

AGERE! @ SPLASH 2012

2nd International Workshop on Programming based on
Actors, Agents, and Decentralized Control



Oct 21-22
Tucson, Arizona - US

ORGANIZERS & PROGRAM CHAIRS

GUL AGHA (University of Illinois at Urbana-Champaign, US)

RAFAEL H. BORDINI (FACIN–PUCRS, Brazil)

ASSAF MARRON (Weizmann Institute of Science, Israel)

ALESSANDRO RICCI (University of Bologna, Italy)

PROGRAM COMMITTEE

Gul Agha, University of Illinois at Urbana-Champaign, USA

Joe Armstrong, SICS / Ericsson, Sweden

Saddek Bensalem, Verimag, France

Rafael H. Bordini, FACIN–PUCRS, Brazil

Gilad Braha, Google, USA

Rem Collier, UCD, Dublin

Tom Van Cutsem, Vrije Universiteit, Brussel, Belgium

Amal El Fallah Seghrouchni, LIP6 - Univ. P and M. Curie, Paris, France

Jurgen Dix, Technical University of Clausthal, Germany

Philipp Haller, Typesafe, Switzerland

Tom Holvoet, Dept. Computer Science K.U.Leuven, Belgium

Einar Broch Johnsen, University of Oslo, Norway

Hillel Kugler, Microsoft, USA

Assaf Marron, Weizmann Institute of Science, Israel

Mark Miller, Google, USA

Olaf Owe, University of Oslo, Norway

Jens Palsberg, UCLA, Los Angeles, USA

Ravi Pandya, Microsoft, USA

Arnd Poetsch-Heffter, University of Kaiserslautern, Germany

Alessandro Ricci, University of Bologna, Italy

Birna van Riemsdijk, Delft University of Technology, The Netherlands

Giovanni Rimassa, Whitestein Technologies, Switzerland

Munindar Singh, North Carolina State University, USA

Marjan Sirjani, Reykjavik University, Iceland

Gera Weiss, Ben Gurion University, Israel

Guy Wiener, HP, Israel

Akinori Yonezawa, University of Tokyo, Japan

āgā, āgis, egi, actum, āgĕre

latin verb meaning “to act, to lead, to do”

common root for

actors

agents

the workshop is about

computing & programming paradigms based on
actor-oriented & agent-oriented abstractions

...and on any paradigm promoting a *decentralized
control mindset* in design and programming

- programming languages
- computing & programming models and architectures
- platforms and tools
- applications
- design aspects & methodologies
- ...

OOPSLA's topics but
focusing on actors, agents
and decentralized control approaches
as programming paradigms
evolution of the OOP one.

WHY AGERE?

“The Free Lunch is Over” call

“Software and the concurrency revolution”
ACM Queue: Tomorrow’s Computing
Today, 3(7):54–62, Sept. 2005.

concurrency is no more a an issue
related to specific application domains

aspects of concurrency are more and
more pervasive in programming &
software development

no only mechanisms for improving
performances, but proper models,
methodologies and technologies to
improve design and abstraction

DIRECTIONS (I)

go on as far as possible with the free lunch

keeping old good paradigms and languages and way of thinking (OO & functional & declarative..) and adding specific mechanisms

hide concurrency, parallelism, decentralization as much as possible (=> e.g. automatic parallelization)

mainstream prog. languages + “small” extension/libraries/framework (hiding the new paradigm)

DIRECTIONS (2) - **AGERE!'S ONE**

explore and embrace new metaphors / paradigms

“as effective as OO has been for monolithic/non-concurrent programs”

strong impact on the way on which systems are designed and programmed

- new languages & related ecosystem
- traditional languages + libraries/framework exposing to the paradigm shift

AGERE!'S FOCUS

programming paradigms based on

- actor-oriented
- agent-oriented
- decentralized-control

Impact on

- performances, but not only
- on the way of solving problems and thinking and designing modern programs

REMARK

apparently direction (I) is more common in the mainstream

see OOPSLA/SPLASH 2011 main track

- > strong emphasis on performances
- > ...and *mechanisms* to improved it
- > multi/many-core rush

...with some exception

- > e.g. Krab's invited on Erlang
- > “not only performance but conceptual design & abstraction”

BACK TO ACTORS

back to OOPSLA '90, 22 years ago

Workshop on

OBJECT-BASED CONCURRENT PROGRAMMING

(Gul Agha, Carl Hewitt, Peter Wegner, Akinori Yonezawa)

convergence to

the actor model and actor-related approaches

as unification of objects + concurrency

programs as ensemble of

autonomous computing entities

communicating through asynchronous message passing

BACK TO ACTORS

ROOTS BACK INTO 70ies and 80ies

[Alan Kay](#). Original Object idea, centered on messages and reactivity principle (1969)

[Carl Hewitt](#). Viewing Control Structures as Patterns of Passing Messages Journal of Artificial Intelligence. June 1977.

[Gul Agha](#) (1986). Actors: A Model of Concurrent Computation in Distributed Systems. Doctoral Dissertation. MIT Press

[Akinori Yonezawa](#), [Jean-Pierre Briot](#), Etsuya Shibayama: Object-Oriented Concurrent Programming in ABCL/I. 258-268, OOPSLA 86

...

ACTOR-BASED OR INSPIRED LANGUAGES, FRAMEWORKS, LIBRARIES

Act 1, 2 and 3

Acttalk

Ani

Cantor

Rosette

...

Erlang

ABCL family

E

AmbientTalk

SALSA

...

Scala

Kilim

Scratch

ActorFoundry

WebWorker

DART

SO WHY AGERE! IN 2011/2012?

The news is that...

actors are going to be more and more in
mainstream programming...

- an answer to the free lunch call

e.g.

Scala actor library

Akka - A Java and Scala framework with Actors, STM & Transactors

F# MailboxProcessor - an Actor Model in the F# core library

GPars - the concurrency and actor library for Groovy

Ericsson Erlang

Microsoft Axum/Orleans

Google DART

...

SO WHY AGERE! IN 2011/2012?

...to be used pervasively for developing software
in “every-day” programming

not only a matter of parallel programming & related specific
application domains..

eventually applied to hot application domains
- web & cloud, mobile, smart systems, ...

...eventually integrated with existing paradigms

...eventually dealing with large software systems
& programming-in-the-large complexities

AGERE! OBJECTIVES (I)

making actor-based prog languages & technologies
as robust as OOP has been
for large "traditional" software systems

linked with other main aspects that concern
development and software engineering
- e.g. design, methodologies

AGENTS?

AGENTS & *PROGRAMMING*??

aren't agents about AI & (D)AI?

a largely overloaded concept / set of concepts

slightly different meanings in different contexts

PERSPECTIVES ON AGENTS

AI and DAI (the root)

- **autonomous** entity
 perceiving and **acting** in some **environment**
 so as to achieve some designed **goal(s)**,
 eventually **communicating** with other agents
- **multi-agent** systems

distributed systems

- what (undefined) consumes services (SOA perspective)
- mobile active entities
- ...

concurrent programming & prog languages

- synonym of *actor* (= *active entity* + *message passing*)
 e.g. GPar, F#, Clojure (?)...

AGENT-ORIENTED PROGRAMMING

- ROOTS

Shoham (Stanford), 1993 - AGENT-0 language

“Agent-oriented programming”. Artificial Intelligence, 60(1):51–92, 1993.

- AI context

- *specialization* of OOP (...& actor-based computing)

- introducing a *mentalist* and *societal view*
on computation

agents as autonomous entities

integrating reactive behaviour & long-term goal-oriented activity

with a state defined in mentalistic attitude (belief, goal, commitments..)

and with speech-act based communication abilities

AGENT-ORIENTED PROGRAMMING

- DEVELOPMENT (90ies-00ies-today)

Several languages, frameworks, platforms



grew up in a (D)AI context

AAMAS conference, ProMAS/DALT workshops, JAAMAS journal

AGENT-ORIENTED PROGRAMMING

- DEVELOPMENT (90ies-00ies-today)

high-level cognitive/human-inspired models

- **BDI** (Belief-Desire-Intention) model

sound formalizations & theoretical background

typically applied in the context of distributed
“intelligent” systems

no meaningful impact on the mainstream research in
Programming Languages and Software Engineering

AGERE! OBJECTIVES (II)

exploring & developing programming paradigms
based on high-level agent-oriented abstractions
- as specialization/extension of object/actor ones

develop AOP and MULTI-AGENT PROGRAMMING
taking programming & software development
as reference contexts
- an answer to the free lunch call

AGERE! OBJECTIVES (III)

BESIDES ACTORS and AGENTS

exploring & developing programming paradigms
based on decentralized control & interaction
thinking

> a main recent example:

Behavioral Programming (Harel, Marron, Weiss)

CACM 55, 7 - 2012

1st DAY PROGRAMME

- 8:30 9:00 Welcome
 - Introduction to the AGERE! workshop: Motivation, Objectives, Agenda.
- 9:00 10:00 Invited talk

On the integration of the actor model in mainstream technologies The Scala perspective by Philipp Haller, Typesafe
- 10:00 10:30 Coffee Break
- 10:30 12:00 Research Papers session I Actors
 - *Domains: Safe sharing among actors*
Joeri De Koster, Tom Van Cutsem and Theo D'Hondt – Vrije Universiteit (Belgium)
 - *Soter: An Automatic Safety Verifier for Erlang*
Emanuele D'Ossualdo, Jonathan Kochems and Luke Ong – Oxford University (United Kingdom)
 - *Leveraging Actors for Privacy Compliance*
Jeffery Von Ronne – The University of Texas at San Antonio (United States)
- 12:00 13:30 Lunch

1st DAY PROGRAMME

- 13:30 14:30 Invited talk

20 years of Agent-Oriented Programming in Distributed AI: History and Outlook by Birna van Riemsdijk Delft University of Technology, The Netherlands

- 14:30 15:00 Research Paper Session IIa Agent-Oriented Programming

- *Adding Distribution and Fault Tolerance to Jason*

Álvaro Fernández Díaz, Clara Benac Earle and Lars-Ake Fredlund

- Universidad Politecnica de Madrid (Spain)

- 15:00 15:30 Coffee Break

- 15:30 17:00 Research Paper Session IIb Agent-Oriented Programming

- *Programming Abstractions for Integrating Autonomous and Reactive Behavior: An Agent-Oriented Approach*

Alessandro Ricci, Andrea Santi – University of Bologna (Italy)

- *Empirical Software Engineering for Agent Programming*

Birna Van Riemsdijk – TU Delft (The Netherlands)

- *Messages with Implicit Destinations as Mobile Agents*

Ahmad Ahmad-Kassem, Stphane Grumbach and Stphane Ubda – INRIA INSA Lyon, INRIA (France)

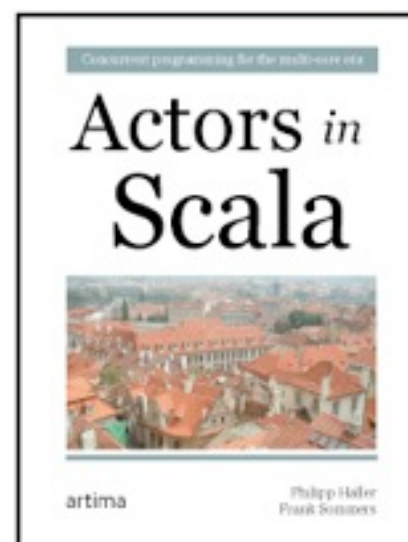
INVITED TALK:

PHILIPP HALLER

“On the integration of the actor model in mainstream technologies - The Scala perspective”

Philipp Haller is a post-doctoral researcher at Stanford University, USA, and at EPFL, Switzerland. He got his Ph.D. in Computer Science from EPFL, Switzerland. As a member of the Scala team at EPFL he has been working on programming abstractions for concurrency, as well as type systems to check their safety. Philipp created Scala Actors, a library for efficient, high-level concurrent programming. Now Philipp works at Typesafe.

Philipp is author of the book:
“Actors in Scala”



INVITED TALK:

BIRNA VAN RIEMSDIJK

“20 years of Agent-Oriented Programming in Distributed AI: History and Outlook”

Birna van Riemsdijk is assistant professor at the Man-Machine Interaction group at TU-Delft, The Netherlands. She got her PhD from Utrecht University, where she gave important contributions to agent oriented programming, in particular about the theory of cognitive goal-based agent programming languages. Since 2012, she is member of the International Foundation of Agents and Multi-Agent Systems (IFAAMAS) board.



2nd DAY PROGR.

- 8:30 9:00 Welcome
 - Brief summary of the first day and agenda of the second day. Preparation for the panel discussion.
- 9:00 10:00 Invited Talk

Agents, Concurrent Objects, and High Performance Computing
by Akinori Yonezawa University of Tokyo, Japan
- 10:00 10:30 Coffee Break
- 10:30 12:00 Research Paper Session III Decentralized control programming with Behavioral Abstractions
- Introduction by Assaf Marron
 - *A Decentralized Approach for Programming Interactive Applications with JavaScript and Blockly*
Assaf Marron, Gera Weiss and Guy Wiener – Weizmann Institute of Science, Ben Gurion University, HP Labs (Israel)
 - *Optimized Distributed Implementation of Multiparty Interactions with Observation*
Saddek Bensalem, Marius Bozga, Jean Quilbeuf and Joseph Sifakis
– VERIMAG, VERIMAG/CNRS (France)
 - *Distributed Priority Synthesis using Knowledge*
Chih-Hong Cheng, Rongjie Yan, Harald Ruess and Saddek Bensalem
– Fortiss (Germany), State Key Laboratory of Computer Science Institute of Software (China), VERIMAG (France)
- 12:00 13:30 Lunch time

2nd DAY PROGR.

- 13:30 15:00 Research Paper Session IV Actors
 - *Parallel Gesture Recognition with Soft Real-Time Guarantees*
Thierry Renaux, Lode Hoste, Stefan Marr and Wolfgang De Meuter
– Vrije Universiteit (Belgium)
 - *A Relational Trace Logic for Simple Hierarchical Actor-Based Component Systems*
Ilham W. Kurnia and Arnd Poetzsch-Heffter – University of Kaiserslautern (Germany)
 - *Timed-Rebeca Schedulability and Deadlock-Freedom Analysis Using Floating-Time Transition System*
Ehsan Khamespanah, Zeynab Sabahi Kaviani, Ramtin Khosravi, Marjan Sirjani and Mohammad-Javad Izadi – Tehran University (Iran), Reykjavik University (Iceland)
- 15:00 15:30 Coffee Break
- 15:30 17:00 Research Paper V and Discussion Session
 - *Actor Idioms*
Dale Schumacher – United States
 - Research directions for Agent, Actor and Decentralized Control: panel and open discussion
- 17:00 SPLASH Poster session with one poster about AGERE! summary & contributions

INVITED TALK (MONDAY):

AKINORI YONEZAWA

*“Agents, Concurrent Objects,
and High Performance Computing*

Akinori Yonezawa

- Professor at Department of Computer Science, University of Tokyo
- Ph.D. in Computer Science from the Massachusetts Institute of Technology in 1977.
- ACM Fellow since 1999
- AITO Dahl-Nygaard Prize in 2008 for his pioneer work on the concept, theory and applications of concurrent objects
- "Medal of Honour with a Purple Ribbon of Japan (Shiju-housho)" at Imperial Palace in 2009
- heavily involved in the in Japanese National Supercomputer Project since 2007, he is member of the executive board of the national high performance computing infrastructure

He is author of several books, including:

"Object-Oriented Concurrent Programming", MIT Press, 1987.

"ABCL: an Object-Oriented Concurrent System", MIT Press 1990.

"Research Directions in Concurrent Object-oriented Programming", MIT Press, 1993.



ENJOY

CONTRIBUTE