Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work

JCircus 2.0: an extension of an automatic translator from Circus to Java

Samuel Lincoln Magalhães Barrocas¹ Marcel Vinícius Medeiros Oliveira¹

¹Universidade Federal do Rio Grande do Norte, Brazil

Communicating Process Architectures 2012

Introduction	Circus	JCSP	JCircus	Extensions to JCircus	Performance Analysis	Conclusions and Future Work

Outline

- Introduction
 - Motivation
- 2 Circus
 - The Circus language
- 3 JCSP
 - JCSP concepts
 - JCSP limitations
- 4 JCircus
 - JCircus
- 5 Extensions to JCircus
 - Extensions to JCircus
- 6 Performance Analysis
 - Performance Analysis
 - Conclusions and Future Work
 - Conclusions and Future Work

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work	
Motivation							
Why translating formal specifications into code?							

It is possible to reduce the occurrence of errors in the implementation of software systems from formal specifications...



Figure : Formal specification to code

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Motivation						
Goal						

To extend JCircus by providing

- an optimized strategy to translate the Multi-synchronisation primitive;
- a strategy to translate communications with arbitrary field decorations (e.g. c?x!y.10);
- a strategy to translate alphabetized parallelism;
- the translation of deadlock-free GUI processes to interact with the generated processes;
- the integration of JCircus with a refinement tool, called CRefine;

Introduction	Circus ●○○	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work		
The Circus language								
Circus Syntax								

- Z + CSP + Refinement Calculus + G.C. language;
- Circus Specification formed by paragraphs: Z paragraph,
 - Channel declaration paragraph,
 - Channel set declaration paragraph and
 - Process paragraph;

Introduction	Circus ○●○	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work	
The Circus language							
Roulett	е						

process Roulette
$$\hat{=}$$

begin
• $\mu X \bullet start \rightarrow \begin{pmatrix} result.RED \rightarrow X \\ \sqcap result.BLACK \rightarrow X \end{pmatrix}$
end

Figure : The Roulette process

Introduction	Circus ○○●	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
The Circus lan	nguage					
Croupie	er					

process Croupier
$$\hat{=}$$

begin
StartRoulette $\hat{=}$ start \rightarrow TakeBet
TakeBet $\hat{=}$ enter?id \rightarrow bet.id?b \rightarrow result?x
 $\begin{pmatrix} \mathbf{if} \ (x = b) \rightarrow (pay.ident \rightarrow Skip) \\ \|(x \neq b) \rightarrow Skip \\ \mathbf{fi} \end{pmatrix}$
 \sqcap stopBet \rightarrow Skip

• μX • *StartRoulette*; *X* end

process $Table \cong Roulette [[{ <math>start, result }]]$ Croupier

Figure : The Croupier process

Introduction	Circus 000	JCSP ●○	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work	
JCSP concept	s						
JCSP concepts							

- Java API for implementing CSP constructs;
- Processes are classes with the CSProcess interface;
- The behavior of each process is implemented in its run method;
- Parallelism: Parallel class;
- External choice: Alternative;
- Multi-Synchronisation: AltingBarrier;

Introduction	Circus 000	JCSP ○●	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
JCSP limitation	าร					
JCSP li	mitati	ons				

- Concurrency model different from CSP's concurrency model;
- Partial implementation of the Communication primitive: JCSP channels only communicate one value at most;
- Multi-synchronisation without communication;
- External choice without alting processes;

Introduction	Circus 000	JCSP oo	JCircus ●○○	Extensions to JCircus	Performance Analysis	Conclusions and Future Work	
JCircus							
JCircus' definition							

Java application that translates Circus specifications into Java code;

Introduction	Circus 000	JCSP oo	JCircus o●o	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
JCircus						
JCircus	s' moo	dules				

- Parser
- Typechecker
- Pre-processor
- Translator

Introduction	Circus 000	JCSP oo	JCircus ○○●	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
JCircus						

JCircus' initial screen

🛃 JCircus		x
Input specification:	C:\Users\sam\Softwares\workspace\ControleDeLargadaPAntigo\controle	
Project name:	paper	
Project path:	C:\Users\sam\Softwares\workspace\ControleDeLargadaPAntigo\src	
Project folders will I	be created in:	
C:\Users\sam\Softw	rares\workspace\ControleDeLargadaPAntigo\src\paper\src	
	Translate	
Log:		_

Figure : JCircus' initial screen

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to	JCircus					
Extensi	ons r	orovid	led to	JCircus		

- Optimized Multi-synchronisation;
- Complex communications;
- Alphabetized parallelism;
- Deadlock-free GUI;
- Integration with CRefine;

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to	JCircus					
Optimiz	zed M	ulti-s	ynchro	onisation		

- Protocol x Alting Barriers;
- Use of alting barriers;

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to J	lCircus					
Comple	x con	nmun	icatior	ns :: Definitio	n	

- Simple communication x Complex communication; ch?x?y is a complex communication; ch.5?x and ch!4!7 are not complex communications;
- JCSP does not implement communications with more than one field

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to J	lCircus					
Comple	x con	nmun	icatior	ns :: Strategy		

- Expansion of the possibilities of communication;
- Assignment of the values to the input variables;
- Inference of the next action based on the chosen communication (in the case of an external choice);

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to J	Circus					

Complex communications :: Strategy

$$\{0,1\} \times \{0,1,2,3\}$$

$$c_1?x?y \to A_1(x,y) \Box c_2?x?y \to A_2(x,y)$$

$$\{0 \mapsto \{x \mapsto 0, y \mapsto 0\}, \dots, 7 \mapsto \{x \mapsto 1, y \mapsto 3\}\}$$

$$\{0 \mapsto 0, \dots, 7 \mapsto 0, 8 \mapsto 1, \dots, 15 \mapsto 1\}$$

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to	JCircus					
Alphab	etized	l Para	llelisn	n		

- Hoare's parallelism x Alphabetized parallelism
- JCSP: implements Hoare's parallelism;
- Circus: Parallelism is alphabetized;
- Challenge: To force interleaving between occurrences of common events;

Alphabe	etizer	Para	llolien	n ··· Translatio	n	
Extensions to .	JCircus					
Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work

The strategy uses channel renaming, and consists of the following steps:

- Identification of the parallel branches;
- Construction of the synchronisation sets;
- Defining the renaming of each parallel branch;
- Renaming processing;
- Hiding the renaming from the interface;

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to J	Circus					

Alphabetized Parallelism



Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to	JCircus					
Deadlo	ck-fre	e GU				

- For each process generated in JCSP, it is generated, also in JCSP, a GUI that interacts with it;
- There was a strategy for generating a GUI in the original version of JCircus, but it was not deadlock-free;

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
Extensions to	JCircus					
Deadlo	ck-fre	e GU				

🛃 Roulette
result BLACK -
start
timerSync: GUI and Process synchror timerSync: GUI and Process did NOT
timerSync: GUI and Process synchron \equiv
Ţ

Extensions to JCircus Integration with CRefine								
Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work		

- CRefine is a tool that automates the application of laws to refine specifications;
- The integration demanded the migration of JCircus to CZT's newest parser of Circus;

Introduction	Circus	JCSP	JCircus	Extensions to JCircus	Performance Analysis	Conclusions and Future Work
				0000000000		

Extensions to JCircus

Integration with CRefine

🖼 😸 🕐 LaTeX 🔾 Unicode Collect Actions 🔻 : Colec	t Action Click here to select the law
Refinement	n" 🖪 🗌 Code
ComplexComms	(begin(circus)
Wagh(circus) -kinchamel data, chan2: Inat cross Vinat cross Vinat Venchamel data, chan2: Inat cross Vinat cross Vinat Venchamel data, chan2: Vinat cross Vinat cross Vinat Venchamel data, strat cross Vinat cross Vinat Venchamel data, strat cross Vinat cross Vinat Venchamel data, strat cross Vinat cross Vinat Venchamel data, Vinat cross Vinat cross Vinat Venchamel data, Vinat cross Vinat cross Vinat cross Vinat Venchamel data, Vinat cross Vinat cross Vinat cross Vinat Venchamel data, Vinat cross Vinat cross Vinat cross Vinat Venchamel chan5: Vinat cross Vinat cross Vinat Venchamel chan5: Vinat cross Vinat cross Vinat Venchamel chan5: Vinat Venchamel chan5: Vinat Venchamel chan5: Vinat	Circumel Char, Huhar, 'Nat cross Inat end/circus) Veghi(circus) Veghi(
Proof Oblication	chan47x?yithen IStép
ladowr	Vcircend
amplayComme	initial circulat
Onpiexcomms	(begin(circus)
i staus i rredicate	l circprocess SCC1 loircdef loircbegin loircstable Default == [true]
	\circspot chan?x7y.1 \then

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis ●00	Conclusions and Future Work		
Performance Analysis								

Multi-synchronisation

Number of	Translat	ion (ms)	Executi	on (ms)	Memory (KB)	
Processes	Protocol	Barriers	Protocol	Barriers	Protocol	Barriers
3	257	115	157,8	106,4	11276	14732
4	314	130	170,8	98	11232	14568
5	254	143	152,8	97,8	11208	14520
30	1764	344	307,3	136,6	11228	15636
100	7052	17949	548,6	378	11692	19680



Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work	
Performance Analysis							

Complex Communications



Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work		
Performance Analysis								

Alphabetized Parallelism



Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work ●○			
Conclusions a	Conclusions and Future Work								
Conclusions and Future Work									

- Translation of Hiding;
- Translation of Schema types;
- Optimization of complex communications;

Introduction	Circus 000	JCSP oo	JCircus 000	Extensions to JCircus	Performance Analysis	Conclusions and Future Work ○●		
Conclusions and Future Work								
Questic	ons?							

