EPTCS 27

Proceedings of the

# First International Workshop on Decentralized Coordination of Distributed Processes

**Amsterdam, The Netherlands, 10th June 2010**

Edited by: Tom Van Cutsem and Mark Miller

7th June 2010
ISSN: 2075-2180
Open Publishing Association

# Table of Contents

# Preface

This volume contains the papers presented at the first International Workshop on Decentralized Coordination of Distributed Processes, DCDP 2010, held in Amsterdam, The Netherlands on June 10th, 2010 in conjunction with the 5th International Federated Conferences on Distributed Computing Techniques, DisCoTec 2010.

The central theme of the workshop is the decentralized coordination of distributed processes. Decentralized meaning there is no single authority in the network that everything is vulnerable to. Coordinated meaning processes need to cooperate to achieve meaningful results, potentially in the face of mutual suspicion. Distributed, meaning processes are separated by a potentially unreliable network.

The workshop opened with a Keynote by Tyler Close (Google) titled "Using the Web for decentralized coordination of distributed processes. You *can* get there from here." Afterwards authors presented two full papers and two extended abstracts.

We wish to thank all members of the Program Committee who carefully reviewed the submissions:

- Fred Spiessens, Evoluware, Belgium
- Carl Hewitt, MIT EECS (Emeritus), USA
- Ben Laurie, Google, UK
- Alan Karp, Hewlett-Packard, USA
- Peter Van Roy, Universit Catholique de Louvain, Belgium
- Dean Tribble, Microsoft, USA
- Toby Murray, University of Oxford, UK
- Tyler Close, Google, USA
- Mark Miller, Google, USA
- Tom Van Cutsem, Vrije Universiteit Brussel, Belgium

We also thank Tyler Close for willing to give a Keynote speech. We thank Marcello Bonsangue for giving us the chance to organize this workshop at DisCoTec, and to the Editorial Board of the Electronic Proceedings in Theoretical Computer Science (EPTCS) for publishing the workshop proceedings.

*Tom Van Cutsem and Mark S. Miller*
*DCDP 2010 Organizers*

Tom Van Cutsem and Mark S. Miller (Eds.):
Decentralized Coordination of Distributed Processes 2010
EPTCS 27, 2010, pp. ii–ii, doi:10.4204/EPTCS.27.0

# A Wave-like Decentralized Reconfiguration Strategy for Self-organizing Resource-Flow Systems

Jan Sudeikat[1], Jan-Philipp Steghöfer[2], Hella Seebach[2], Wolfgang Reif[2], Wolfgang Renz[1], Thomas Preisler[1], Peter Salchow[1]
[1]Multimedia Systems Laboratory (MMLab),
Faculty of Engineering and Computer Science, Hamburg University of Applied Sciences,
Berliner Tor 7, 20099 Hamburg, Germany
{jan.sudeikat, wolfgang.renz, thomas.preisler, peter.salchow}@haw-hamburg.de
[2]Institute for Software and Systems Engineering,
Augsburg University, Universitätsstrasse 6a, 86135 Augsburg, Germany
{steghoefer, seebach, reif}@informatik.uni-augsburg.de

In resource-flow systems, e.g. production lines, agents are processing resources by applying capabilities to them in a given order. Such systems profit from self-organization as they become easier to manage and more robust against failures. In this paper, we propose a decentralized coordination process that restores a system's functionality after a failure by propagating information about the error through the system until a fitting agent is found that is able to undertake the required function. The mechanism has been designed by combining a top-down design approach for self-organizing resource-flow system and a systemic modeling approach for the design of decentralized, distributed coordination mechanisms. The systematic conception of the inter-agent process is outlined and initial evaluations show the convergence to stable, i.e. fully operative configurations.

A domain in which self-organization is beneficial are production automation systems. In traditional production lines, resources are transported from one workstation to the other on conveyor belts or with similar, rigid and inflexible transportation mechanisms. While relatively easy to manage, these systems come to a complete halt whenever one of their constituent parts ceases to function. Also, if the requirements of the production lines change, the system will have to be retooled, a difficult, expensive, and time-consuming process. To add failure tolerance to such systems and to make this rigid structure more flexible, production automation systems could instead use autonomous guided vehicles (AGVs) that transport resources and robots that have several tools they can change in order to apply different capabilities as required [1]. We call this class of systems *Self-Organizing Resource-Flow Systems* with application domains in production automation and logistics. Their basic structure can be described with the Organic Design Pattern (ODP) [2]. *Resources* are processed according to a *task* by independent *agents*. Each agent has a number of *capabilities* it can apply to the resource to alter it according to the task. Agents can exchange resources with other agents as, e.g., given by the layout of a shop floor.

In this paper, we propose a decentralized, self-organizing process for the class of self-organizing resource-flow systems. This process is analyzed and modeled with the tools provided by the SodekoVS[1] project [3]. Agents change their local configurations until the system in its entirety has restored a stable state. During reconfiguration, parts of the system that are not affected by the process or have already been reconfigured are still able to resume their normal work.

The utilization of self-organization principles in the development of distributed software systems is considered in the research project "Selbstorganisation durch Dezentrale Koordination in Verteilten

[1]Acknowledgments: This research is partly sponsored by the German research foundation (DFG) in the project SodekoVS and in the DFG special priority program "Organic Computing" (SPP 1183) in the project SAVE ORCA.

Tom Van Cutsem and Mark S. Miller (Eds.):
Decentralized Coordination of Distributed Processes 2010
EPTCS 27, 2010, pp. 32–33, doi:10.4204/EPTCS.27.4

Systemen" (SodekoVS) [3]. The supplementation of self-organizing features to software systems is based on the externalization of coordination models [4]. A coordination middleware serves as a reference model for the integration of decentralized self-organizing processes in MAS. It contains two types of functional elements for the encapsulation of these aspects. First, agents are connected with each other via so-called *Coordination Media*. These are communication infrastructures that allow to encapsulate specific interaction modes. Secondly, the utilization of these media is separated from the agent-logic by associating agents with *Coordination Endpoints*. They initiate and participate in medium-mediated interactions on behalf of the associated agents and effectuate modifications on agents when appropriate. This middleware separates and automates the activities, i.e. interactions and local adjustments, that are conceptually related to coordination.

This reference architecture has been used to realize a completely decentralized reconfiguration approach. It is based on the idea that a wave of role re-allocation runs through the system in order to re-establish the resource-flow. Assuming that each agent is capable to exhibit a set of capabilities, a correct resource flow can be (re-)established by the appropriate swapping of roles. Failing agents adopt actable roles and in return other agents help out by providing the unactable roles. Waves of reallocations originate from the failing agents as these send requests for assistance along the resource flow. Recipients of these request decide locally whether they are capable and willing to swap roles. Depending on the system configuration, it may be the case that a single swap of roles does not reestablish the correct sequence of activities, thus *transitive* changes of roles are required. Prior to the detailed design and embedding of this decentralized process, we anticipated the resulting system dynamics. The decentralized reconfiguration is transfered to a stochastic $\pi$-calculus model that can be simulated. Then the decentralized reconfiguration strategy has been realized on top of an agent-based simulation model of production lines. This implementation makes use of the freely available *Jadex* agent framework [3]. A realization of the *Coordination Middleware* for this agent platform is utilized [4]. Evaluations show the quick convergence to stable states and the reconfigurations only affect system partitions.

The outlined algorithm works by exchanging responsibilities with neighboring agents and by propagating change requests until all of them could be satisfied. Thus, the reconfiguration propagates through the system like a wave. An interesting aspect of the proposed mechanism is that configuration changes are preferred to happen in the neighborhood of the deficient agents. By keeping the reconfiguration local, other parts of the system are not impaired by a failure and can continue to run normally. Future work includes a more detailed elaboration of the combination of top-down design methodologies as promoted with the ODP and bottom-up design of coordination methods as proposed in SodekoVS. This will also include a comparison of their respective advantages and problems.

## References

[1] A. Hoffmann, F. Nafz, H. Seebach, A. Schierl & W. Reif (2010): *Developing Self-organizing Robotic Cells using Organic Computing Principles*. In: *Workshop on Bio-Inspired Self-Organizing Robotic Systems, Proceedings of the International Conference on Robotics and Automation 2010*.

[2] H. Seebach, F. Ortmeier & W. Reif (2007): *Design and Construction of Organic Computing Systems*. *IEEE Congress on Evolutionary Computation, 2007* , pp. 4215–4221.

[3] J. Sudeikat, L. Braubach, A. Pokahr, W. Renz & W. Lamersdorf (2009): *Systematically Engineering SelfOrganizing Systems: The SodekoVS Approach*. *Electronic Communications of the EASST* 17. ISSN 1863-2122.

[4] J. Sudeikat & W. Renz (2009): *Programming Adaptivity by Complementing Agent Function with Agent Coordination: A Systemic Programming Model and Development Methodology Integration*. *Communications of SIWN* 7, pp. 91–102. ISSN 1757-4439.