Introduction to Choreographies

In concurrent and distributed systems, processes can complete tasks together by playing their parts in a joint plan. The plan, or protocol, can be written as a choreography: a formal description of the overall behaviour that processes should collaborate to implement, like authenticating a user or purchasing an item online. Formality brings clarity, but not only that. Choreographies can contribute to important safety and liveness properties.

This book is an ideal introduction to theory of choreographies for students, researchers, and professionals in computer science and applied mathematics. It covers languages for writing choreographies and their semantics, and principles for implementing choreographies correctly. The text treats the study of choreographies as a discipline in its own right, following a systematic approach that starts from simple foundations and proceeds to more advanced features in incremental steps. Each chapter includes examples and exercises aimed at helping with understanding the theory and its relation to practice.

FABRIZIO MONTESI is Professor of Computer Science at the University of Southern Denmark. He is a Villum Young Investigator and recipient of several awards for science and innovation, including the EAPLS Best PhD Dissertation Award and the Best Thesis in ICT Award from the General Confederation of Italian Industry.

Introduction to Choreographies

FABRIZIO MONTESI
University of Southern Denmark

CAMBRIDGE
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment,
a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of
education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108833769
DOI: 10.1017/9781108981491

First published 2023

A catalogue record for this publication is available from the British Library.

A Cataloging-in-Publication data record for this book is available from the Library of Congress.

ISBN 978-1-108-83376-9 Hardback

Contents

Illustrations

Notations

$\hookrightarrow$	An exercise for which a solution is provided	9
!	A (possibly) challenging exercise	9
conn(A, B)	There is a direct connection from A to B	16
walk(A, B)	There is a walk from A to B	16
City	The set of cities considered in Chapter 1	17
$\triangleq$	Definition symbol, read 'is defined as'	17
$\in$	Set membership, read 'is an element of'	17
size($\mathcal{D}$)	The size of derivation $\mathcal{D}$	20
height($\mathcal{D}$)	The height of derivation $\mathcal{D}$	20
$\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$	Derivations	22
$\mathcal{D} :: p$	The derivation $\mathcal{D}$, which concludes the proposition p	22
$=$	Equality, read 'is equal to'	24
PName	The set of process names	39
p	A process name	39
C	A choreography	39
SimpleChor	The language of Simple Choreographies	39
0	A terminated program, either in a choreographic language or a process language	40, 56
p → q	The term or label of a communication from p to q	40
$\longrightarrow$	A transition relation	41
s	A state in a labelled transition system	41
μ	A transition label	41
$s \xrightarrow{\mu} s'$	The state s can make a transition with label μ to the state s'	41
pn	The function that computes the set of process names in a label or term	41, 80
$S_1 \# S_2$	Shortcut to $S_1 \cap S_2 = \emptyset$ (the sets S_1 and S_2 have empty intersection)	43
$S_1 \cap S_2$	The intersection of the sets S_1 and S_2	43
$\emptyset$	The empty set	43
$s \xrightarrow{\mu}$	The state s can make a transition with label μ	46
$s \not\xrightarrow{\mu}$	The state s cannot make a transition with label μ	46
$\vec{\mu}$	A sequence of transition labels (possibly empty)	46
ϵ	The empty sequence	46

$s \xrightarrow{\vec{\mu}}\!\!\!\!\rightarrow s'$	There is an execution from s to s' with trace $\vec{\mu}$	47
$\twoheadrightarrow$	A multi-step transition relation	47
$s \xrightarrow{\vec{\mu}}\!\!\!\!\rightarrow$	The state s has the trace $\vec{\mu}$	47
$s \not\xrightarrow{\vec{\mu}}\!\!\!\!\rightarrow$	The state s does not have the trace $\vec{\mu}$	47
P	A process term	56
$\mathsf{p}!$	Send a message to p	56
$\mathsf{p}?$	Receive a message from p	56
SimpleProc	The language of Simple Processes	56
N	A network	57
SimpleNet	The set of networks in Simple Processes	57
supp	The function that maps functions to their supports	57
$\mathsf{p}[P]$	The atomic network that returns P for p	57
$N \mid M$	The parallel composition of N and M	58
$N \setminus \mathsf{p}$	The network obtained by removing p as running process from N	67
$\vec{\mathsf{p}}$	A sequence of process names	69
$N\!\upharpoonright_{\{\mathsf{p}_1,\ldots,\mathsf{p}_n\}}$	The restriction of N to $\{\mathsf{p}_1,\ldots,\mathsf{p}_n\}$	69
$N\!\upharpoonright_{\mathsf{p}_1,\ldots,\mathsf{p}_n}$	Shortcut for $N\!\upharpoonright_{\{\mathsf{p}_1,\ldots,\mathsf{p}_n\}}$	69
$[\![C]\!]_{\mathsf{p}}$	The projection of choreography C on process p	77
$[\![C]\!]$	The endpoint projection (EPP) of choreography C	78
Var	The set of (local) variables	95
x	A variable (used in local stores and expressions)	95
Val	The set of (local) values	95
v	A value (used in local stores and expressions)	95
I	An instruction, either in a choreographic or a process language	95
$I; C$	Do I and then C, in a choreographic language	95
$\mathsf{p}.e \rightarrow \mathsf{q}.x$	Process p communicates the evaluation of e to q, which stores it in variable x	95
e	A (local) expression	95
$\mathsf{p}.x := e$	Process p evaluates e and stores the result in x	95
$f(\vec{e})$	The invocation of function f with the evaluations of $\vec{e}$ as arguments	96
f	A function name	96
StatefulChor	The language of Stateful Choreographies	96
σ	A process store	98
PStore	The set of process stores	98
$\sigma[x \mapsto v]$	The store obtained from σ by updating the value for variable x to v	98
Σ	A choreographic store	98
$\Sigma[\mathsf{p}.x \mapsto v]$	The choreographic store obtained from Σ by updating the value for variable x of process p to v	98
$\sigma \vdash e \downarrow v$	The expression e is evaluated to v under σ	99
$\langle C, \Sigma \rangle$	A configuration in Stateful Choreographies	100

τ @p	The label of an internal action at process p	100
p.$v \rightarrow$ q	The label of a communication of the value v from p to q	100
$I; P$	Do I and then P, in a process language	102
p!e	Send the evaluation of e to p	102
p?x	Receive a value from p and store it in x	102
$x := e$	Store the evaluation of e in x	102
StatefulProc	The language of Stateful Processes	102
StatefulNet	The set of networks in Stateful Processes	102
$\langle N, \Sigma \rangle$	A configuration in Stateful Processes	103
if p.e then C_1 else C_2	Run the choreography C_1 if p evaluates e to *true*, C_2 otherwise	106
true	The Boolean value true	107
false	The Boolean value false	107
ConditionalChor	The language of Conditional Choreographies	107
$\fatsemi$	The sequential composition operator, either for choreographies or for process terms	108
ConditionalProc	The language of Conditional Processes	113
ConditionalNet	The set of networks in Conditional Processes	113
SLabel	The set of all selection labels	121
L	A selection label	121
p $\rightarrow$ q[L]	Process p communicates the selection of label L to q	121
SelectiveChor	The language of Selective Choreographies	121
p $\oplus$ L	Send to p the selection of label L	124
p $\& \{\mathrm{L}_i : P_i\}_{i \in I}$	Receive from p a label L_j, for $j \in I$, and then proceed as P_j	124
SelectiveProc	The language of Selective Processes	124
$\sqcup$	The merging operator	127
X	A procedure name	134
RecursiveChor	The language of Recursive Choreographies	134
$\mathscr{C}$	A set of choreographic procedure definitions	134
$X(\vec{\mathsf{p}}) = C$	Procedure X has parameters $\vec{\mathsf{p}}$ and body C	134
$X(\vec{\mathsf{p}})$	Call procedure X with arguments $\vec{\mathsf{p}}$	134, 147
C[r/s]	The choreography C, but with r replacing s	139
RecursiveProc	The language of Recursive Processes	147
$\mathscr{P}$	A set of process procedure definitions	147
$X(\vec{\mathsf{p}}) = P$	Procedure X has parameters $\vec{\mathsf{p}}$ and body P	147
P[r/s]	The process P, but with r replacing s	149
$\sqsupseteq$	The branching partial order	157
$\langle \mathscr{C}, \{\vec{\mathsf{p}}\}, C \rangle \checkmark$	The choreography C is well-formed in the context of $\mathscr{C}$ and involves at most the processes in $\vec{\mathsf{p}}$	161
$\forall$	Universal quantification, read 'for each', 'for any', or 'for all'	163
EnrichedChor	The language of Enriched Choreographies	179
$\lfloor C \rfloor$	The desugaring, or expansion, of choreography C	179

$\mathsf{p}.\overset{e}{y} \rightleftarrows \mathsf{q}.\overset{x}{e'}\{C\}$	A Request-Reply, where p communicates a request to q and the latter replies after the choreography C is executed	180
if $\mathsf{p}_1.e_1$ and $\cdots$ and $\mathsf{p}_n.e_n$ then C_1 else C_2	A distributed conditional, where C_1 is executed if each condition e_i evaluates to *true* in the state of its respective process p_i, and C_2 is executed otherwise	182
$C_1 +_{\mathsf{p}} C_2$	Process p chooses between C_1 and C_2	185
$P + Q$	The process that behaves as P or as Q	188
m	A message	191
(p, v)	A message that consists of a value v sent by a process p	192
(p, ℓ)	A message that consists of a label ℓ sent by a process p	192
$\vec{m}$	A sequence of messages (possibly empty)	192
Queue	The set of message queues	192
K	A messaging state	192
$\mathsf{p}.v \rightarrow \mathsf{q}!$	The label of an asynchronous transmission of the value v from p to q	192
$\mathsf{p}.v \rightarrow \mathsf{q}?$	The label of an asynchronous reception by q of the value v from p	192
$\mathsf{p} \rightarrow \mathsf{q}[\ell]!$	The label of an asynchronous transmission of the label ℓ from p to q	192
$\mathsf{p} \rightarrow \mathsf{q}[\ell]?$	The label of an asynchronous reception by q of the label ℓ from p	192