with id: b00d0PhqYSowJ was not in location p5 but in p7
1488061	1793	05:53:23.38	trade2	bSowJ	RC	resource has event-state: ('b00d0PhqYSowJ' 1550490919 105.0 100) ,but model-state is: ('b00d0PhqYSowJ' 1550490919 101.0 -100)
1488061	1793	05:53:23.38	trade2	sSowL	RC	resource has event-state: ('s00d0PhqYSowL' 1550490947 105.0 0) ,but model-state is: ('s00d0PhqYSowL' 1550490947 100.0 0)
1488061	end	-	-	bSowJ	NT	resource with id: b00d0PhqYSowJ was not in final location p7 but in p5
1488062	1803	05:53:31.38	trade2	bSowN	RC	resource has event-state: ('b00d0PhqYSowN' 1550490899 100.0 100) ,but model-state is: ('b00d0PhqYSowN' 1550490899 105.0 100)
1488062	1804	05:53:31.38	trade1	bSowN	RC	resource has event-state: ('b00d0PhqYSowN' 1550490899 101.0 0) ,but model-state is: ('b00d0PhqYSowN' 1550490899 100.0 0)
9088012	end	-	-	bmkq9	NT	resource with id: b00d0PiS3mkq9 was not in final location p7 but in p5
9088012	end	-	-	smkqA	NT	resource with id: s00d0PiS3mkqA was not in final location p8 but in p6
9088015	end	-	-	sSSZd	NT	resource with id: s00d0Pi88SSZd was not in final location p8 but in p6

RC: Resource (data attributes) corrupted. RV: Priority Rule violation.
CF: Control-flow deviation. NT: Non-proper termination.

Concluding Remarks

Main Results of This Thesis

1. Models based on *Petri net extensions* are defined for **formal modeling** certain aspects of processes in complex distributed systems such as trading systems. The considered aspects are: the interaction of individual objects, their autonomous behavior, and the transformation of their data attributes.

2. *New* conformance checking methods for **system validation** have been developed. The methods seek for concrete deviations in processes of distributed systems by comparing the proposed models with different event logs from the system.

3. Prototypical implementations of the conformance methods were developed, and their **experimental evaluation** for validating real systems are reported in this thesis. Experimental evaluations with artificial data are also reported.

Reserch Publications

1. Carrasquel, J.C., Mecheraoui, K. Object-Centric Replay-Based Conformance Checking: Unveiling Desire Lines and Local Deviations. **Journal on Modeling and Analysis of Information Systems (MAIS)**, vol.28, 2021.

2. Carrasquel, J.C., Mecheraoui, K., Lomazova, I.A. Checking Conformance Between Colored Petri Nets and Event Logs. AIST 2020, **LNCS, Springer**, vol. 12602, 2021.

3. Carrasquel, J.C., Chuburov, S.A., Lomazova. Pre-processing Network Messages of Trading Systems into Event Logs for Process Mining. TMPA 2019, **CCIS, Springer**, vol. 1288, 2021.

4. Mecheraoui, K., Carrasquel, J.C., Lomazova, I.A. Compositional Conformance Checking of Nested Petri Nets and Event Logs of Multi-Agent Systems. MACSPro 2020, **CEUR**, vol. 2795, 2020.

5. Carrasquel, J.C., Lomazova, I.A, Rivkin A., Modeling Trading Systems using Petri Net Extensions. PNSE 2020, **CEUR**, vol. 2651, 2020.

6. Carrasquel, J.C., Lomazova, I.A. Modeling and Validation of Trading and Multi-Agent Systems: An Approach Based on Process Mining and Petri Nets. ICPM Doctoral Consortium, **CEUR**, vol. 2432, 2019.

7. Carrasquel, J.C., Lomazova, I.A., Itkin, I.L. Towards a Formal Modeling of Order-driven Trading Systems using Petri Nets: A Multi-Agent Approach. MACSPro 2019, **CEUR**, vol. 2478, 2019

Conference Presentations

1. 6th International Conference on Software Testing, Machine Learning and Complex Process Analysis (**TMPA 2021**). **Online**. 25-27 November 2021. Talk: Searching for Deviations in Trading Systems: Combining Control-Flow and Data Perspectives.

2. 12th Workshop on Program Semantics, Specification and Verification (**PSSV 2021**). **Online**. 4-5 November 2021. Talk: Validating Real Behavior of Agents in Trading Systems using Nested Petri Nets.

3. 2nd Conference on Modeling and Analysis of Complex Systems and Processes (**MACSPro 2020**). **Online**. 20-24 October 2020. Talk: Compositional Conformance Checking of Nested Petri Nets and Event Logs of Multi-Agent Systems.

4. 9th International Conference on Analysis of Images, Social Networks and Texts (**AIST 2020**). **Online**. 15-16 October 2020. Talk: Checking Conformance Between Colored Petri Nets and Event Logs.

5. International Workshop on Petri Nets and Software Engineering (**PNSE 2020**). **Online**. 24 June 2020. Talk: Modeling Trading Systems using Petri Net Extensions..

6. 5th International Conference on Software Testing, Machine Learning and Complex Process Analysis (**TMPA 2019**). **Tbilisi, Georgia**. 7-9 November 2019. Talk: Pre-processing Network Messages of Trading Systems into Event Logs for Process Mining.

7. Doctoral Consortium at the 1st International Conference on Process Mining (**ICPM 2019**). **Aachen,Germany**. 23 June 2019. Talk: Modelling and Validation of Trading and Multi-Agent Systems: An Approach Based on Process Mining and Petri Nets.

8. 1st Conference on Modeling and Analysis of Complex Systems and Processes (**MACSPro 2019**). **Vienna, Austria**. 21-23 March 2019. Talk: Towards a Formal Modelling of Order-driven Trading Systems using Petri Nets: A Multi-Agent Approach.

Possible Directions of Future Research

- Experimental evaluation with additional real-life data.
- Tool extension.
- Relaxation of model constraints.
- Use of Alignments.

Thank You!

Formal Modeling and Validation of
Stock Trading Systems Behavior: A Petri Net Approach

Julio C. Carrasquel
jcarrasquel@hse.ru

National Research University Higher School of Economics
Faculty of Computer Science

PhD Pre-Defense

May 2022

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ