COMPLEX RESPONSIVE PROCESSES IN A MULTI-AGENT SYSTEM

Knowledge Sharing in a Self-Adaptive Multi-Agent System based on the principles of Complex Responsive Processes of Relating

Guido Willemsen
PhD student ISCTE-IUL/ISTA

Guido T. H. J. Willemsen
ISCTE-IUL/ISTA
University Institute of Lisbon
Lisbon, Portugal
Email: gwnou@iscte-iul.pt

Luis M. P. Correia
Dept. Informática, LASIGE,
Faculdade de Ciências
Universidade de Lisboa
Lisbon, Portugal
Email: luis.correia@ciencias.ulisboa.pt

Marco A. Janssen
School of Sustainability /
School of Complex Adaptive Systems
Arizona State University
Phoenix, USA
Email: marco.janssen@asu.edu

IARIA

Guido Willemsen

- Master in Business Administration

- 25 years professional experience as IT entrepreneur, project and interim manager (Viacom, ThermoFisher Scientific, Philips, Vodafone, Heineken)

- Special interest in Enterprise Architecture, Business Process Management Systems and Human Centric Automation

- Main challenge: how to apply social complex systems in autonomic automation?

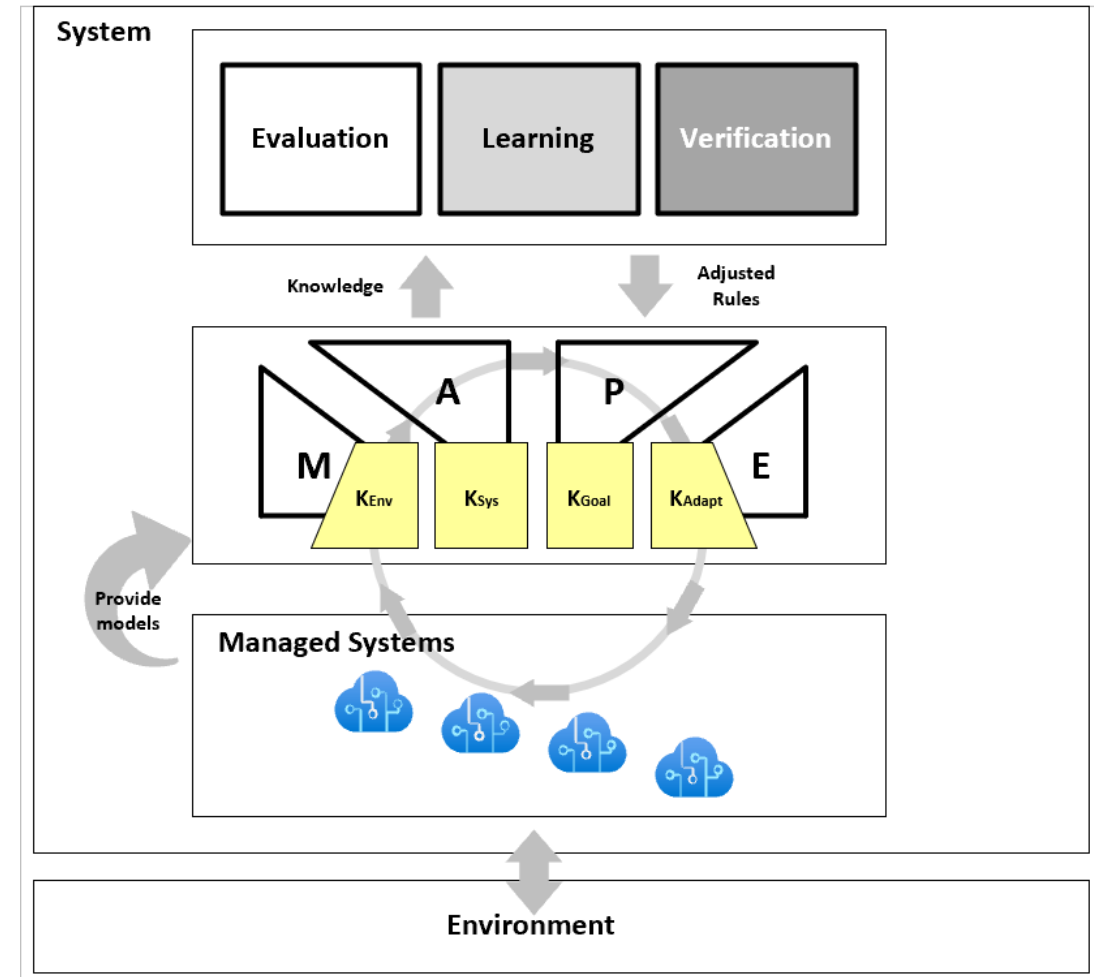# Multi-Agent Distributed Decision Making

- No central control

- Improve reaction time

- Sub-optimal, incomplete information

- Redundancy

- Good enough for now, safe enough to try

- Cooperation and collaboration required

CENTRALIZED   DECENTRALIZED   DISTRIBUTED
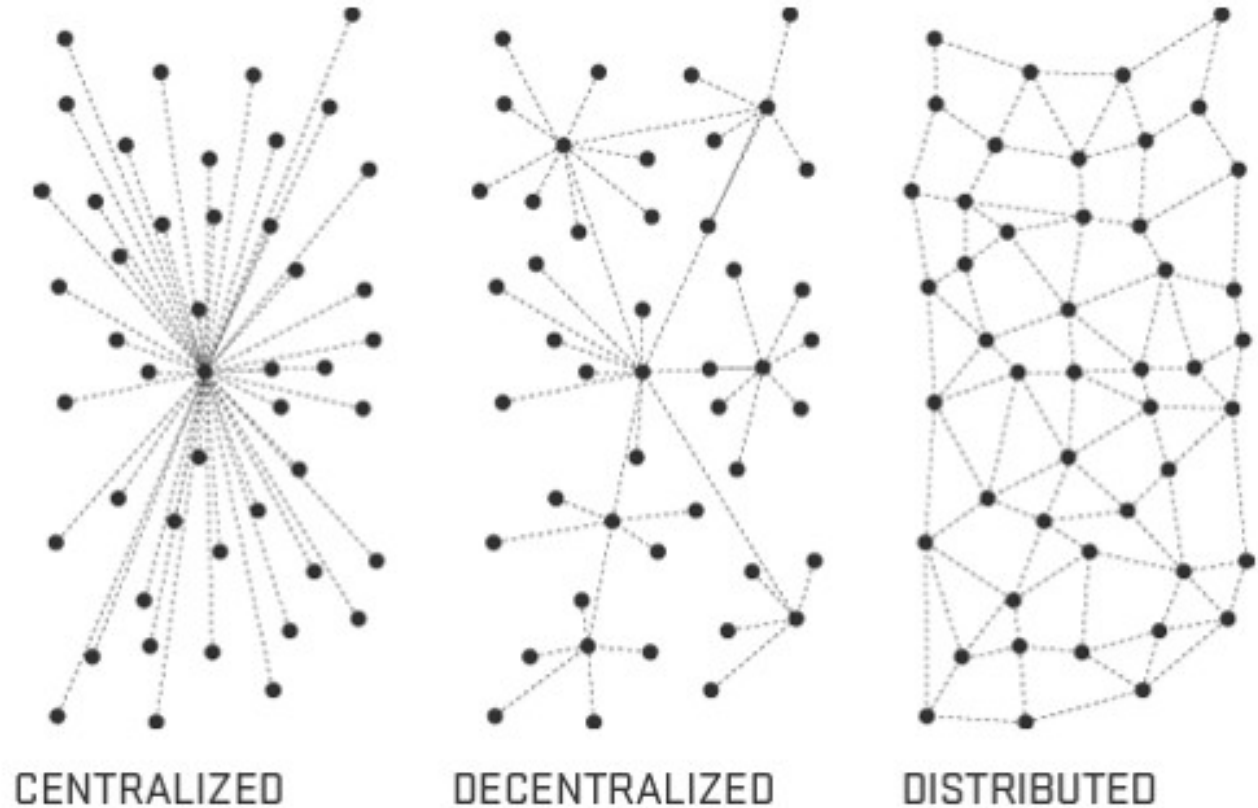
Source: P. Baran, 1964

Autonomic Automation – MAPE-K (ext.)

- Operational Systems are controlled by Agents

- Execution is triggered by an agent and instance-specific control cycle which makes the system adaptive
  - *Monitor*
  - *Analyze*
  - *Plan*
  - *Execute*

- Extension by Kloes *et al.:* Double loop learning by challenging the assumptions
  - *Evaluate*
  - *Learn*
  - *Verify*

- To maximize the advantage of this capability, the architecture of the operational system should be suitable

Source: V. Kloes,*et al.* (2018)

Frameself Architecture for MAPE-K

**Analyzer**

Policy Resolution
- -ValidatedPolicyList
- +DetectConflict()
- +ResolveConflict()

Policy Validator
- -AppliedPolicy
- -ValidatedPolicyList
- +checkPolicy()
- +validatePolicy()

Symptom Receiver
- +receiveSymptoms()
- -Symptom

Analyzer Manager
- -Symptom
- -KnowledgeObject
- -RFC
- -Policies
- -AppliedPolicy
- -ValidatedPolicyList
- +manageAnalysis()
- +applyPolicy()

RFC Generator
- -KnowledgeObject
- -Symptom
- -RFC
- +reasonOnKnowledge()
- +createRFC()

RFC Sender
- -RFC
- +sendRFC()

Knowledge Base Communicator
- -KnowledgeObject
- +readKnowledge()
- +writeKnowledge()

Planner

Policy Transformer
- -AppliedPolicy
- -ExecutablePolicy()
- +convertPolicy()

RFC Receiver
- -RFC
- -SubscriptionObject
- +subscribeMessages()
- +receivePublishedRFC()

Knowledge Base Communicator
- -Effector
- -Policy
- +readKnowledge()
- +writeKnowledge()

Planner Manager
- -RFC
- -InterpretedPolicy
- -SelectedPolicy
- -AppliedPolicy
- -ActionPlan
- +planAction()
- +SelectPolicy()

Policy Interpreter
- -Policy
- -InterpretedPolicy
- +employPolicy()

Plan Generator
- -Effector
- -(Applied) Policy?
- -ActionPlan
- -RFC
- +generateActionPlan()

Plan Sender
- -ActionPlan
- -PublishPlan()

Monitor

Event Filter
- -Event
- + discardEvent()

Event Masker
- -Event
- + ignoreEvent ()

Event Normalizer
- -Event
- + transformEvent ()
- + extendEvent ()
- + reformatEvent()

Symptom Sender
- -Symptom
- -publishSymptom()

Monitor Manager
- -Event
- -Symptom
- -KnowledgeObject
- -manageMonitoring()

Event Collector
- -Event
- + collectEvent ()

Knowledge Base Communicator
- -KnowledgeObject
- +readKnowledge ()
- + writeKnowledge ()

Event Aggregator
- + mergeDuplicates ()
- -Event

Symptom Generator
- -Event
- -KnowledgeObject
- -Symptom
- + inferSymptom ()

Knowledge Base

Models

Rules

Policies

Symptoms

RFCs

Plans

Sensors Description

Topology

Effectors Description

Executer

Knowledge Base Communicator
- -KnowledgeObject
- +readKnowledge()
- +writeKnowledge()

Workflow Engine
- -ActionSequence
- -ActionPlan
- -DeployedProcessFlow
- +CreateWorkflow

Action Formatter
- -Executable
- -Action
- +convertAction()

Action Dispatcher
- -Action
- -DestinationEffector
- +connectActionDispatcher()

Plan Receiver
- -SubscriptionObject
- -ActionPlan
- +subscribeMessages()
- +receivePublishedPlan()

Executor Manager
- -SubscriptionObject()
- -SystemPerformance
- -ActionSequence
- -AuthorizedObjects
- -ActionPlan
- -DeployedProcessFlow
- -ProcessState
- -Action
- -DestinationEffector
- -Executable
- -ProcessException
- -controlExecution()

Action Scheduler
- -ActionPlan
- -ActionSequence
- -SystemPerformance
- +scheduleAction()

Action Interpreter
- -ActionPlan
- -AuthorizationObjects
- +checkActionPlan()
- +checkAuthorization()

Process orchestrator
- -DeployedProcessFlow
- -ProcessState
- -Action
- -ProcessException
- +orchestrateProcess()

Patterns instead of predefined processes

- Self-Adapting capabilities enable context specific choices

- Predefined processes often lack context specificity or are too fine grained (and complicated)

- Simple, small scale patterns can be combined into an instant process

- The Monitor and Analysis phase defines the context from the environment and selects appropriate actions.

- The Planning and Execution phase assembles and implements the relevant Process Patterns

- To support this MAPE control cycle, the Knowledge base is kept up-to-date

Source: Oukharijane, J.; Ben Said, *et al*, 2019
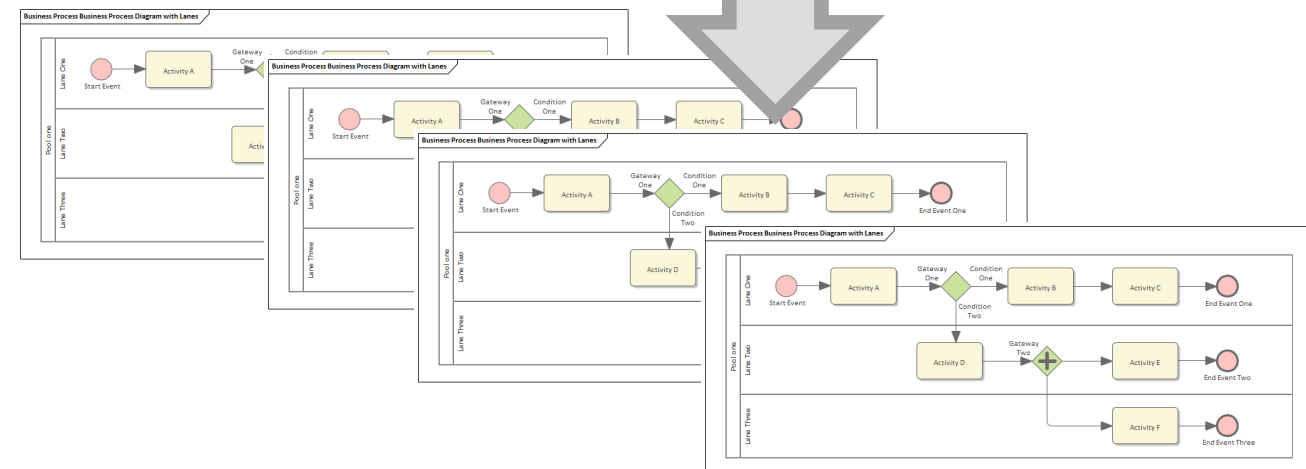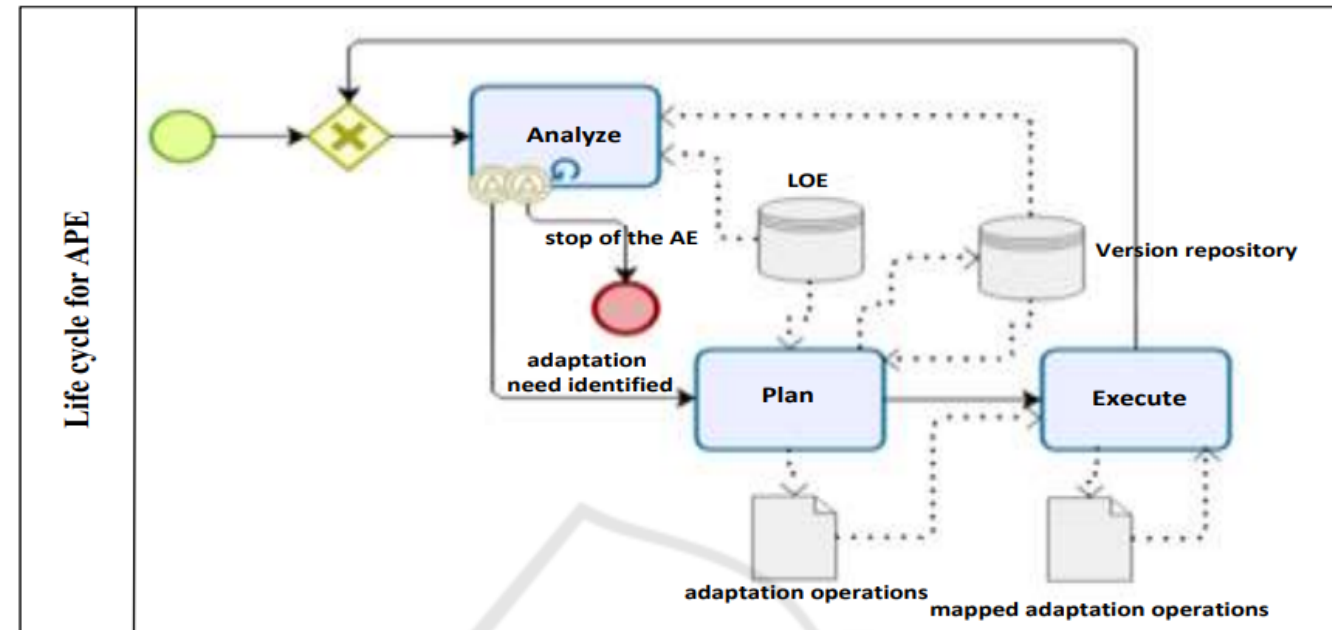
Self Adapting Multi-Agent Complex Systems

Operational Systems
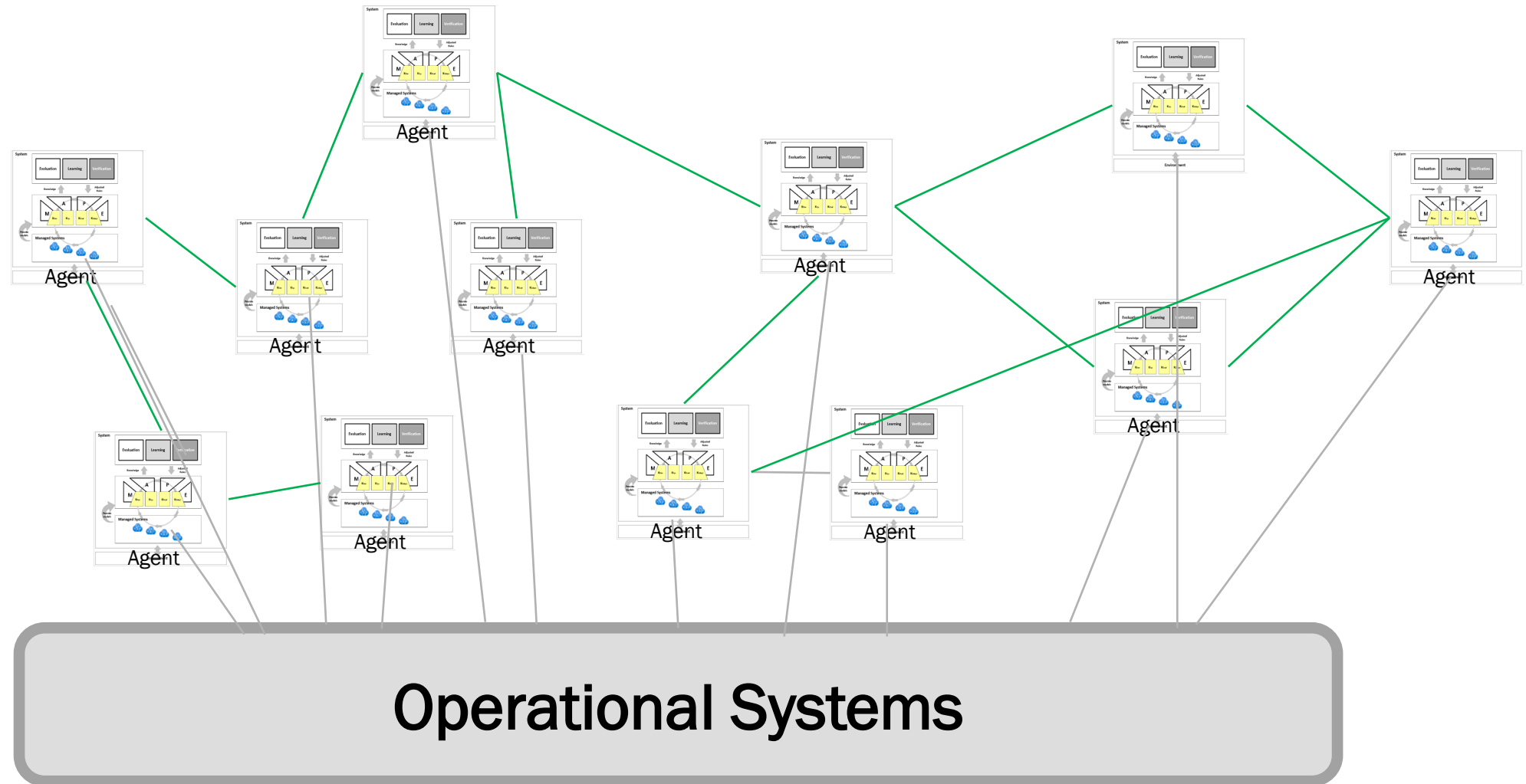
Complex Responsive Processes of Relating

- Zone of Complexity: a myriad complex of agent dynamics

- Gesture: Statement, authority or influence of the agent

- Response: Reaction of the interacting agent

- Dynamics arise in complex interactions between two or a collective of agents

- Gesture + Response = Social Act

- Patterns emerge in the absence of a plan -> base for resilience

- Clustering around agent based on power, knowledge, respect, contracts.

Source Stacey, R., 2008
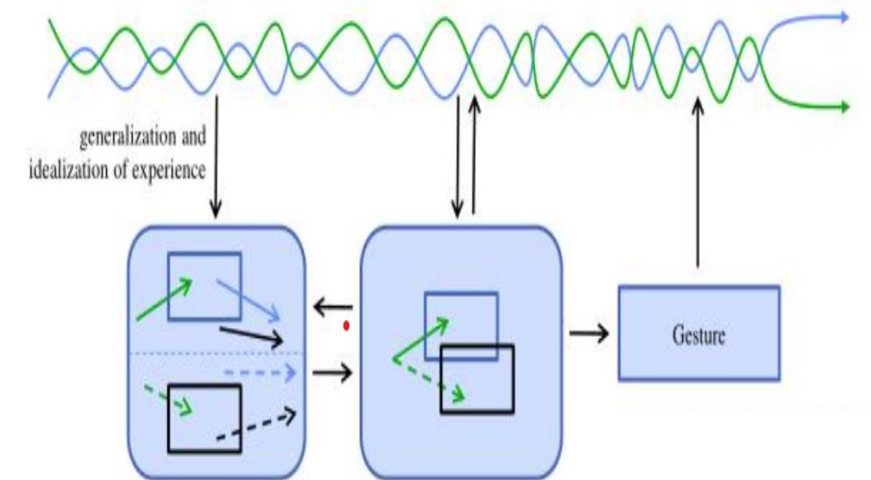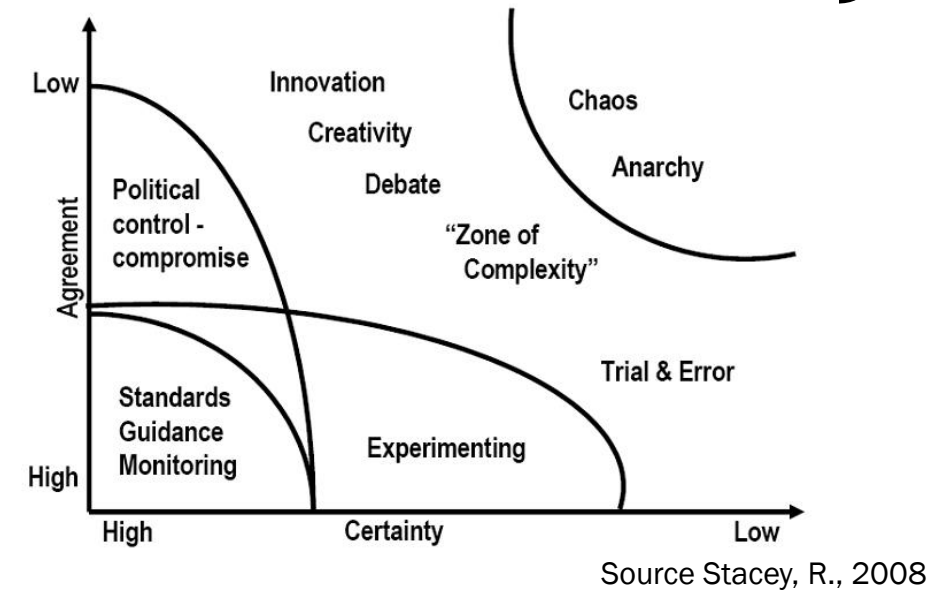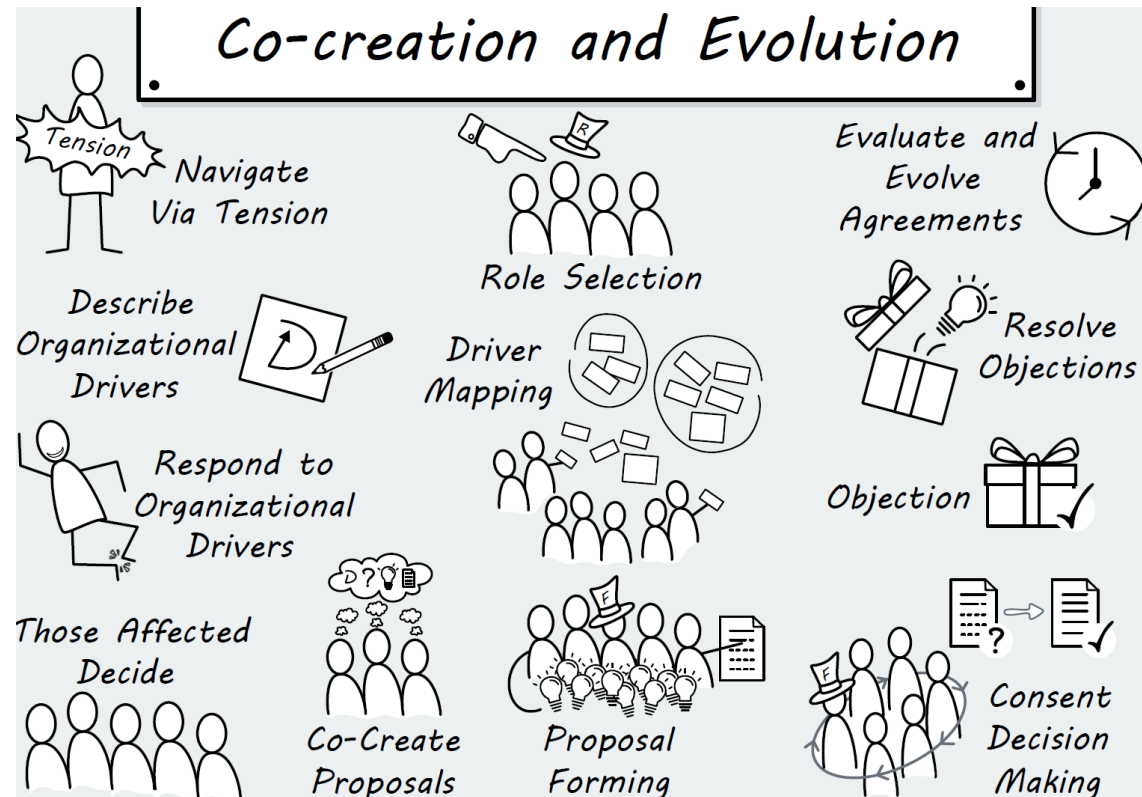
Source: J. Priest, *et al*, 2019

GESTURE & RESPONSE: SOCIOCRACY

Knowledge sharing through interest groups

Operational Systems
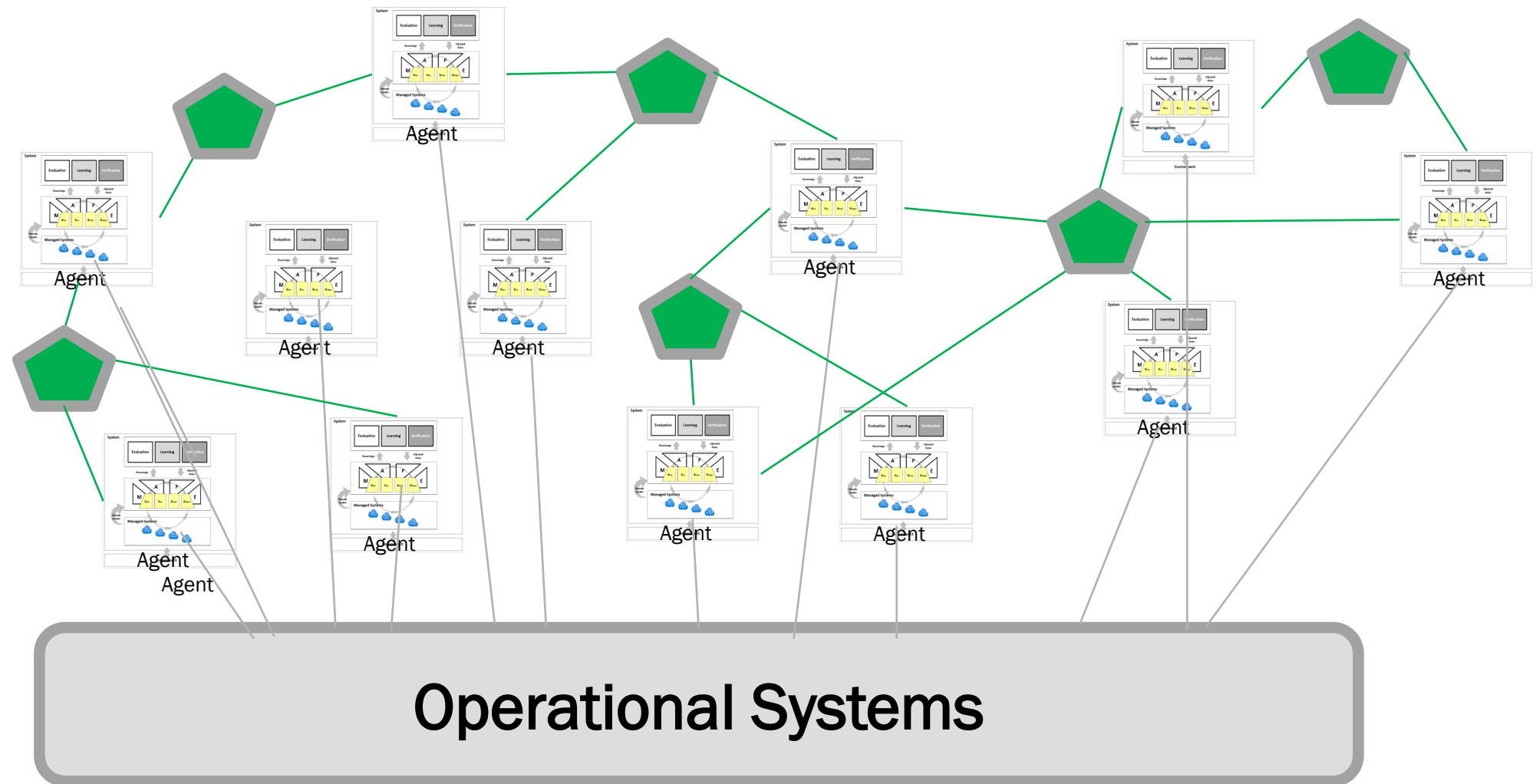
Social Gatherings in Different Time Scales
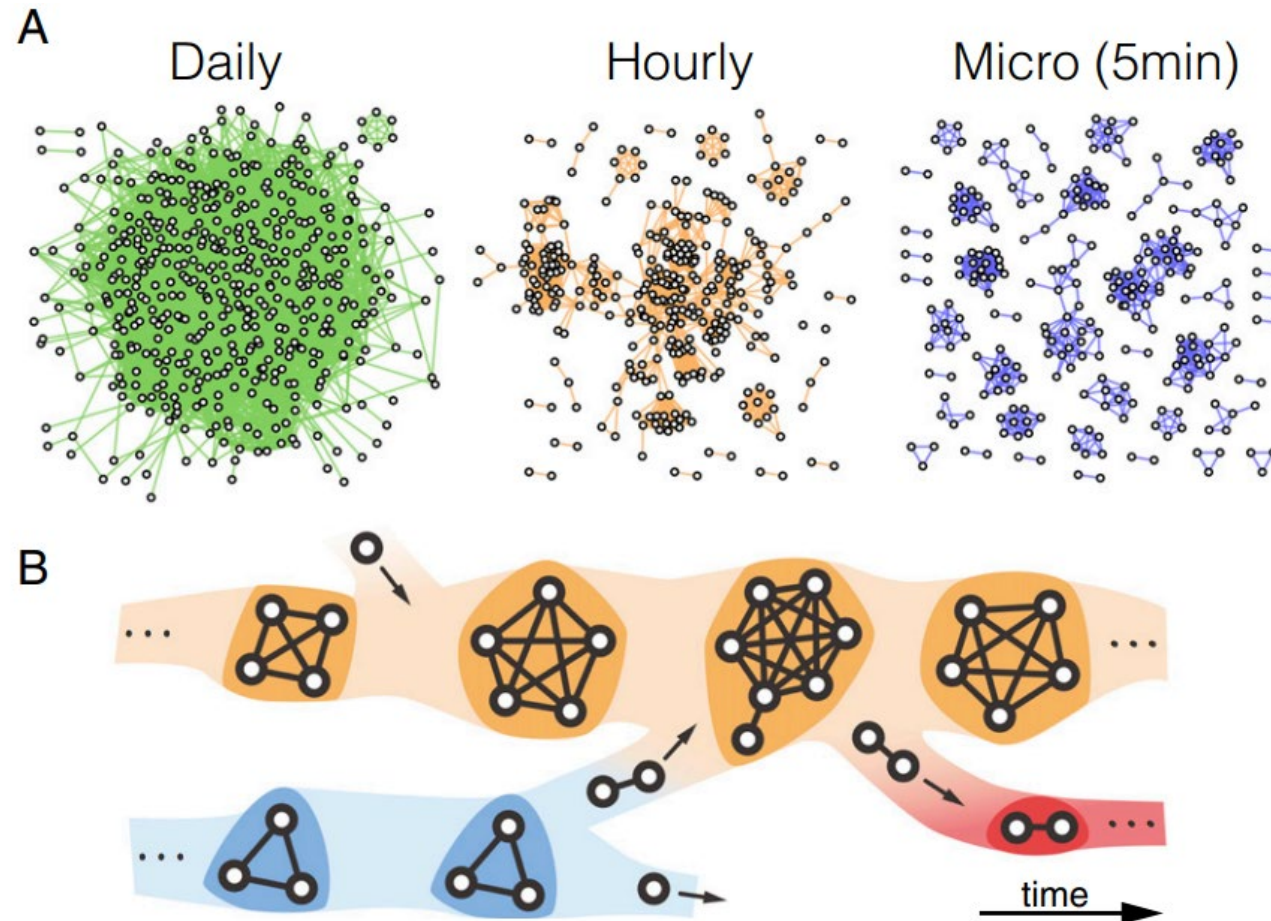
Continuous patterns of gathering and leave

Themes emerge, are challenged, adopted and rejected

Source: Sekaraa *et al*. 2016

CRP MAS ARCHITECTURE
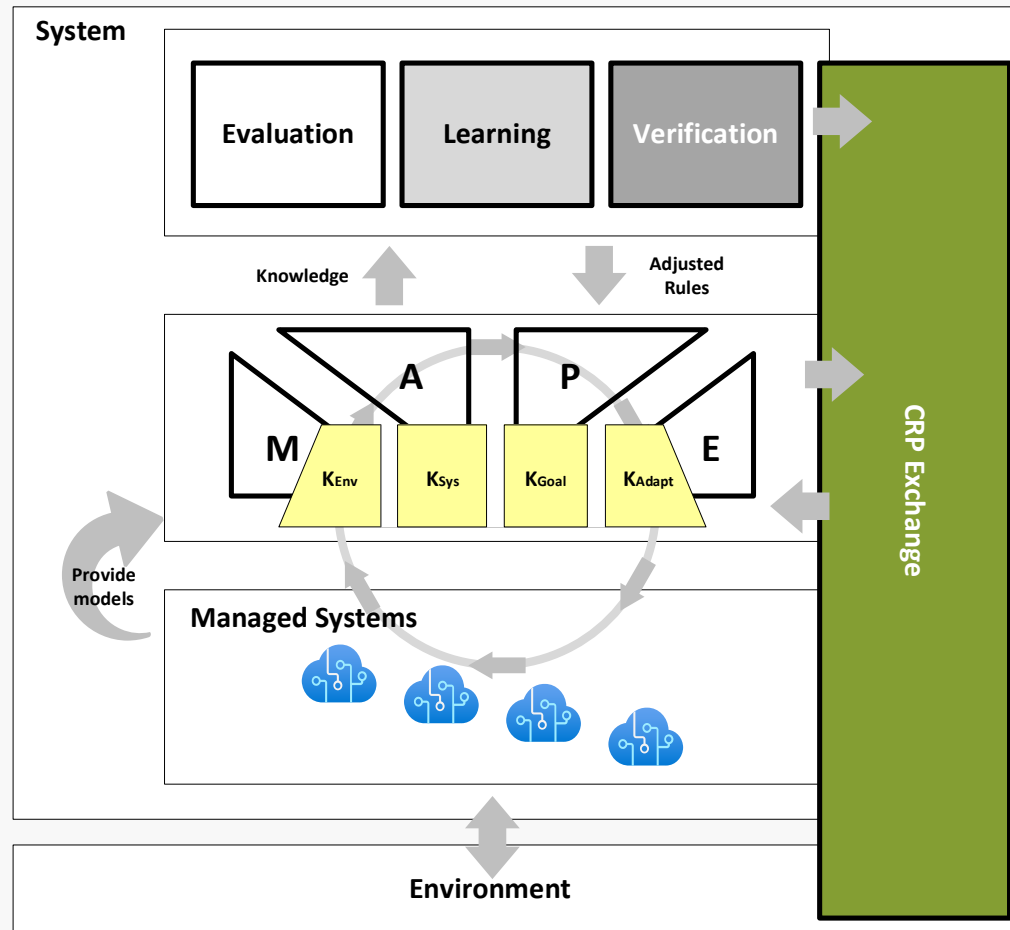
- How to integrate CRP concepts in the FrameseIf UML models?

- CRP extension to the (extended) MAPE-K model

- UML classes and methods required for CRP exchange

- Interaction model to update Knowledge Base based on CRP theme setting

- Application to operational model

MAPE-K CRP model as accelerator for Knowledge Creation
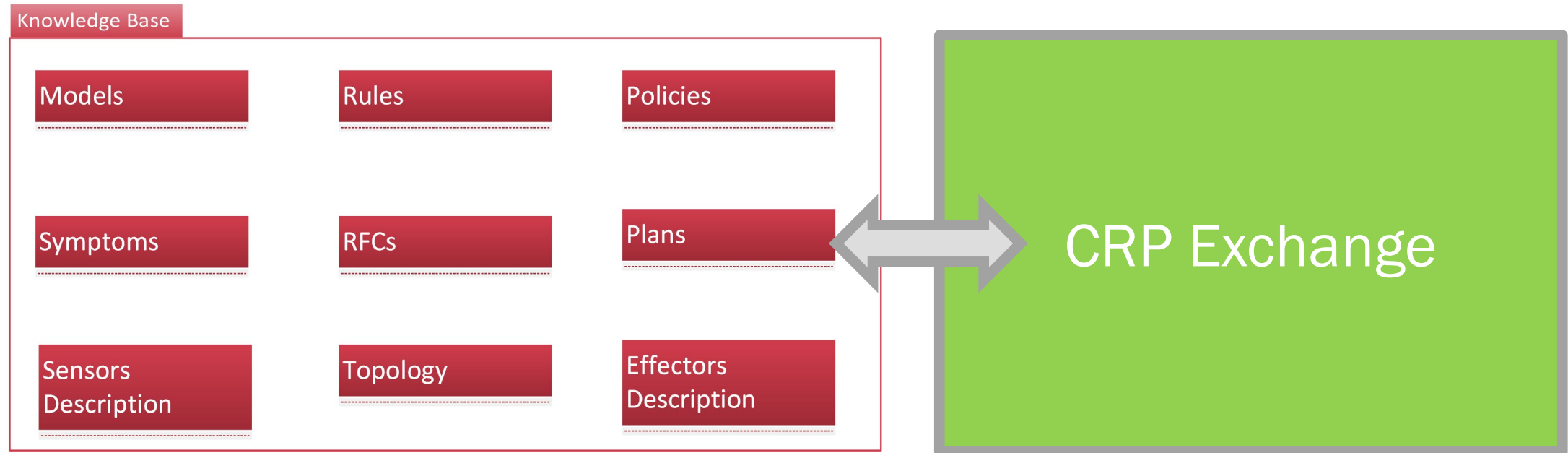
References

■ P. Baran, "On distributed communications networks." IEEE transactions on Communications Systems", no. 12(1), pp. 1-9, 1964.

■ V. Kloes, T. Goethel, and S. Glesner, "Comprehensible and dependable self-learning self-adaptive systems," Journal of Systems Architecture, no. 85-86, pp. 28-42, 2018.

■ M. B. Alaya and T. Monteil, "FRAMESELF: A generic autonomic framework for self-management of distributed systems - Application on the self-configuration of M2M architecture using semantic and ontology," in Int. Conference on Collaboration Technologies and Infrastructures, Toulouse (F), 2012.

■ J. Luoma, R.P. Hämäläinen, E. Saarinen, "Acting with systems thinking: complex responsive processes and systems intelligence", 2008, Manuscript

■ R. Stacey, The Stacey Matrix, http://www.gp-training.net/training /communication_skills/consultation/equipoise/complexity/stacey.htm, 2002

■ J. Priest, L. David and J. Cumps, "A Sociocracy 3.0 resource", https://sociocracy30.org, 2019

■ V. Sekaraa, A. Stopczynskia and S Lehmann, "Fundamental structures of dynamic social networks", PNAS, Vol 113, No.36, pp. 9977–9982, 2016

COMPLEX RESPONSIVE PROCESSES IN A MULTI-AGENT SYSTEM

Knowledge Sharing in a Self-Adaptive Multi-Agent System based on the principles of Complex Responsive Processes of Relating

Guido Willemsen
PhD student ISCTE-IUL/ISTA

Guido T. H. J. Willemsen
ISCTE-IUL/ISTA
University Institute of Lisbon
Lisbon, Portugal
Email: gwnou@iscte-iul.pt

Luis M. P. Correia
Dept. Informática, LASIGE,
Faculdade de Ciências
Universidade de Lisboa
Lisbon, Portugal
Email: luis.correia@ciencias.ulisboa.pt

Marco A. Janssen
School of Sustainability /
School of Complex Adaptive Systems
Arizona State University
Phoenix, USA
Email: marco.janssen@asu.edu