An Automated Approach to Management of a Collection of Autonomic Systems

Thomas J. Glazier and David Garlan
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA
tglazier@cs.cmu.edu · garlan@cs.cmu.edu

***Abstract*—Modern enterprise IT systems are increasingly becoming compositions of many subsystems each of which is an autonomic system. These individual autonomic systems act independently to maintain their locally defined SLAs but can take actions which are inconsistent with and potentially detrimental to the global system objective. Currently, human administrators intervene to resolve these conflicts but are challenged by complexity in the prediction of current and future states of the constituent systems and their managers, multiple conflicting quality dimensions which may change over time, combinatorially large configuration space across the set of constituent systems, and the time critical nature of the decisions to be made to prevent further degradation. To address these challenges, this paper proposes an approach that enables the creation of a higher level autonomic system, referred to as a *meta-manager*, that does not subsume the control functions nor does it directly orchestrate the actions of the sub-autonomic managers. Instead, we encapsulate and abstract the behavior of each subsystem as a parameterized adaptation policy which can be adjusted by the meta-manager to tune the adaptive behavior of the subsystem adaptation. We can effectively instantiate this idea by considering each of the subsystems as a player in a stochastic multi-player game against it's local environment, and synthesize an adaptation strategy using off-the-shelf tools for stochastic game analysis.**

I. Introduction

To meet the demands of high availability and optimal performance in dynamic environments, modern systems deploy autonomic or self-adaptation mechanisms. These autonomic mechanisms are responsible for continuously monitoring operating conditions and effecting changes in the system to ensure defined quality objectives are achieved. However, increasingly today's enterprise systems are compositions of many constituent systems, each an adaptive system. Each constituent system has its own defined objectives, reasoning methods, and adaptation tactics.

Typically, each autonomic manager operates to maintain locally defined service level agreements (SLAs), but their independent actions often lead to globally sub-optimal results. For example, the autonomic manager of an n-tiered enterprise system could be scaling up the capacity of the middle tier while the manager for the database tier is scaling down: at least one of them is likely to be inconsistent with the best global action. These globally sub-optimal behaviors are the result of the constituent systems having incomplete information about the current and future state of their environment, interdependency between systems propagating detrimental behavior, and changes in the global definition of optimal behavior due to shifting organizational priorities. Some of these sub-optimal results can be potentially catastrophic to the collective system. For example, the Northeast Blackout of 2003 was the result of a fault in a specific electrical grid, an autonomic system, which eventually cascaded to over 100 power plants and affected 10 million people in Ontario, Canada, and 45 million people in 8 US states with an estimated economic impact of $6.4 billion.[1]

Commonly, human administrators handle situations in which the collection of autonomic systems is behaving sub-optimally by reconfiguring the system manually. However, generating a plan to change the configurations of the constituent autonomic managers is a complex, challenging, and error-prone task. This task includes analysis of the current and potential future states of each constituent system each subject to multiple types of uncertainty, considering multiple quality dimensions for each constituent system, selecting an appropriate plan of action from a combinatorially large set of options, and performing all of these actions on a timescale appropriate to the context.

Given the complexity of managing such systems, a natural solution would be to introduce a global autonomic manager, which would oversee the operation of the subsystems. However, it is not entirely clear how one should do this. There are two readily available approaches; a subsumption approach [2] [3] that replaces the autonomic management of the subsystems with a higher-level manager that subsumes the control functions of the sub-autonomic managers, and an orchestration approach [4] that updates the knowledge models of the sub-autonomic managers with information relevant to their operation (e.g., the availability of a common resource or an action taken by another sub-autonomic manager). Both of these approaches have a number of problems.

First, it may not be possible to directly control the systems under the management of the individual autonomic managers – for example, if they are provided by third parties and provide only partial external control. Second, for a subsumption approach, the complexity of the analysis required to appropriately select or synthesize an adaptation strategy across the set of all managed resources, which will grow exponentially in the number of control actions and subsystem states, will likely make such a solution infeasible for non-trivial systems. Further, for an orchestration approach, the number of relevant pieces of information and the challenge of maintaining consistent information across a distributed system becomes prohibitive for any system of practical size. Finally, both fail to exploit the engineering advantages of separating the concerns of global management and local adaptation.

In this paper we propose an approach that addresses these shortcomings by enabling the creation of a higher level autonomic system, referred to as a *meta-manager*, that does not subsume the control functions nor does it directly orchestrate the actions of the sub-autonomic managers. The key idea of the approach is to exploit the homogeneity in the type of resource being managed and assume that the behavior of each autonomic subsystem is described by a parameterized adaptation policy. That is, each subsystem provides a set of parameters that allows the meta-manager to tune subsystem adaptation within a specified range of behaviors. The meta-manager can then synthesize a plan that determines the configuration settings of the subsystems most likely to improve global aggregate utility.

As we will show in this paper, we can effectively instantiate this idea by considering each of the subsystems as a player in a stochastic multi-player game against its local environment. The encapsulation of a subsystem as the set of adaptation strategies that it will use locally (dependent on its tuning parameters) allows us to encapsulate those subsystems, reducing complexity and preserving separation of concerns, while still allowing a range of global supervisory control. Specifically, the contribution of this paper is:

1) A game-theoretic approach to meta-management of a collection of autonomous subsystems that respects local autonomy, but allows global optimization through the synthesis of strategies for stochastic games.

The approach presented in this paper is applicable to collections of autonomic systems which are non-adversarial in nature and where each subsystem provides an interface to adjust the configuration parameters of the autonomic manager and can elaborate what adaptive action they will employ for a given state of the environment under a set of configuration parameters. While these applicability conditions apply to a significant subset of all collections of autonomic systems, the common threat to practicality is the scalability of the approach. Because our approach operates on an abstraction of the adaptive behavior and does not subsume control by performing the decision analysis for each subsystem, our approach is believed to scale to collections of autonomic systems with practical scale. This will be discussed further in Section V.

This paper is organized as follows: Section II gives an exemplar scenario of collections of autonomic systems in enterprise environments, Section III discuses the approach to synthesizing adaptation strategies in detail, Section IV provides the background on related work in relevant areas, and Section V outlines the future work to be conducted in this space.

II. Exemplar Scenario

This section describes an exemplar scenario for a collection of autonomic systems, which will be used as context to elaborate on the automated approach to managing a collection of autonomic systems, referred to as meta-management, and the proof-of-concept experiment that illustrates how this can be implemented using an off-the-shelf probabilistic model checker.

A large scale web system, like amazon.com, is built to handle a wide range of functional use cases including the ability to display products, manage the shopping cart process, and playback video for subscribers. Because each of these functional use cases has a different set of quality objectives, the system has been designed as a collection of sub-autonomic systems. For the purposes of this exemplar, we will assume that the system has the following four constituent systems: a shopping cart system, a video playback system, a common middle services tier, and a back end data services tier. Figure 1 is a simplified diagram of the system.

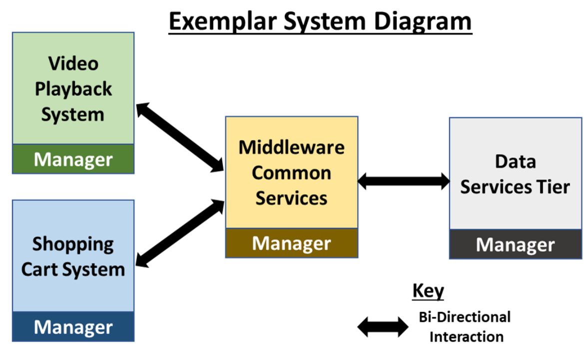

Figure 1: Exemplar System Diagram

Each of these constituent systems has an autonomic manager with the ability to make changes to the managed system to maintain performance against defined quality objectives in response to various environmental stimuli. However, autonomic managers do not typically react to the environmental stimuli directly. Instead, they track the values of key system metrics that directly relate to the desired quality of service objectives (see Table I), referred to as QoS properties, which are dependent upon both the environment and the architectural configuration of the managed system. Focusing the example on the shopping cart web system, the environment establishes the user load for the system, the environmental stimuli, and the autonomic manager tracks the average web page response time which is influenced by both the user load on the system and various architectural properties such as the number of servers and the fidelity[5] of the content being presented.

When the autonomic manager of the shopping cart web system determines that the quality of service objective for average page response time is either not currently being met or is unlikely to be met within a particular time horizon then the autonomic manager examines potential alternative configurations for the architectural properties of the managed system. For example, adding servers or lowering the fidelity of the content, or both, are potential alternative architectural configurations of the managed system that will influence the average page response time.

However, the autonomic manager cannot simply select *any* course of action that will improve performance against a quality of service objective; it must select the 'best' option under some constraints and preference conditions. For example, adding servers to the shopping cart system will also increase the run-time cost of the system, and lowering content fidelity [5] will

TABLE I. Exemplar Sub-System Properties

System	Primary Users	Tactics	Utility Dimensions	Config Parameters
Video System	Internet Users	Add Server, Change Fidelity	Response Time, Runtime Cost	Capacity Buffer
Shopping Cart	Internet Users	Add Server, Add Bandwidth, Change Fidelity	Response Time, Runtime Cost	Capacity Buffer
Middleware	Front End Systems	Add Server	Response Time, Runtime Cost	Capacity Buffer
Data Services	Middle Tier Systems	Add Server, Change Replication	Response Time, Runtime Cost	Capacity Buffer

decrease users' level of interaction with the system, both of which are undesirable and should be minimized.

Therefore, the organization can establish a preference for adding servers and increasing costs, up to a specific maximum, instead of decreasing fidelity of the content. This allows the autonomic manager the option of adding servers as long as the maximum cost constraint is not violated, but when the system reaches that value then the only architectural configuration option available, in this exemplar, is to reduce content fidelity.

In addition to the constraints and preferences that define the tradeoff space for each autonomic manager, there are potentially an additional set of configuration options available to the autonomic manager that serve as architectural guidelines which should be adhered to when possible. This is in contrast to a constraint which is applicable in all states of the managed system. For example, the autonomic manager for the shopping cart system might have a 'capacity buffer' setting which sets the guideline for how much spare processing capacity the web system should have available to handle small fluctuations in user load. In the event that the shopping cart system is running near its maximum cost, this is likely to be a state in which the 'capacity buffer' setting would no longer be applicable in order to comply with the maximum cost constraint which applies in all system states.

Further, there are also defined constants that are a result of something in the operating context of the autonomic manager and managed system and cannot be changed. For example, the cost per server per unit time for the shopping cart system is a constant defined by the context and potentially required for the autonomic manager to make appropriate adaptation decisions.

Finally, the adaptive actions of the shopping cart web system can affect the other constituent systems. An example of this interdependency is that the systems all draw from a common resource pool (e.g., allocated monetary budget or network bandwidth) that if overused by the shopping cart system (e.g., adding too many servers or overusing bandwidth) can ultimately lead to fewer adaptation choices for the other constituent systems.

While the autonomic managers for the managed systems in this exemplar are distinct with different implementations, architectural properties, preferences, constraints, constants, and guidelines, each of them will function similarly to the shopping cart example as described. Table I outlines the similarities and differences between the subsystems in the exemplar.

III. APPROACH

Our approach enables the creation of a higher level autonomic system, referred to as a meta-manager, that does not subsume the control functions, nor does it directly orchestrate the actions of the sub-autonomic managers. Instead our approach exploits the homogeneity in the type of resources being managed to assume that each sub-autonomic manager provides a set of configuration parameters that allow the meta-manager to tune the subsystem adaptation within a specified range of behaviors.

This is accomplished by understanding and defining the nature of an individual autonomic manager. A critical function of autonomic managers is the ability to apply adaptation actions to the managed system to improve its ability to meet the defined SLAs. However, when the autonomic manager applies an adaptation action, the result of that action is probabilistic. The new state of the managed system that results from the application of the adaptation action is influenced by the state of the system prior to the application of the adaptive action, the adaptation action applied, and the state of the environment. For example, in the exemplar, the local environment could increase the user load on the product catalog system which causes the average page response time to rise above a configured acceptable level. The resulting state of the system could be dependent upon several factors including if the managed system is operating near or well below maximum cost and whether or not the system characterises the state of the environment as a 'slashdot effect' or a transient spike in user load.

To attempt to select the 'best' adaptation action available, the autonomic manager measures each of the possible resulting states resulting from the application of each adaptation and, depending on the analysis, applies the one that will improve the managed systems ability to meet its defined SLAs. However, if one is able to enumerate the possible states of the managed system and the environment, it is possible to predetermine what adaptive action the autonomic manager will apply given any combination of managed system state and environment state. This enumeration of the combinations of managed system state, environment state, and adaption action is referred to as the adaptation policy.

However, the adaptation policies are influenced by the configuration of the autonomic manager itself. In the exemplar, if the configuration options were changed to significantly decrease the amount of monetary resources available, then the autonomic manager may no longer be able to select 'Add 1 Server' as an adaptation action, leaving only 'Reduce Content Fidelity'. This would result in a different adaptation action being selected for a specific combination of managed system state and environment state resulting in a different adaptation policy.

We can formalize this by, defining an autonomic manager as a tuple $\mathcal{A} = (C, E, F, D, \lambda, \pi, \sigma, \theta)$ where:

- C is a set of possible states of the managed system,
- E is a set of possible states of the local environment,
- F is the set of configuration options where each $f \in F$ is a function, $f(c) \to [0|1]$ where $c \in C$,
- D is the set of adaptation actions available to the autonomic system,
- $\lambda(c, c', d, e) = Pr(c_{t+1} = c' | c_t = c, d_t = d, e_t = e)$ is the probability that action $d \in D$ in system state $c \in C$ and environment state $e \in E$ will lead to state c' at time $t+1$,
- $\pi(c) \to [0, 1]$ where $c \in C$ is the utility function,
- $\sigma(c, e, F, D) \to d$ returns the adaptive action that maximizes utility where $d \in D$, $c \in C$ and is the current state of the managed system, and $e \in E$ is the current state of the environment and is defined as:

$$\sigma(c, e, F, D) = \underset{d \in D}{\operatorname{argmax}} \left\{ \sum_{c'} \lambda(c'|c, d, e) \pi(c'|c, d, e) H \right\} \tag{1}$$

where:

$$H = \min_{f \in F} f(c'|c, d, e) \tag{2}$$

The term H ensures that the configuration parameters for the system are satisfied. If any of the terms are unsatisfied, then the term will return 0, ensuring that strategy cannot produce a maximum utility score.

- $\theta(c, e, F, d) \to \mathbb{R}$ where $c \in C$ and $e \in E$ returns a measure of the utility generated for the managed system.

There are several ways of potentially instantiating θ including calculating the utility generated by the adaptation action with the maximum score, defined as:

$$\theta(c, e, F, d) = \pi(c'|c, d, e) \tag{3}$$

and the total amount of utility created by a set of configuration options, defined as:

$$\theta(c, e, F, d) = \sum_{c'} \lambda(c'|c, d, e) \pi(c'|c, d, e) H \tag{4}$$

where H is the same as previously defined. The choice of how to score the value of a given set of configuration parameters is important and can lead to different adaptive behaviors. For example, calculating θ based on only the state resulting from the best adaptation tactic can overvalue a small set of desirable states and possibly lead to poorer overall utility. However, selection of a total aggregate utility measure can fail to take advantage of the existence of a highly desirable state in favor of the possibility of broader utility gains.

The adaptation policy for the autonomic manager is then defined as: $P(F, D) = \{(c, e, \sigma(c, e, F, D)) : \forall c \in C \wedge \forall e \in E\}$. The set P contains the strategy that the autonomic manager will use for any state of the environment and any state of the managed system given the configuration. It is also necessary to understand the measure of utility generated by an adaptation policy which is defined as: $P^*(P, F) = \sum_{p \in P} \theta(p_1, p_2, F, p_3)$ where p_x references the x^{th} element (i.e., projection) of the tuple p.

By using the set P, the adaptation policy, one can then elaborate the extensive form of a game [6] between the autonomic manager and its local environment. For example, for each time step, t, the local environment will have a current state, e_t, this will be the 'move' of the environment. Further, at the same time step, the managed system will have a current state, c_t. Using e_t and c_t, one can use P to determine which adaptation strategy the autonomic manager will use, this is the 'move' of the autonomic manager. This process can repeat for as long as desired and over as many differing branches as need to define the extensive form of the game.

The fact that each of the adaptation policies is sensitive to the configuration options of the autonomic manager provides an opportunity to influence its adaptive behavior without subsuming its adaptation functions. This is how the meta-manager operates. The meta-manager evaluates how changes to the configuration options, referred to as meta-tactics, will change the adaptation polices of a sub-autonomic system and determines which configuration change will improve the global aggregate utility.

We can formalize this by defining the meta-manager as a tuple $\mathcal{M} = (A, N, Q, V, W, \Sigma)$ where:

- $A = \mathcal{A}_1, ..., \mathcal{A}_n$ is the set of all autonomic managers in the collection
- $N = \{P_{\mathcal{A}_1}, ..., P_{\mathcal{A}_n}\}$ is the set of adaptation policies for each system in the collection where N_x denotes the adaptation policy for autonomic system x,
- $Q = \{F_{\mathcal{A}_1}, ..., F_{\mathcal{A}_n}\}$ is the set of configuration options for each autonomic manager and Q_x will denote the configuration options for autonomic system x,
- $V = \{D_{\mathcal{A}_1}, ..., D_{\mathcal{A}_n}\}$ is the set of the adaptation tactics for each autonomic manager and V_x will denote the set of adaptation tactics for autonomic system x,
- W is the complete set of meta-adaptation tactics available to the meta-manager where each $w \in W$ is a function such that $w(F_x) \to F'_x$ where F_x and F'_x are sets of configuration options for autonomic system x and $W_x \subseteq W$ is the set of adaptation tactics relevant to autonomic system x and $|W_x| \geq 1$ as each W_x will have a 'null' adaptation tactic defined as $w(F_x) = F_x$,
- $\Sigma(A, Q, W, V) \to Z$ where $Z \subseteq W$ is a set of meta-adaptation tactics and is defined as:

$$\Sigma(A,Q,W,V) = \{S(W_i,Q_i,V_i) : i \le |A|\} \quad (5)$$

where:

$$S(W,q,v) = \underset{w \in W}{\operatorname{argmax}} \left[P^*(P(w(q),v),w(q))\right] \quad (6)$$

There are a number of candidate techniques that could be used to perform this analysis in practice. Due to the ability to elaborate an extensive form of a game for each autonomic subsystem a likely candidate is stochastic game analysis. Stochastic game analysis examines different probabilistic paths resulting from alternating actions taken between the individuals players to determine the value of a specific type of payoff (e.g., best base, worst case, expected case). This type of analysis can potentially handle the complexity of the required analysis and automate the process of selecting or synthesizing a meta-strategy most likely to improve global aggregate utility on time scale appropriate to the context. Further, this approach also enables us to leverage available off-the-shelf probabilistic model checking tools to generate the set of meta-tactics to be applied. The use of stochastic game analysis to synthesize adaptation strategies is well established, [7], but its specialization to this context presents a new challenges.

IV. Related Work

There are three key areas of relevant background and related work: (1) collections of adaptive systems, (2) strategy synthesis and assurance in autonomic systems, and (3) control theory for autonomic systems. Each of these areas will be discussed in more detail.

A. Collections of Autonomic Systems

To meet the complex functional objectives of organizations, autonomic systems are often composed together into an ensemble. The most common architectural approach to composing individual autonomic systems into an ensemble is an agent based approach as described in [8]. These individual systems interact with each other to achieve the functional objectives of the ensemble system. However, as noted in [9], agent based software systems suffer from a significant drawback: the behavior of the overall system is unpredictable because of the strong possibility of emergent behavior. This is problematic in contexts which require high degrees of assurance and predictability in the future states of the system.

The meta-management approach described in this paper provides a global coordination and management mechanism which partially addresses this drawback of agent based architectures. The parameterized adaptation policies from each of the sub-autonomic systems enable the meta-manager to compose a game representative of the complete collection of autonomic systems. This enables the use of stochastic game analysis to determine what the likely behavior of the collection will be. This provides a measure of assurance about the behavior of the collection. While this approach is not generally applicable across all ensembles of autonomic systems or agent based architectures, there is a significant subset of them for which this assumption is appropriate.

B. Control Theory for Autonomic Systems

The proposed approach to meta-management includes the creation of a higher level autonomic system with the goal of improving and providing assurance about the performance of a collection of autonomic systems. This approach establishes a form of hierarchical control for which there is an extensive body of work from control theory.

Control theory has established a common approach to the creation of hierarchical control systems which decomposes the complex behavior into individual units to divide the decision making responsibility. Each unit of the hierarchy is linked to a node in the tree and commands, tasks, and goals to be achieved flow down the tree from superior nodes, whereas sensations and commands results flow up the tree [2] [3]. This approach is commonly referred to as a subsumption architecture [2] [3]. This approach assumes that the management of the lower level components directly under the control of the higher level components, which may not always be the case in a practical system. Further, the complexity of the analysis necessary for a collection of autonomic systems would grow exponentially in the number of control actions and sub-autonomic system states making such a solution infeasible.

However a hierarchical control approach does serve as a guideline for the creation of a meta-manager for a collection of autonomic systems. The control theory approach to hierarchical control is dependent upon the ability to specify, typically in the form of differential equations, the dynamics of the system under control. This approach would be generally impractical, if not impossible, for collections of autonomic systems. Our approach of using the parameterized adaption policies from each of the sub-autonomic systems provides a practical method of specifying the behavior of a collection of autonomic system and serves a similar function.

C. Strategy Synthesis and Assurance in Autonomic Systems

Probabilistic model checking has provided encouraging initial results in improving the performance of and providing assurances about the outcomes of individual autonomic systems, [7]. Other work has focused on the challenges, frameworks, benchmarks, and approaches to providing these assurances at run-time including [10] and [11]. However, most of the existing work focuses on the use of probabilistic model checking in the context of a single autonomic systems. There is some limited work in the probabilistic model checking of collections or ensembles of autonomic systems including [12], [13], and [14]. This work focuses on verifying individual properties about the communication or negotiation protocols amongst agent based systems, not on their control or mitigation of globally undesirable behaviors. Our approach uses stochastic game analysis to synthesize an adaptation strategy for a collection of autonomic systems that tunes the configuration parameters of the sub-autonomic systems.

V. Discussion & Future Work

In this paper we introduced a game-theoretic approach to meta-management of a collection of autonomous subsystems that respects local autonomy, but allows for improvement of global aggregate utility. However, there is nothing about this approach that is specific to any individual off-the-shelf analysis tool nor strategy synthesis technique. For example, PRISM-Games has the limitation that it currently only works over zero sum games which may not be appropriate for all contexts. Or, to further enhance scalability, one could use a method with a non-exhaustive state space exploration (e.g., Monte Carlo Analysis). This partially enables the technique to have a significant degree of generality across a number of potential domains and types of collections of autonomic systems.

While any approach that synthesizes strategies is challenged by the scalability of the solution, individual analysis and strategy synthesis techniques have various methods of improving their scalability. For example, in stochastic multiplayer game analysis using PRISM-Games one can consolidate players into coalitions [15]. Each of these choices available for each strategy synthesis technique can enhance the scalability of the solution and provide timeliness appropriate to the context, but with the possibility of a loss of fidelity or assurance regarding the potential outcomes. By allowing for a variety of choices in this tradeoff space, our approach becomes more broadly applicable across the landscape of use cases for collection of autonomic systems.

Future work in this area will focus on (1) better understanding the relationship between the various composition techniques and choices and the potential loss of assurance in the final result, (2) better understanding of the applicability of various strategy synthesis techniques and their applicability to various use cases based upon the level of assurance they provide and the timeliness of their analysis, and (3) better understanding of how to represent and exploit various types of global knowledge that are likely to exist (e.g., correlations or dependencies between autonomic subsystems).

Another area of future work is to potentially loosen the goal of the collection of autonomic system from improving global aggregate utility to improving against a global objective. Setting the goal of a collection of autonomic systems to improve global aggregate utility allows for a certain set of assumptions, specifically that it is in the best interest of the collection for each system to individually maximize utility, that may not hold. For example, in a security context, it might be in the collections best interest to hold the attention of an attacker by sacrificing the currently compromised system to allow time for the other members of the collection to mitigate the threat to themselves.

VI. Conclusion

This paper presents an approach to meta-manage a collection of autonomic subsystems that respects local autonomy, but allows for improvement of global aggregate utility through the synthesis of strategies for stochastic games.

VII. Acknowledgements

This work is supported in part by award N000141612961 from the Office of Naval Research. Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research. Additionally, this material is based upon work supported by the NSA under Award No. H9823018D0008.

References

[1] “Northeast blackout likely to reduce us earnings by $6.4 billion,” http://www.andersoneconomicgroup.com/Portals/0/upload/Doc544.pdf, accessed: 2018-08-20.

[2] J. S. Albus, “A reference model architecture for intelligent systems design.”

[3] W. Findeisen, “Control and coordination in hierarchical systems,” 1980.

[4] IBM, “An Architectural Blueprint for Autonomic Computing,” IBM, Tech. Rep., 06 2005.

[5] B. Schmerl, J. Cámara, J. Gennari, J. Garlan, P. Casanova, G. A. Moreno, T. J. Glazier, and J. M. Barnes, “Architecture-based self-protection: Composing and reasoning about denial-of-service mitigations,” *HotSoS 2014: 2014 Symposium and Bootcamp on the Science of Security*, 2014.

[6] “Extensive form game,” https://en.wikipedia.org/wiki/Extensive-form_game, accessed: 2019-02-25.

[7] J. Cámara, W. Peng, D. Garlan, and B. Schmerl, “Reasoning about sensing uncertainty in decision-making for self-adaptation,” in *Proceedings of the 15th International Workshop on Foundations of Coordination Languages and Self-Adaptive Systems (FOCLASA 2017)*, 2017.

[8] N. R. Jennings, “An agent-based approach for building complex software systems,” *Communiations of the ACM Vol. 44, No. 4*, 2001.

[9] N. Jennings, *On Agent-Based Software Engineering*, 03 2000.

[10] D. Weyns, N. Bencomo, R. Calinescu, J. Cámara, C. Ghezzi, V. Grassi, L. Grunske, P. Inverardi, J.-M. Jezequel, S. Malek, R. Mirandola, M. Mori, and G. Tambrrellii, *Perpetual Assurances for Self-Adaptive Systems*. Springer, 2017, no. 9640.

[11] B. Cheng and et al., *Using Models at Runtime to Address Assurance for Self-Adaptive System*. Springer, 2014, no. 8378.

[12] P. Ballarini, M. Fisher, and M. Wooldridge, “Uncertain agent verification through probabilistic model-checking,” 2006.

[13] V. Shmatikov, “Probabilistic model checking of an anonymity system.”

[14] T. Chen, M. Kwiatkowska, D. Parker, and A. Simaitis, “Verifying team formation protocols with probabilistic model checking.”

[15] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, “Prism-games: A model checker for stochastic multi-player games,” in *Tools and Algorithms for the Construction and Analysis of Systems*, N. Piterman and S. A. Smolka, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 185–191.