MATEC Web of Conferences **161**, 03024 (2018) https://doi.org/10.1051/matecconf/201816103024
13th International Scientific-Technical Conference on Electromechanics and Robotics "Zavalishin's Readings" - 2018

A decentralised solution for coordinating decisions in large-scale autonomic systems

Olga Melekhova[1,*], *Jacques* Malenfant[1], and *Roman* Mescheriakov[2]and *Aleksandr* Chueshev[3]

[1]Sorbonne Universités, UPMC Univ Paris 06, CNRS, UMR 7606 LIP6, 4 place Jussieu, 75005, Paris, France
[2]Tomsk State University of Control Systems and Radioelectronics, 634050, pr. Lenina, 40, Tomsk, Russia
[3]Polzunov Altai State Technical University, 656038, pr. Lenina, 46, Barnaul, Russia

Abstract. In this paper, we address the large-scale coordination of decisions impacting the consumption of a common shared resource of limited capacity by managed elements. We propose a decentralised token-based scheme, where each token represents a share of the resource. Our token-based protocol is meant to provide statistical guarantees on the average total resource usage and the average lateness of node actions due to the coordination. Experiments with the coordination of 10.000 autonomic managers have shown very good results for large spectrum of parameter values and system's regimes.

1 Introduction

The enormous growth in complexity of today's computer systems and applications has pushed IBM to propose a comprehensive effort, called *autonomic computing*, to automate their management [1]. In IBM's vision, an autonomic system is composed of *autonomic elements i.e.*, a *managed element* observed and adapted by an *autonomic manager* that operates a closed-loop control in the sense of control theory. Autonomic elements implement the well-known MAPE-k loop from [1]. As today's systems are more and more large-scale, new schemes are needed that allow a large number of autonomic managers to coordinate effectively with each others [2].

In this paper, we address the problem of coordinating decisions where the dependency among them is system-wide and implicit, namely the level of consumption of a central shared resource. The objective is to make sure that autonomic managers take decisions at the pace of changes in the need of their managed elements while the total required resource never exceeds its total capacity. More precisely, the coordination is meant to provide for statistical guarantees both on the global resource usage and on the fair distribution of the resource during the whole execution of the system. Coordination can be made to *emerge* [3–5] from local decisions, but making the current levels of resource usage of all managed elements available to every autonomic manager would inevitably leads to breach the time constraints of local control loops as the system scales.

We rather propose a token-based scheme, inspired from decentralised mutual exclusion algorithms [6]. The total amount of available resource is divided up in tokens that represent a right to use the corresponding amount of resource. Autonomic managers are organized into an overlay network [7] over which tokens circulate using random walks [6]. When an autonomic manager takes a decision that implies a raise in resource consumption, it first limits it to the amount represented by tokens it holds and then try to acquire new tokens to raise it further. When the autonomic manager takes a decision that lowers the consumption, tokens are reinjected in the overlay network for other autonomic managers.

Such a coordination scheme must exhibit both a fair distribution of the resource among autonomic managers, and the best possible global usage of the shared resource. These properties are highly influenced both by the quality of the interconnection provided by the overlay network and by the fluidity in the flow of tokens. In this paper, we propose an algorithm for the overlay network and we study a token-based coordination scheme exposing different parameters and show how their values can be chosen to balance the above objectives and scale. We report on experiments on systems of 10.000 autonomic managers that validate the approach and provide insights to tune the parameters to the application to get the best possible coordination.

2 Coordinating decisions in autonomic computing

Large-scale coordination of autonomic managers can be seen as a dynamic distributed decision-making problem. Centralised solutions, where all of the information is collected and decisions taken by a central entity do not scale. To scale, decisions must be made locally, and then coordinated by disseminating the information to every decision-making sites. But collecting locally all of the information at decision-time does not scale when the pace of decision-

*e-mail: melekhova@gmail.com

making is faster than the time required for the collection. Approaches to decouple the collection of information from the decision-making have been proposed, yet they do not scale further when the time for a required information to travel from the site where it is generated to the site where it is needed exceeds the deadlines of local decisions it impacts. In autonomic computing, where decisions must be made within the strict time limits imposed by the local control, the decoupling approach must also ensure the local availablility and freshness of data within these strict time constraints.

We presente our general token-based approach to large-scale coordination decomposes into three steps: (1) define a suitable semantics of tokens with regards to the local decision-making processes ; (2) link autonomic managers through a specific overlay network and adopt a suitable token communication policy ; (3) adopt a local decision-making process able to execute within the decision deadlines.

2.1 A use case in geotracking

Our use case considers the geotracking of large fleets of trucks coming from the French ANR project SALTY [8], however the scope can be different: robot/sensor networks for global resource charing, for example, communications, etc.. Typically, geotracking is used to follow trucks as to provide users with alerts as soon as they exit a predefined corridor or, perhaps more commonly, when approaching logistic bases to trigger resource allocation (doors, personal) for their arrival. Our industrial partner offers value-added services by making its customers receive their alerts through a unique centralised position processor called GeoHub[1]. Operators of fleets have trucks with GPS that send positions to the GeoHub through GSM networks.

2.2 Related work

Today, perhaps the most promising approaches are bio-inspired algorithms where global behavior *emerges* from purely local decisions [3, 9]; some applications have been conceptually explored [5] and only recently experimented for the first time in autonomic computing [10]. Crucially depending on information flows [11], emergence approaches for large-scale autonomic computing are also limited by the convergence delays in the broadcasting of information and in reaching an overall stability [12, 13]. Indeed, emergence approaches still require a lot of experiments, to which our work claims to contribute.

Louleiro *et al.* [14] address a large-scale resource allocation problem closely related to ours. Autonomic managers gather each workloads, then its autonomic manager uses its utility function to find the right decision. The authors use an *epidemic protocol* [12] to broadcast the individual workloads to all entities. Simulated on overlay networks of 500 entities, with degrees between 5 and 7, their approach requires more than 10 cycles of message exchanges between all of the nodes to propagate the information over the whole network. The number of cycles can be diminished by augmenting the degrees, but the number of messages per cycle would also augment. Picard *et al.* [15] propose a similar solution for distributed constraint satisfaction, without experimenting a dynamic version of the problem though. Other works on protocols for resource discovery in grids have shown that broadcasting approaches have an exponential complexity in the number of messages [16, 17].

Few related works address decision-making in the context of decentralised control where the system has strict timing constraints. Decentralised, networked control is still an open problem in control theory [18], even at the small-scale. Nevertheless, efforts to coordinate decisions in distributed autonomic systems have been made. IBM's teams have tackled the problem of energy management in data centers, but for small-scale systems only [19]. Work from Oliveira *et al.* [20] on synchronising autonomic control loops also targets the small-scale and doesn't take into account the timing issues of control. Many of the other works published recently are in the same vein [21–23].

Coordination in general has been studied in many fields, among which multi-agents systems (MAS), to such an extent that it is impossible to cover all of them. Compared to our proposal, most of these works address relatively small-scale systems or do not consider the timing issues of control [24]. For example, Gerber and Jung [25] consider a solution where local decisions are taken without delays, but use a global reasoning coordinating them that would hardly scale. Rustogi and Singh [26], as us, use local decision-making and show that giving locally more global information does not always lead to better decisions, concurring to more decoupling between local decision-making and information dissemination.

In the field of decision-making, reinforcement learning and *collaborative reinforcement learning* (CRL) has also been studied as a coordination mechanism, but most of the works are for a few decision-makers [27–30]

In conclusion, our contribution is the design and the implementation of new approaches and techniques in the field of decentralized decision-making subject to time constraints, more or less strict.

3 Overlay network

The coordination decisions in large-scale systems demands all nodes be interconnected to respect the fluidity in the information exchanging and the management of the dynamic network. We propose a self-organization approach based on decentralised algorithms for the construction and the maintenance of overlay network with the very weak distribution of node degrees. In more detail, the overlay network management algorithms and the results of experimentations are introduced in [31] and are excluded from our actual discussion.

[1]The term "GeoHub" belongs to Deveryware.

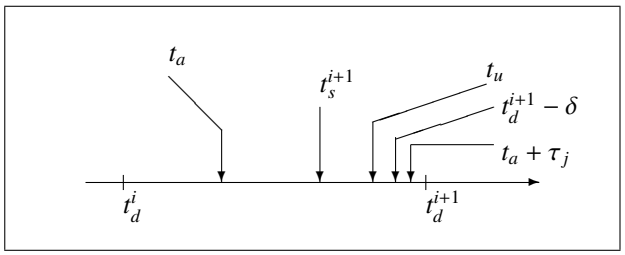

Fig. 1. Main instants when deciding and handling arriving tokens.

4 Token-based coordination

Based on the overlay network management algorithms, nodes exchange tokens controlling the (quantitative) access to the central shared resource, the GeoHub. Tokens circulate on the overlay network by random walks *i.e.*, token messages do not identify their destination but are rather sent to a randomly selected neighbour which either captures it for its own usage or immediately releases it and sends it again to a randomly selected neighbour among its own ones. Each token corresponds to the same rate of position sendings. To enforce its rate, the token has a period p_j and an earliest time of next utilisation t_u; each time $t \geq t_u$ a token is used to enable a position sending, its t_u is reset to $t + p_j$.

4.1 Coordination algorithms

The coordination decomposes in two main coordinated decision-making: handling position receptions and handling token arrivals. The coordinated decision-making occurs first when a position is received from the GPS of the truck through GeoHub, at which time the local decision-making process proposes a delay to the next position sending according to the needs of the geotracking applications and tries to plan it if the necessary token is available. When a token arrive, the autonomic manager must decide if it can capture or reemit it, and in the case of capture, plan a new position sending according to the most recently required time for it..

A central goal of the coordination is to minimise the waiting time for a token when an autonomic manager needs one and one is available somewhere. This goal is achieved by favouring, everywhere in the algorithms, the fluidity in the circulation of tokens against their retention. Considering the events introduced in the previous paragraph, the issue boils down to the conditions under which tokens can be kept after usage (at position reception time) or captured (at token arrival time). Our proposal is to use a unique parameter to control the fluidity, namely the time period τ_j during which a token can be kept before using it. The larger τ_j is, the more likely a token can be kept or captured. This criteria is therefore used both at token arrival time, to decide if it can be captured, and after a decision to decide if it can be kept.

Figure 1 illustrates the different important instants for some illustrative scenario. At time t_d^i, a position is received and the local decision-making proposes a delay for the next position sending that should occur at t_s^{i+1}. The coordinated decision is then to plan the next sending at $t_d^{i+1} \geq t_s^{i+1}$ that corresponds to the least t_u of a token the autonomic manager holds. Figure 1 shows a scenario where the autonomic manager does not have a token matching t_s^{i+1}, so $t_d^{i+1} > t_s^{i+1}$. When a token arrives at t_a, the autonomic manager captures it only if the conditions over τ_j hold and if it can use it to advance or better match t_d^{i+1}. The presented scenario illustrates the case where $t_a \leq t_s^{i+1} \leq t_a + \tau_j$ and $t_u \leq t_a + \tau_j$, which implies that the token can be used to advance t_d^{i+1}, hence it can be captured. Pragmatically, modifying t_d^{i+1} involves sending an order to the GPS through the GeoHub, hence the change must worth this effort (and respect a minimal delay). This requirement is formulated as the constraints $t_s^{i+1} + \epsilon \leq t_d^{i+1} - \delta$ and $t_u \leq t_d^{i+1} - \delta$ for some predefined ϵ and δ representing the shortest delay for modification and the smallest allowed one.

Algorithm 1 defines more precisely the coordinated decision at position reception time. First, the token that was previously captured to "cover" the current sending is marked as used by resetting its t_u to $t_d^i + p_j$. Next, the token is looked up to see if it must be released. Again, it will be kept only if it can be used for the next required sending, *i.e.* if the token can be used and the next sending planned within the next τ_j period of time, or more formally if $t_{u'} \leq t_d^i \wedge t_s^{i+1} \leq t_d^i + \tau_j$. If the token is kept, the next position sending can be planned (line 4). Otherwise, the autonomic manager will have to wait for the arrival of another, capturable token (line 7). Line 4 shows how the pragmatical parameter ϵ is used to force the next position sending to occur in at least ϵ seconds. Algorithm 2 implements the coordinated decision at token arrival time. To capture a token, the autonomic manager must need one and/or be able to use it during the next τ_j period. From the above discussion about Figure 1, the following conditions must be observed to (re)plan the next sending:

- $t_s^{i+1} \leq t_d^{i+1} - \delta$ *i.e.*, t_d^{i+1} can be advanced;
- $t_a + \epsilon \leq t_d^{i+1} - \delta$ *i.e.*, advancing it is feasible;
- $t_u \leq t_a + \tau_j$ *i.e.*, the token can be used;
- $t_u \leq t_d^{i+1} - \delta$ *i.e.*, and be used for that;
- $t_s^{i+1} \leq t_a + \tau_j$ *i.e.*, and the next sending would occur before τ_j.

Common data (algorithms 1 and 2)
f_j, tokens' position sending frequency
$p_j = 1/f_j$, the period of tokens
ϵ, minimal delay between a decision and the actual sending
δ, minimal modification in the time of the next sending justifying its change
t_u^j, earliest time for the next utilisation of token j
t_d^i, time of the i^{th} local decision
t_s^{i+1}, requested time for the next position sending
t_d^{i+1}, time for the next position sending as imposed by the coordination

Algorithm 1: Position reception time t_d^i

let *t be the token captured for this sending* **in**
  // mark the token for the current sending
1 set $t_{u'} \leftarrow t_d^i + p_j$
  // release the token, if required
2 **if** $t_{u'} \leq t_d^i \wedge t_s^{i+1} \leq t_d^i + \tau_j$ **then**
3 keep the token t
  // choose the time of next sending t_d^{i+1}
4 set $t_d^{i+1} \leftarrow max(t_s^{i+1}, t_{u'}, t_d^i + \epsilon)$
5 **else**
6 release the token t
  // no token left, so no next sending
7 set $t_d^{i+1} \leftarrow \infty$
end

Algorithm 2: Token t arrival time t_a

if $t_s^{i+1} \leq t_d^{i+1} - \delta \ \wedge \ t_a + \epsilon \leq t_d^{i+1} - \delta \wedge$
  $t_u \leq t_a + \tau_j \ \wedge \ t_u \leq t_d^{i+1} - \delta \ \wedge \ t_s^{i+1} \leq t_a + \tau_j$
then
  if *a token has been captured since* t_d^i **then**
    release the previous token
  keep the new token
  set $t_d^{i+1} \leftarrow max(t_s^{i+1}, t_u, t_a + \epsilon)$
else
  if *first token since* t_d^i **then**
    release the token
  else
    let t' *be the previously seized token* **in**
      if PREFER(t_d^{i+1}, t, t') **then**
        keep t and release t'
      else
        keep t' and release t
      end

Again to augment the fluidity of tokens, only one can be captured between two position sendings. The first to arrive will be if close enough to t_s^{i+1}. If another token has been captured before, then the autonomic manager will keep the one that it prefers. Two criteria for the function PREFER have been tested: always keep the previously captured token t' or keep t if its t_u is closer to t_d^{i+1}, in which case $t_d^{i+1} = t_s^{i+1}$ and then $t'_u < t_u \leq t_s^{i+1}$

5 Experimental results

Our experimental setup uses a discrete-event simulation of the GeoHub to receive each request for position sendings from autonomic managers and to plan a push of data at the requested moment. Autonomic managers are implemented as concurrent and distributed components in Java connected through RMI. 10.000 autonomic managers were distributed among 50 JVM (Oracle Java 7 update 51 on Mac 0s X 10.9), themselves running by group of 10 on 5 computers (Mac mini late-2012, with 2.6GHz Intel Core i7 and 16Gb 1600MHz DDR3) connected through Ethernet. Another computer (Mac pro mid-2012, with 2×2.4GHz Core Intel Xeon and 24Gb 1333MHz DDR3 ECC) runs the GeoHub simulator.

The overlay network topology was very stable, with each autonomic managers having an average of 15.5 neighbours, with a standard deviation of 1.5.

5.1 The core experiments

The core experiments were targeting two major questions: is the token-based coordination effective and fair enough, and can we characterise to some extent the spectrum of parameters of the algorithms that works well and in what situations? As a measure of effectiveness, we have looked at the average global resource usage as well as its standard deviation. To evaluate the fairness among the different autonomic managers, we have measured for each autonomic manager the average lateness in each of its position sendings (*i.e.* the delay between the requested time and the actual time the sending was done), and then we have looked at the distribution, the mean and the standard deviation of these average latenesses.

To cover different and meaningful situations, we have evaluated three main system's regimes: overloaded, even and underloaded. In each of them, the capacity of the Geohub has been fixed at 50 positions processed per second (*pps*).

As the number of tokens can influence the coordination, we have experimented with 400, 800, 1200 and 1600 tokens, giving token periods $p_j = 8, 16, 24$ and 32 seconds. Our previous work [31] has shown through simulations that the coordination works well when $\tau_j \approx p_j$. So experiments used as values of τ_j roughly $0.75p_j$, p_j, $1.125p_j$, $1.25p_j$ and $1.5p_j$. Each run lasted 1.5 hours. To compute the distribution of the rate of positions actually processed, we have measured it by periods of 30 seconds and then computed the average and standard deviation of these values. To compute the mean of lateness means, we have computed the mean for each autonomic manager, and then we have computed the mean of these means and the standard deviation.

As expected from our previous simulations, these experiments show that good results are obtained for values of τ_j between p_j and $1.5p_j$, but slightly dropping as τ_j increases over $1.25p_j$. The influence of the number of tokens appears less important, but a too low or a too large number of tokens doesn't seem to help, especially for small τ_j. Under the overloaded scenario, the global usage of the resource can be very close to the optimal 50 pps. The distribution of the mean latenesses also show a very good fairness, though it should be enhanced and finding new criteria for capturing tokens that would decrease the standard deviation and increase the fairness is certainly a goal of future research. For larger τ_j the distributions are very similar and relatively grouped, while for the smaller ones they tend to be less grouped, with the one for $\tau_j = 16$ having a somewhat long tail, hence favouring $\tau_j > p_j$ The variants on the PREFER function (Algorithm 2) did not show significant differences, so they are not included here.

The even and underloaded scenarios make it easier to assess the loss of resource imposed by the algorithm. In

the even scenario, we can see a larger standard deviation in global resource usage and a larger relative standard deviation in the mean of mean latenesses. This is due to the fact that when the pressure on the central resource is no longer controlled, when under 50, the full randomness of the requested positions shows directly. This is even more the case in the underloaded scenario, where basically no control is imposed by the tokens, but only a lower performance due to the potential unavailability of tokens at the right moment. Hence the means in these two cases are more important than the standard deviations. And we can observe that in the underloaded scenario, the mean resource usage and the mean of mean lateness are close to 33.3 and 0, showing a low impact of the token-based scheme *per se*.

6 Conclusion

In this paper, we have presented a new token-based scheme for large-scale coordination in decision-making for autonomic systems that provides for statistical guarantees on the quality and fairness of the coordination. Our main contributions lie in the quantitative nature of tokens experimented in our coordination scheme, in a novel simple local rule to capture and release tokens, and in large-scale experiments on systems of 10.000 autonomic managers. Experiments on our geotracking use case have shown that, on an overloaded scenario, our scheme can achieve as much as 98–99% in the global usage of the shared resource with a standard deviation of less than 1%. Fairness is also very good. In the same overloaded scenario, the ratio standard deviation over average is kept under 25%. Overall, very good coordination results are obtained for a wide spectrum of parameter values and system's regimes.

References

[1] IBM, White paper, IBM Corporation (2003), 1st Edition, 4th in June 2006

[2] J.O. Kephart, *Engineering decentralized autonomic computing systems*, in *SOAR'10* (ACM, New York, NY, USA, 2010), 1–2, ISBN 978-1-4503-0087-2

[3] L. Steels, *Towards a Theory of Emergent Functionality*, in *From Animals to Animats (First Int. Conf. on Simulation of Adaptive behaviour)* (MIT Press, 1990), 451–461

[4] T.D. Wolf, T. Holvoet, Tech. Rep. CW384, Department of Computer Science, KU Leuven (2004)

[5] T.D. Wolf, T. Holvoet, *Emergence versus self-organisation: different concepts but promising when combined*, in *Engineering Self Organising Systems: Methodologies and Applications* (Springer-Verlag, 2005), Vol. 3464 of *Lecture Notes in Computer Science*, 1–15

[6] A. Israeli, M. Jalfon, *Token Management Schemes and Random Walks Yield Self-Stabilizing Mutual Exclusion*, in *ACM Symposium on Principles (PODC'90)* (1990), 119–131

[7] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, IEEE Communications Surveys and Tutorials **7**, 72 (2005)

[8] O. Melekhova, M.A. Abchir, P. Châtel, J. Malenfant, I. Truck, A. Pappa, *Self-Adaptation in Geotracking Applications: Challenges, Opportunities and Models*, in *ADAPTIVE 2010* (IEEE, 2010), 68–77, ISBN 978-1-61208-001-7

[9] A. Ronzhin, A.I. Saveliev, O. Basov, S. Solyonyj, *Conceptual Model of Cyberphysical Environment Based on Collaborative Work of Distributed Means and Mobile Robots*, in *Interactive Collaborative Robotics - First International Conference, ICR 2016, Budapest, Hungary, August 24-26, 2016, Proceedings* (2016), 32–39

[10] T.D. Wolf, G. Samaey, T. Holvoet, D. Roose, *Decentralised Autonomic Computing: Analysing Self-Organising Emergent Behaviour using Advanced Numerical Methods*, in *IEEE ICAC'05* (2005)

[11] T.D. Wolf, T. Holvoet, *Designing Self-Organising Emergent Systems based on Information Flows and Feedback-loops*, in *IEEE SASO 2007* (2007)

[12] M. Jelasity, A. Montresor, O. Babaoglu, ACM Transactions on Computer Systems **23**, 219 (2005)

[13] I. Hegedũs, R. Ormándi, M. Jelasity, *Gossip-based Learning under Drifting Concepts in Fully Distributed Networks*, in *IEEE SASO'12* (2012), 79–88

[14] E. Loureiro, P. Nixon, S. Dobson, ACM Transactions on Autonomous and Adaptive Systems **7**, article 14:1 (2012)

[15] G. Picard, M.P. Gleizes, P. Glize, *Distributed Frequency Assignment Using Cooperative Self-Organization*, in *IEEE SASO'07* (2007)

[16] E. Jeanvoine, C. Morin, D. Leprince, *Un protocole de découverte de ressources optimisé pour l'allocation de ressources dans les grilles*, in *CFSE 2006* (2006), 49–59

[17] A. Iamnitchi, I.T. Foster, D. Nurmi, *A Peer-to-Peer Approach to Resource Location in Grid Environments*, in *IEEE HPDC-11)* (2002), 419

[18] V. Kumar, N. Leonar, A.S. Morse, eds., *Cooperative Control*, Vol. 309 of *Lecture Notes in Control and Information Sciences* (Springer-Verlag, 2005)

[19] J. Kephart, H. Chan, R. Das, D. Levine, G. Tesauro, F. Rawson, C. Lefurgy, *Coordinating Multiple Autonomic Managers to Achieve Specified Power-Performance Tradeoffs*, in *ICAC'07* (2007)

[20] F.A. de Oliveira Jr., R. Sharrock, T. Ledoux, *Synchronization of Multiple Autonomic Control Loops: Application to Cloud Computing*, in *Coordination'12* (Springer-Verlag, 2012), Vol. 7274 of *LNCS*, 29–43

[21] P. Vromant, D. Weyns, S. Malek, J. Andersson, *On Interacting Control Loops in Self-Adaptive Systems*, in *ACM SEAMS'11* (2011), 202–207

[22] D. Weyns, S. Malek, J. Andersson, *On Decentralized Self-Adaptation: Lessons from the Trenches and Challenges for the Future*, in *ACM SEAMS'10* (2010), 84–93

[23] L. Baresi, S. Guinea, G. Tamburrelli, *Towards Decentralized Self-adaptive Component-based Systems*, in *ACM SEAMS'08* (2008), 57–64

[24] G. Mahdi, A. Gouaïch, F. Michel, *Towards an Integrated Approach of Real-Time Coordination for Multi-agent Systems*, in *KES-AMSTA'10* (Springer-Verlag, 2010), Vol. 6070 of *LNAI*, 253–262

[25] C. Gerber, C. Jung, *Resource Management for Boundedly Optimal Agent Societies*, in *ECAI'98* (John Wiley & Sons, 1998), 1–6

[26] S. Rustogi, M. Singh, *The Bases of Effective Coordination in Decentralized Multi-agent Systems*, in *ATAL'98* (Springer-Verlag, 1999), Number 1555 in LNAI, 149–161

[27] L. Busoniu, R. Babuska, B. De Schutter, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **38**, 156 (2008)

[28] J.R. Kok, N. Vlassis, Journal of Machine Learning Research **7**, 1789 (2006)

[29] I. Dusparic, V. Cahill, *Research Issues in Multiple Policy Optimization Using Collaborative Reinforcement Learning*, in *ACM SEAMS'07* (2007), 18:1–7, ISBN 0-7695-2973-9

[30] J. Dowling, S. Haridi, in *Reinforcement Learning: Theory and Applications* (I-Tech Education and Publishers, 2008), chap. 8, 143–166

[31] O. Melekhova, Technique et science informatiques **33**, 31 (2014)