

# A Plugin System for Charlie

Jan-Thierry Wegener<sup>1</sup>   Martin Schwarick<sup>2</sup>   Monika Heiner<sup>2</sup>

<sup>1</sup> Otto-von-Guericke-Universität  
Institute of Biology  
Universitätsplatz 2, 39106 Magdeburg, Germany

<sup>2</sup> Brandenburg University of Technology  
Department of Computer Science  
Postbox 10 13 44, 03013 Cottbus, Germany

CS&P 2011, Pułtusk, Poland



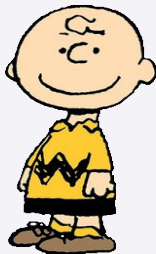
# Outline

- 1 Introduction
- 2 Charlie's Main Features
- 3 Requirements on Charlie's Plugin System
- 4 Charlie's Plugin System
- 5 Conclusion

# Charlie

Charlie is a tool for analyzing Petri nets that

- is written in Java;
- has an intuitive and easy to use GUI;
- is able to analyze several properties;
- is able to conclude further results from results (rule system).



# Charlie

Charlie is a tool for analyzing Petri nets that

- is written in Java;
- has an intuitive and easy to use GUI;
- is able to analyze several properties;
- is able to conclude further results from results (rule system).

Charlie is not a tool for

- drawing Petri nets;
- simulating Petri nets.



# Charlie

Charlie is not a tool for

- drawing Petri nets;
- simulating Petri nets.



# Charlie

Charlie is not a tool for

- drawing Petri nets;
- simulating Petri nets.

Charlie can read following file-formats:

- Snoopy;
- INA;
- APNN.

# History of Charlie

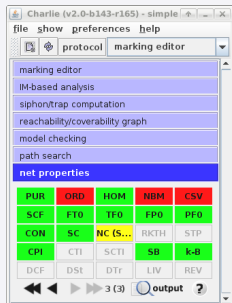
- 2003 last version of INA
- 2006 first version of Charlie (Master Thesis by Martin Schwarick)
- 2009 added reachability graph for time-dependent Petri nets (Diploma Thesis by Ansgar Fischer)
- 2009 new GUI and redesign of Charlie's code (Diploma Thesis by Andreas Franzske)
- 2011 integration of a plugin system (by Jan-Thierry Wegener)

# INA and Charlie

```

The net has a nonempty clean trap.
The net has no transitions without pre-place.
The net has no transitions without post-place.
The net is connected.
ORD HOM NBM PUR CSV SCF CON SC Ft0 tF0 Fp0 pF0 MG SM FC EFC ES
Y Y N Y N N Y N N N Y Y N N N N N
DTP SMC SMD SMA CPI CTI B SB REV Dst Bst DTr DCF L LV L&S
N ? N N ? N Y Y ? ? ? ? ? N ? N
Analysis menu:
Decide structural boundedness.....B
Non-reachability test of a partial marking using the state equation.....N
Compute the symmetries of the net.....Y

```





# INA and Charlie

```

The net has a nonempty clean trap.
The net has no transitions without pre-place.
The net has no transitions without post-place.
The net is connected.
ORD HOM NBM PUR CSV SCF CON SC  Ft0 tF0 Fp0 pF0 MG  SM  FC  EFC ES
Y  Y  N  Y  N  N  Y  N  N  N  Y  Y  N  N  N  N  N  N  N
DTP SMC SMD SMA CPI CTI  B  SB  REV DSt BSt DTr DCF  L  LV L&S
N  ?  N  N  ?  N  Y  Y  ?  ?  ?  ?  ?  ?  N  ?  N
Analysis menu:
Decide structural boundedness.....B
Non-reachability test of a partial marking using the state equation.....N
Compute the symmetries of the net.....Y

```

Charlie (v2.0-b158-r208) - travel. . . . . X

file show preferences help

protocol marking editor

Output of analysis

1 [ ] compute

2 [ ] [1..T\_1] :1. ice matrix

3 [ ] [2..T\_2] :1.

4 [ ] [3..T\_3] :1.

5 [ ] [4..T\_4] :1.

6 [ ] [5..T\_5] :1

7 [ ] [6..T\_6] :1 boundedness

8 [ ] [7..T\_7] :1

9 [ ] [8..T\_8] :1

10 [ ] [9..T\_9] :1

11 [ ] [10..T\_10] :1

12 [ ] [11..T\_11] :1

Structural equal conflict sets:

1 [ ] [10..T\_0] :1

2 [ ] [11..T\_1] :1

3 [ ] [12..T\_2] :1

4 [ ] [13..T\_3] :1.

5 [ ] [14..T\_4] :1.

6 [ ] [15..T\_5] :1 graph

7 [ ] [16..T\_6] :1

8 [ ] [17..T\_7] :1

9 [ ] [18..T\_8] :1

10 [ ] [19..T\_9] :1

11 [ ] [20..T\_10] :1

12 [ ] [21..T\_11] :1

time: 0 s 0 s

Applying rule:

SC => CON

Results:

PUR	DRD	HOM	NBM	CSV	SCF	Ft0	tF0	Fp0	pF0	CON	SC	NC
N	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	ES
SCCS	SECS	9	8									



# Main Features

The main feature of Charlie are:

- open files: Snoopy, INA and APNN file-format;
- analyzers:
  - incidence matrix based analysis;
  - siphon/trap computation;
  - reachability/coverability graph;
  - model checking (CTL/LTL);
  - path search;
  - visualization of properties;
- rule system.



# Charlie's Rule System

Rule  $\hat{=}$  Theorem: set of results (pre-conditions) and another set of results (post-conditions)

Advantage: retrieving results without heavy computations,  
e.g.,  $CPI \Rightarrow SB \Rightarrow k-B$



# Charlie's Rule System

Rule  $\hat{=}$  Theorem: set of results (pre-conditions) and another set of results (post-conditions)

Advantage: retrieving results without heavy computations,  
e.g., CPI => SB => k-B

net properties				
PUR	ORD	HOM	NBM	CSV
SCF	FT0	TF0	FP0	PF0
CON	SC	NC (n...	RKTH	STP
CPI	CTI	SCTI	SB	k-B
DCF	DSt	DTr	LIV	REV

Navigation: << < > >> 1 (2) 🔍 output ?

# Charlie's Rule System

Rule  $\hat{=}$  Theorem: set of results (pre-conditions) and another set of results (post-conditions)

Advantage: retrieving results without heavy computations, e.g.,  $CPI \Rightarrow SB \Rightarrow k-B$

The screenshot shows a software interface with a grid of results and a dialog box titled "Apply rules?".

The grid is titled "net properties" and contains the following results:

PUR	ORD	HOM		
SCF	FT0	TF0		
CON	SC	NC (n		
CPI	CTI	SCTI	SB	k-B
DCF	DSt	DTr	LIV	REV

The dialog box "Apply rules?" contains the following text:

The following rules can be applied to this result set:  
 $CPI \Rightarrow k-B$   
 $CPI \Rightarrow SB$

Buttons: Yes, No

# Charlie's Rule System

Rule  $\hat{=}$  Theorem: set of results (pre-conditions) and another set of results (post-conditions)

Advantage: retrieving results without heavy computations, e.g.,  $CPI \Rightarrow SB \Rightarrow k-B$

The screenshot shows the Charlie's Rule System interface. It features a 'net properties' table and a dialog box titled 'Apply rules?'.

**net properties (Left):**

PUR	ORD	HOM		
SCF	FT0	TF0		
CON	SC	NC (n...		
CPI	CTI	SCTI	SB	k-B
DCF	DSt	DTr	LIV	REV

**Apply rules? Dialog:**

The following rules can be applied to this result set:  
 CPI => k-B  
 CPI => SB

**net properties (Right):**

PUR	ORD	HOM	NBM	CSV
SCF	FT0	TF0	FP0	PF0
CON	SC	NC (n...	RKTH	STP
CPI	CTI	SCTI	SB	k-B
DCF	DSt	DTr	LIV	REV

# Charlie's Rule System

Rule  $\hat{=}$  Theorem: set of results (pre-conditions) and another set of results (post-conditions)

Advantage: rules are shown to user  $\Rightarrow$  can be used in teaching



# Advantages of a Plugin System (or Why do we need a Plugin System?)

Charlie has a good list of analyzers and rules.



# Advantages of a Plugin System (or Why do we need a Plugin System?)

Charlie has a good list of analyzers and rules.  
However, since Charlie is (currently) closed software writing new analyzers and deploying a new version

- takes some time;
- maybe is not wanted to be included by Charlie's authors;
- usability suffers from too many unneeded options.

# Advantages of a Plugin System (or Why do we need a Plugin System?)

Charlie has a good list of analyzers and rules. However, since Charlie is (currently) closed software writing new analyzers and deploying a new version

- takes some time;
- maybe is not wanted to be included by Charlie's authors;
- usability suffers from too many unneeded options.

With a plugin system, users can

- easily extend a program by writing a plugin or by loading a plugin;
- implement algorithm that they think are useful;
- activate the features they want to have activated.

# Advantages of a Plugin System (or Why do we need a Plugin System?)

Developers of a tool profit from a plugin system by:

- standardizing several features;
- implementing a good framework;
- force themselves to follow a design.

# Advantages of a Plugin System (or Why do we need a Plugin System?)

Developers of a tool profit from a plugin system by:

- standardizing several features;
- implementing a good framework;
- force themselves to follow a design.

Therefore

- it is easier for developers, new to the project, to get started;
- overall productivity increases.

# Basic Requirements

We stated the following requirements to a plugin system:

- 1 possibility of accessing Charlie's core, e.g., starting analyzers, evaluating results, adding information to the log file, ...;
- 2 basic Petri net framework;
- 3 handle external libraries in plugin file;
- 4 possibility of sharing code between plugins;
- 5 dependency check;
- 6 start and stop plugins during runtime.

# Basic Requirements

We stated the following requirements to a plugin system:

- 1 possibility of accessing Charlie's core, e.g., starting analyzers, evaluating results, adding information to the log file, ...;
- 2 basic Petri net framework;
- 3 handle external libraries in plugin file;
- 4 possibility of sharing code between plugins;
- 5 dependency check (Future Work);
- 6 start and stop plugins during runtime (Future Work).

# Charlie's Plugin System

With a plugin one can extend the following features of Charlie:

- the set of analyzers;
- the set of rules;
- the GUI;
- the set of supported file formats.

# Charlie's Plugin System

With a plugin one can extend the following features of Charlie:

- the set of analyzers;
- the set of rules;
- the GUI;
- the set of supported file formats.

Plugins do not need to provide any of the features above, e.g., if a plugin shall share common code only.



# Charlie's Plugin System

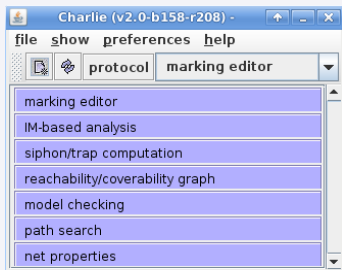
With a plugin one can extend the following features of Charlie:

- the set of analyzers;
- the set of rules;
- the GUI;
- the set of supported file formats.

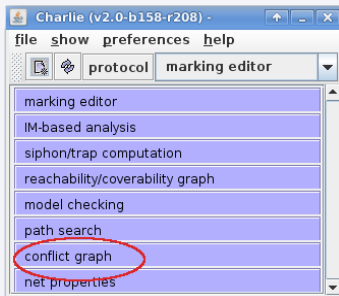
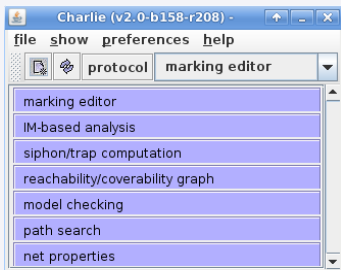
Plugins do not need to provide any of the features above, e.g., if a plugin shall share common code only.

Note: there is no dependency check yet.

# Plugin: GUI & Analyzer



# Plugin: GUI & Analyzer



# Plugin: Rule & Net Properties

net properties				
PUR	ORD	HOM	NBM	CSV
SCF	FT0	TF0	FP0	PF0
CON	SC	NC (S...	RKTH	STP
CPI	CTI	SCTI	SB	k-B
DCF	DSt(0)	DTr	LIV	REV

Navigation: ⏪ ⏩ 5 (5) 🔍 output ?

# Plugin: Rule & Net Properties

net properties				
PUR	ORD	HOM	NBM	CSV
SCF	FT0	TF0	FP0	PF0
CON	SC	NC (S...	RKTH	STP
CPI	CTI	SCTI	SB	k-B
DCF	DSt(0)	DTr	LIV	REV

◀◀ ▶▶ 5 (5) 🔍 output ?

net properties				
PUR	ORD	HOM	NBM	CSV
SCF	FT0	TF0	FP0	PF0
CON	SC	NC (S...	RKTH	STP
CPI	CTI	SCTI	SB	k-B
DCF	DSt(0)	DTr	LIV	REV
SPP	PER	DEF(0)	DIF(0)	

◀◀ ▶▶ 10 (10) 🔍 output ?

# Plugin: Rule & Net Properties

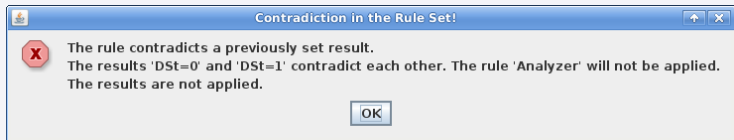
net properties				
PUR	ORD	HOM	NBM	CSV
SCF	FT0	TF0	FP0	PF0
CON	SC	NC (S...	RKTH	STP
CPI	CTI	SCTI	SB	k-B
DCF	DSt(0)	DTr	LIV	REV

5 (5) output ?

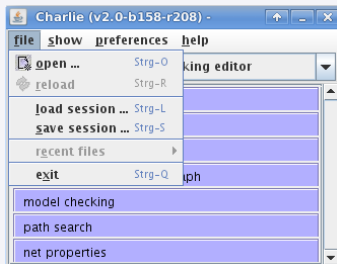
net properties				
PUR	ORD	HOM	NBM	CSV
SCF	FT0	TF0	FP0	PF0
CON	SC	NC (S...	RKTH	STP
CPI	CTI	SCTI	SB	k-B
DCF	DSt(0)	DTr	LIV	REV
SPP	PER	DEF(0)	DIF(0)	

10 (10) output ?

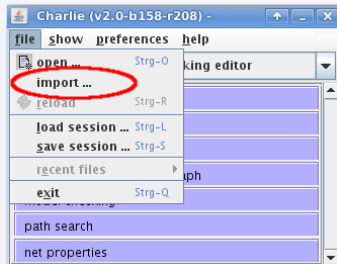
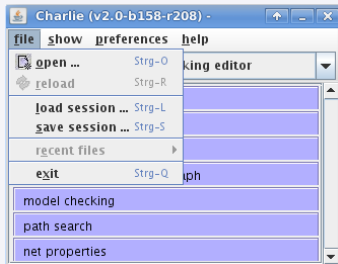
Hint: Rules are easily implemented  $\Rightarrow$  quick check for contradictions



# Plugin: Readers



# Plugin: Readers





# Conclusion

We have seen that Charlie's plugin system:

- are useful for users;
- increases usability;
- can extend important features;
- are powerful due to code sharing;
- can use external libraries.

Future Work:

- dependency check;
- start/stop plugin during runtime.

# Charlie's Website

## Download

Charlie can be downloaded free of charge at

<http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Charlie>

## Bug Report

Bugs can be reported at

<http://www-dssz.informatik.tu-cottbus.de/DSSZ/Bugs/CharlieBugList>

There is also a link on Charlie's website.



## Thank You

