AFRL-AFOSR-VA-TR-2019-0090

RASS: Resilient Autonomic Software Systems

Daniel Menasce
GEORGE MASON UNIVERSITY
4400 UNIVERSITY DR
FAIRFAX, VA 22030-4422

04/14/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/RTA2

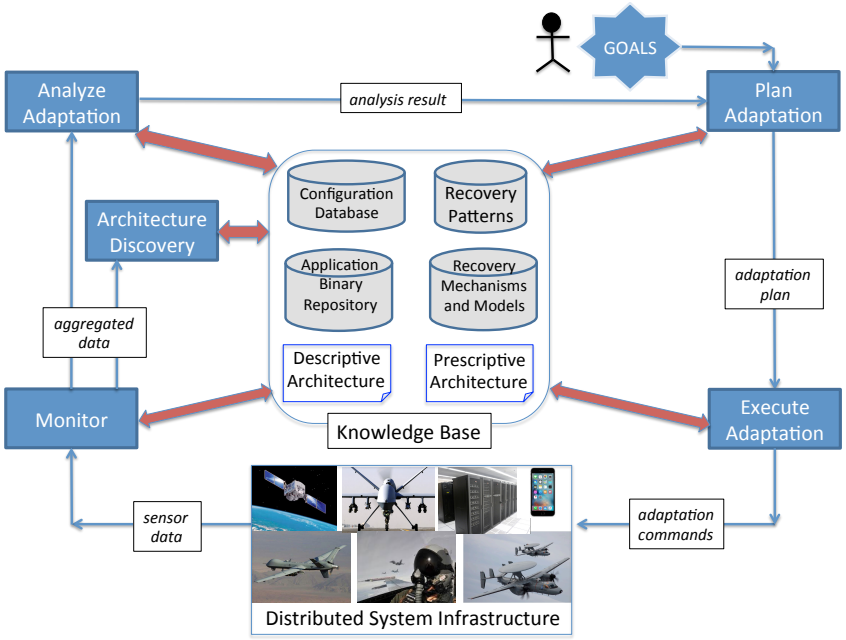
Arlington, Virginia 22203
Air Force Materiel Command

RASS: Resilient Autonomic Software Systems
Award no: FA9550-16-1-0030
Program Manager: Dr. Tristan Nguyen
PI and co-PIs: Daniel A. Menascé, Hassan Gomaa, and Sam Malek
Project web site: http://cs.gmu.edu/~menasce/rass/

Final Report: November 15, 2018
Period of Performance: October 15, 2015 to October 14, 2018

Contact: Prof. Daniel A. Menascé (PI), menasce@gmu.edu

Overview. USAF missions are generally supported by software systems that may become unexpectedly unavailable or degraded due to hostile activity or forces of nature. Under such occurrences, these systems need to dynamically self-adapt to continuously operate in the best possible manner. The systems that support the USAF missions are generally highly dynamic, distributed, very complex, heterogeneous, and need to be extremely resilient and dependable.



Thus, the *objective of this research* is to conceptualize, implement, and evaluate a framework called Resilient Autonomic Software Systems (RASS) inspired by the MAPE-K (i.e., **M**onitor, **A**nalyze, **P**lan, and **E**xecute based on **K**nowledge) paradigm for autonomic systems (aka self-managing systems).

As shown in the figure above, sensor data is obtained from a variety of systems (e.g., airplanes, UAVs, cloud systems, and cell-phones) by the Monitor component of RASS.

These data are aggregated and sent to the Analyze Adaptation component that determines whether self-adaptation is needed and why. The results of this analysis are passed to the Plan Adaptation component, which determines an optimal or near-optimal plan to automatically adapt the system. This plan is received by the Execution Adaptation component, which sends adaptation commands to the parts of the system that need to be adapted and/or recovered, and orchestrates the adaptation to ensure an appropriate and consistent recovery. Because the adaptation plan is driven by the system's software architecture, an Architecture Discovery module recovers the descriptive architecture (i.e., the one that reflects the actual system) or supplements the prescriptive architecture (i.e., the one used to guide the development) when one is not available or has been eroded by system evolution.

The Knowledge Base of RASS includes recovery methods and models that capture their performance overhead, failure recovery software patterns, adaptation patterns, a configuration database, an application binary repository to assist in recovering the software architecture of mobile applications, and the descriptive and prescriptive software architectures.

A significant challenge of our research is that all the activities described above need to be carried out in a completely decentralized fashion. This is essential for achieving high-resiliency and high-dependability.

Project Accomplishments.

Students. The RASS project recruited the four doctoral students listed below:

- Mahmoud Hammad, University of California at Irvine, received his PhD degree in August 2018 with the dissertation *Self-Protection of Android Systems from Inter-Component Communication Attacks*. PhD advisor: Prof. Sam Malek.
- Jason Porter, George Mason University, received his PhD degree in May 2018 with the dissertation *Decentralized Runtime Architecture Discovery and Testbed for Adaptation and Failure Recovery of Large Dynamic Distributed Systems.* PhD advisors: Profs. Daniel A. Menascé and Hassan Gomaa.
- Emad Albassam[1], George Mason University, received his PhD degree in May 2017 with the dissertation *A Model-Based Approach for Self-Healing and Self-Configuration in Component-Based Software Systems.* PhD advisor: Prof. Hassan Gomaa.
- Noor Bajunaid, George Mason University, advanced to candidacy with the dissertation proposal *Modeling and Optimization of Performance and Reliability of Distributed Autonomic Systems*. PhD advisor: Prof. Daniel A. Menascé.

The entire RASS Research Group met once a week for two hours with the Mason and UC Irvine members (i.e., Prof. Sam Malek and Mr. Mahmoud Hammad) participating over Skype. During these meetings we discussed research ideas and members of the group gave presentations regarding their progress. Additionally, Profs. Menascé and Gomaa met once a week with Mason members of the project and each individual student met individually at least once a week with his/her PhD advisor.

[1] Dr. Albassam was not supported as a GRA by the AFOSR grant. However, he was a full-fledged member of the RASS research group.

The RASS group worked on the following interrelated research areas:

Automatic generation of architectural models of mobile applications. Autonomic software systems leverage an abstract representation of the software, often in the form of an architectural model, to manage and adapt the system at runtime. Prior research assumes that these models are developed manually by software engineers. Manual construction of such models is difficult and labor-intensive, hence error prone [26]. Moreover, models used for runtime adaptation of software quickly become obsolete due to changes in the software. In this research, we developed novel static program analysis and dynamic monitoring techniques for the Android system to automatically obtain and maintain a precise architectural model without human intervention [30,35-36,39,41]. Static analysis techniques are used to extract the static architectural model of the Android app from its byte-code [25], whereas dynamic monitoring is used to update the runtime model and keep it synchronized with the running system [21-22]. Due to Android's commercial success as well as the fact that it is an open-source platform, Android is increasingly adopted in many mobile military systems. Our research shows that Android systems are also vulnerable to a variety of security attacks [10,19]. The techniques developed in this thrust of the RASS research facilitates construction of autonomic Android systems that are resilient to changes in system resources (e.g., available battery) as well as unexpected dynamic conditions (e.g., security attacks). We demonstrated how architectural models obtained in this way can be used to improve the security posture of an Android device by reducing the unnecessary privileges granted to its applications [5,28,42].

Robust Decentralized Discovery of Software Architectures of Distributed Systems. As indicated above, autonomic systems typically rely on the knowledge of a software architectural model to plan for adaptation. This thrust of the research considers highly dynamic distributed software systems for which the software architecture is not known in advance because it either suffered erosion (i.e., the software changed over time and architecture prescriptive documents were not updated) or no prescriptive documents were ever produced. We designed and implemented a system called DeSARM (Decentralized Software Architecture DiscoveRy Method) that uses a gossip-like mechanism based on propagation of aggregated message traces to uncover the structural view of the distributed architecture [33]. We conducted experiments to assess the convergence of the process, its termination under stable conditions, its accuracy in discovering the architecture, and its overhead relative to a centralized approach. The experiments were successful and determined the robustness of the method with respect to component and node failures [33].

Analysis and Optimization of Resiliency Mechanisms. Distributed systems can be made more dependable by replacing software components by other components or composite components that implement one or more fault-tolerant mechanisms or add security to the communication between components. Examples of such fault-tolerant mechanisms include checkpointing, replication with or without replica diversity, replication with diversity and voting, consensus building protocols in the presence of byzantine failures. Fault-tolerant mechanisms typically add some level of performance overhead in terms of added computational requirements and/or added communication requirements and latency. We derived and validated a comprehensive analytic model for checkpointing of

concurrent software systems [9, 29]. Our models are the first to consider components that are heterogeneous in terms of (1) resource demands and (2) failure probability distribution. Our first models were based on Markov Chains [29]. While fully automated, their computational complexity was very high for a large number of heterogeneous components. We then developed a novel model based on Queuing Networks [9]; this model is several orders of magnitude faster than the model based on Markov Chains and produces the same results. Because of its speed, the new model can be used at run-time to autonomically determine the optimal checkpointing rate of each component as the environmental conditions change. We implemented such an optimizer based on hill-climbing and showed that it can find the optimal solution by visiting a very small fraction of the solution space [9].

We also developed policies and models for redundant submission of requests to multiple servers [12]. The first replica to finish cancels its identical requests in other servers. We carried out an extensive analytic and simulation analysis to determine the benefits of each policy and the optimal number of replicas in each case.

Decentralized Recovery Patterns and Connectors. This research investigated how recovery patterns, recovery and adaptation connectors, and a decentralized MAPE manager (as shown in the above figure) can be designed and integrated to provide self-healing and self-configuration in distributed software architectures. A distributed software architecture is composed of several interconnected architectural structure patterns (such as client/server, hierarchical, decentralized control, hierarchical control) and distributed communication patterns (such as asynchronous, synchronous with reply, subscription/notification, brokered). A recovery pattern defines how components in an architectural pattern can be dynamically relocated and recovered to a consistent state after a component has failed. Connectors in component-based software architectures interconnect components and encapsulate a communication protocol. In this research, we extend connectors with adaptation and recovery capabilities to assist with component recovery and adaptation [17, 18, 23, 37]. All communication with a component passes through one or more recovery and adaptation connectors, which maintain a copy of messages sent to the component. When a component failure occurs, a decentralized MAPE manager works with the relevant recovery connectors, using the appropriate recovery pattern(s), to analyze the cause of component failure, plan for component recovery, and execute the plan by reconfiguring the system, assigning the recovered component to a different node, and re-establishing its communication with neighboring components to ensure a consistent system state. We also designed reusable secure connectors for secure architectures [16, 40]. Our project researched Software Product Lines for IoT applications [6], recovery and adaptation connectors [23], secure connectors [7], and applications for smart environments [38].

We designed and built a middleware called DARE (**D**istributed **A**daptation and **Re**covery) [24] that uses the services of DeSARM [33] to implement failure recovery and adaptation. DARE is a decentralized, integrated adaptation and recovery framework for providing both self-healing and self-configuration properties to complex and highly dynamic component-based software architectures. DARE was deployed and evaluated in a computer cluster with 30 nodes. The experimental results demonstrated DARE's

capability of dynamically adapting and recovering components in a decentralized environment.

Automated Evaluation of Recovery and Adaptation Frameworks. Evaluating the performance of distributed software systems is very challenging especially in the presence of failures and adaptation. Of particular interest to us is self-healing and self-adaptation middleware such as DARE [24] that detects failures of distributed software systems, analyzes their root causes, devises plans to recover from these failures, and executes these plans. Recovery plans may trigger software architecture adaptations, which may be also initiated by the need to maintain performance and availability goals. In order to provide a comprehensive and scientific evaluation of recovery and adaptation middleware such as DARE, we developed TESS [11], a testbed for automatically generating distributed software architectures and their corresponding runtime applications, deploying them to the nodes of a cluster, running many different types of experiments involving failures and adaptation, and collecting in a database the values of a variety of failure recovery and adaptation metrics. Queries can then be run against the database to provide a thorough and scientific analysis of the efficiency and/or effectiveness of a recovery and adaptation framework.

Autonomic Fog and Cloud Resource Management and Elasticity Control. Many computer systems, such as Internet datacenters and cloud computing environments, consist of a multitude of servers that process user requests. Incoming requests that find all servers busy have to wait until a server becomes idle. This type of queuing system is known as a G/G/c system and has been extensively studied in the queuing literature under steady state conditions. We derived closed-form analytic models for non steady-state G/G/c systems [8] and used these models to design autonomic controllers for multi-server systems subject to *workload surges* of generic shapes during which the average arrival rate of requests exceeds the system's capacity. We performed an extensive evaluation of the these controllers using a G/G/c simulator that we developed (see https://cs.gmu.edu/~menasce/ggc-sim/). We used both synthetic arrival streams and traces from a Google cluster as inputs to the simulator. The results showed that our autonomic controllers are able to successfully vary the number of servers to mitigate the impact of the workload surges [4, 20].

We also studied the problem of job partitioning between fog and cloud servers and developed a performance and cost analytic model [3] and corresponding tool (see https://cs.gmu.edu/~menasce/fogqn/) that we are now using in the design of an autonomic controller that dynamically decides the optimal fraction of processing that should be allocated to a fog server and to the cloud. In [15] we derived analytic models that can be used for capacity planning of IaaS clouds that offer multiple service classes.

Analytic Modeling of Distributed Systems. One of the methods used to design autonomic controllers is to use analytic performance models, typically based on queuing networks, of the system to be controlled. These models are used to compute the values of utility functions to be optimized by the controller. One of the drawbacks of this approach is that it requires expertise from performance engineers. The RASS project developed a framework that automatically derives analytic performance models from relationships

between inputs (e.g., arrival rates) and outputs (e.g., response time and throughput values) [27, 32, 34]. Other modeling activities carried out under the RASS project include the derivation of TDQN, a trace-driven queuing network algorithm [2], and an analytic queuing model for multi-tiered software servers [31] along with an autonomic controller that determines the optimal number of servers in each tier.

Moving Target Defenses. Moving Target Defense (MTD) has recently emerged as a game changer in the security landscape due to its proven potential to introduce asymmetric uncertainty that gives the defender a tactical advantage over the attacker. Many different MTD techniques have been developed, but, despite the huge progress made in this area, critical gaps still exist with respect to the problem of studying and quantifying the cost and benefits of deploying MTDs. In fact, all existing techniques address a very narrow set of attack vectors, and, due to the lack of shared metrics, it is difficult to quantify and compare multiple techniques. We proposed a quantitative analytic model for assessing the resource availability and performance of MTDs, and a method for maximizing a utility function that captures the tradeoffs between security and performance. The proposed model can be applied to a wider range of MTDs and operational scenarios that allow limits to be imposed on the maximum number of resources that can be reconfigured to improve availability and performance. The analytic results were validated by simulation and experimentation, confirming the accuracy of our model [1, 14].

Publications: the RASS project generated 42 publications, all of them acknowledging AFOSR funding. There are two additional publications under review that also acknowledge AFOSR funding.

1. W. Connell, D.A. Menasce, M. Albanese, Performance Modeling of Moving Target Defenses with Reconfiguration Limits, IEEE Tr. Dependable and Secure Computing, accepted for publication on November 15, 2018.
2. D. Menasce and S. Bardhan, TDQN: Trace-Driven Analytic Queuing Network Modeling of Computer Systems, The Journal of Systems & Software, Elsevier, Vol. 147, January 2019, pp. 162-171.
3. U. Tadakamalla and D.A. Menasce, FogQN: An Analytic Model for Fog/Cloud Computing, Proc. 1st Workshop on Managed Fog-to-Cloud (mF2C), joint with 11th IEEE/ACM Intl. Conf. Utility and Cloud Computing, Zurich, Switzerland, December 17-20, 2018.
4. V. Tadakamalla and D.A. Menasce, Model-Driven Elasticity Control for Multi-Server Queues Under Traffic Surges in Cloud Environments, Proc. 2018 Intl. Conf. Autonomic Computing, Trento, Italy, September 3-7, 2018.
5. M. Hammad, J. Garcia, and S. Malek, SALMA: Self-Protection of Android Systems from Inter-Component Communication Attacks. 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2018), Montpellier, France, September 2018.
6. V. Tzeremes and H. Gomaa, A Software Product Line Approach to Designing End User Applications for the Internet of Things, Proc. 13th Intl. Joint Conf. Software Technologies (ICSOFT 2018), Porto, Portugal, July 2018.

7. M. Shin, H. Gomaa, and D. Pathirage, A Software Product Line Approach for Feature Modeling and Design of Secure Connectors, Proc. 13th Intl. Joint Conf. Software Technologies (ICSOFT 2018), Porto, Portugal, July 2018 (**best paper award**, invited to submit an extended version to a Spring LNCS publication).
8. V. Tadakamalla and D.A. Menasce, Analytic Model of Traffic Surges for Multi-Server Queues in Cloud Environments, Proc. Workshop: Cloud Performance and Reliability, IEEE CLOUD 2018 Conf. July 2-7, 2018, San Francisco, CA, USA.
9. N. Bajunaid and D.A. Menasce, Efficient Modeling and Optimizing of Checkpointing in Concurrent Component-Based Software Systems, J. Systems and Software, Vol. 139, May 2018, pp. 1-13.
10. Mahmoud Hammad, Joshua Garcia, and Sam Malek. A Large-Scale Empirical Study on the Effects of Code Obfuscations on Android Apps and Anti-Malware Products. Proc. Intl. Conf. Software Engineering (ICSE), May 2018, Gothenburg, Sweden.
11. J. Porter, D.A. Menasce, H. Gomaa, and E. Albassam, TESS: Automated Performance Evaluation of Self-Healing and Self-Adaptive Distributed Software Systems, 9th ACM/SPEC Intl. Conf. Performance Engineering (ICPE), Berlin, Germany, April 9--13, 2018.
12. D.A. Menasce and Noor Bajunaid, Performance Evaluation of Heterogeneous Multi-Queues with Job Replication 2017 Computer Measurement Group Conf. (imPACt 2017), New Orleans, LA, November 6-9, 2017.
13. Joshua Garcia, Mahmoud Hammad, and Sam Malek. Lightweight, Obfuscation-Resilient Detection and Family Identification of Android Malware ACM Transactions on Software Engineering and Methodology (TOSEM), November 2017 (Accepted)
14. W. Connell, D.A. Menasce, M. Albanese, Performance Modeling of Moving Target Defenses, 2017 Moving Target Defense (MTD) Workshop, Oct 30th - Nov 3rd, 2017, Dallas, Texas.
15. M. Carvalho, D.A. Menasce, and F. Brasileiro, Capacity Planning for IaaS Cloud Providers Offering Multiple Service Classes, Future Generation Computer Systems, Elsevier, 77 (2017), pp. 97–111.
16. M. Shin, H. Gomaa, and D. Pathirage, Reusable Secure Connectors for Secure Software Architectures, Proc. 4th International Workshop on Model-Driven and Component-Based Software Engineering (ModComp), MODELS Conference, Austin, Texas, September 2017.
17. H. Gomaa, E. Albassam, and D.A. Menasce, Run-time Software Architectural Models for Adaptation, Recovery and Evolution, Proc. 12th International Workshop on Models@runtime, MODELS Conference, Austin, Texas, September 2017.
18. E. Albassam, H. Gomaa, and D.A. Menasce, Model-Based Recovery and Adaptation Connectors: Design and Experimentation, ICSOFT 2016, CCIS 743, E. Cabello et al. (Eds.), Lecture Notes in Computer Science, Chapter 6, Springer Intl. Publishing AG, pp. 1–24, 2017, DOI: 10.1007/978-3-319-62569-0_6.
19. Joshua Garcia, Mahmoud Hammad, Negar Ghorbani, and Sam Malek. Automatic Generation of Inter-Component Communication Exploits for Android Applications. 11th Joint Meeting of the European Software Engineering

Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, September 2017.
20. Venkat Tadakamala and D.A. Menasce, Analysis and Autonomic Elasticity Control for Multi-Server Queues Under Traffic Surges, 2017 IEEE Intl. Conf. Cloud and Autonomic Computing (ICCAC), Tucson, AZ, USA, September 18-22, 2017.
21. Alireza Sadeghi, Reyhaneh Jabbarvand, and Sam Malek. PATDroid: Permission-Aware GUI Testing of Android. 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, September 2017.
22. Reyhaneh Jabbarvand and Sam Malek. muDroid: An Energy-Aware Mutation Testing Framework for Android. 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, September 2017.
23. E. Albassam, H. Gomaa, and D.A. Menasce, Variable Recovery and Adaptation Connectors for Dynamic Software Product Lines, 10th Intl. Workshop on Dynamic Software Product Lines (DSPL 2017), Sevilla, Spain, September 25-29, 2017.
24. E. Albassam, J. Porter, H. Gomaa, and D.A. Menasce, DARE: A Distributed Adaptation and Failure Recovery Framework for Software Systems, 14th IEEE Intl. Conf. Autonomic Computing (ICAC 2017), Columbus, OH, July 17-21, 2017.
25. Alireza Sadeghi, Hamid Bagheri, Joshua Garcia, and Sam Malek. A Taxonomy and Qualitative Comparison of Program Analysis Techniques for Security Assessment of Android Software. IEEE Transactions on Software Engineering (TSE), Vol. 43, No. 6, June 2017.
26. Alireza Sadeghi, Naeem Esfahani, and Sam Malek. Ensuring the Consistency of Adaptation through Inter- and Intra-Component Dependency Analysis. ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 26, No. 1, May 2017.
27. Mahmoud Awad and Daniel A. Menasce, Deriving Parameters for Open and Closed QN Models of Operational Systems Through Black Box Optimization, 8th ACM/SPEC Intl. Conf. Performance Engineering, L'Aquila, Italy, April 22--26, 2017.
28. Mahmoud Hammad, Hamid Bagheri, and Sam Malek. DELDroid: Determination and Enforcement of Least-Privilege Architecture in Android. Intl. Conf. Software Architecture (ICSA), Gothenburg, Sweden, April 2017.
29. Noor Bajunaid and Daniel A. Menasce, Analytic Models of Checkpointing for Concurrent Component-Based Software Systems, 8th ACM/SPEC Intl. Conf. Performance Engineering, L'Aquila, Italy, April 22--26, 2017.
30. Hamid Bagheri, Joshua Garcia, Alireza Sadeghi, Sam Malek, Nenad Medvidovic, "Software Architectural Principles in Contemporary Mobile Software: from Conception to Practice," J. Systems and Software 119, 2016, pp. 31-44.

31. Daniel A. Menasce and Noor Bajunaid, "Modeling and Optimization of Multitiered Server-based Systems," 2016 Computer Measurement Group Conference (imPACT 2016), La Jolla, CA, November 7-10, 2016.
32. Mahmoud Awad and Daniel A. Menasce, "A Knowledge Base to Support the Automatic Derivation of Performance Models of Operational Systems," 2016 Computer Measurement Group Conference (imPACT 2016), La Jolla, CA, November 7-10, 2016.
33. Jason Porter, Daniel A. Menasce, and Hassan Gomaa, "DeSARM: A Decentralized Software Architecture Discovery Mechanism for Distributed Systems." 11th International Workshop on Models@run.time (MODELS 2016), Saint-Malo, France, October 4, 2016.
34. Mahmoud Awad and Daniel A. Menasce, "Performance Model Derivation of Operational Systems Through Log Analysis," IEEE Intl. Symp. Modeling, Analysis and Simulation of Computer Systems and Telecommunication Systems (MASCOTS 2016), Imperial College, London, UK, September 19-21, 2016.
35. Bradley Schmerl, Jeff Gennari, Alireza Sadeghi, Hamid Bagheri, Sam Malek, Javier Camara, and David Garlan, "Architecture Modeling and Analysis of Security in Android Systems," Proc. 10th European Conf. Software Architecture, Istanbul, Turkey, September 2016.
36. Sam Malek, Kyle Canavera, and Naeem Esfahani, "Automated Inference Techniques to Assist with the Construction of Self-Adaptive Software," in Managing Trade-offs in Adaptable Software Architectures, eds. Ivan Mistrik, Nour Ali, John Grundy, Rick Kazman, and Bradley Schmerl, Elsevier.
37. Emad Albassam, Hassan Gomaa, and Daniel A. Menasce, "Model-based Recovery Connectors for Self-Adaptation and Self-Healing," Proc. 11th Intl. Joint Conf. Software Technologies (ICSOFT 2016), July 24-26, 2016, Lisbon, Portugal.
38. Vasilios Tzeremes and Hassan Gomaa, "A Multi-Platform End User Software Product Line Meta-model for Smart Environments," Proc. 11th Intl. Joint Conf. Software Technologies (ICSOFT 2016), July 24-26, 2016, Lisbon, Portugal.
39. Hamid Bagheri, Alireza Sadeghi, Reyhaneh Jabbarvand Behrouz, and Sam Malek. "Practical, Formal Synthesis and Autonomic Enforcement of Security Policies for Android." Proc. 46th Annual IEEE/IFIP Intl. Conf. Dependable Systems and Networks (DSN 2016), Toulouse, France, June 2016.
40. Michael Shin, Hassan Gomaa, and Don Pathirage, "Reusable Secure Connectors for Secure Software Architectures," Proc. 15th Intl. Conf. Software Reuse, Limassol, Cyprus, June 2016.
41. Nariman Mirzaei, Joshua Garcia, Hamid Bagheri, Alireza Sadeghi, and Sam Malek. "Reducing Combinatorics in GUI Testing of Android Applications," Proc. 38th Intl. Conf. Software Engineering (ICSE 2016), Austin, TX, May 2016.
42. Eric Yua and Sam Malek, "Mining Software Component Interactions to Detect Security Threats at the Architectural Level." Proc. 13th Working IEEE/IFIP Conf. Software Architecture (WICSA 2016), Venice, Italy, April 2016.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
14-04-2019	Final Performance	15 Oct 2015 to 14 Oct 2018

4. TITLE AND SUBTITLE
RASS: Resilient Autonomic Software Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-16-1-0030

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Daniel Menasce, Hassan Gomaa, Sam Malek

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
GEORGE MASON UNIVERSITY
4400 UNIVERSITY DR
FAIRFAX, VA 22030-4422 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2019-0090

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objective of this research is to conceptualize, implement, and evaluate a framework called Resilient Autonomic Software Systems (RASS) inspired by the MAPE-K (i.e., Monitor, Analyze, Plan, and Execute based on Knowledge) paradigm for autonomic systems (aka self-managing systems). Sensor data is obtained from a variety of systems (e.g., airplanes, UAVs, cloud systems, and cell-phones) by the Monitor component of RASS. These data are aggregated and sent to the Analyze Adaptation component that determines whether self-adaptation is needed and why. The results of this

15. SUBJECT TERMS
System Software, Software Systems, Mission Support Software, Dynamic Resource Allocation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			NGUYEN, TRISTAN
Unclassified	Unclassified	Unclassified	UU		19b. TELEPHONE NUMBER *(Include area code)* 703-696-7796

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

AFOSR Deliverables Submission Survey

Response ID:10683 Data

1.

Report Type

Final Report

Primary Contact Email

Contact email if there is a problem with the report.

menasce@gmu.edu

Primary Contact Phone Number

Contact phone number if there is a problem with the report

703-993-1537

Organization / Institution name

George Mason University

Grant/Contract Title

The full title of the funded effort.

RASS: Resilient Autonomic Software Systems

Grant/Contract Number

AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-16-1-0030

Principal Investigator Name

The full name of the principal investigator on the grant or contract.

Daniel A. Menasce

Program Officer

The AFOSR Program Officer currently assigned to the award

Tristan Nguyen

Reporting Period Start Date

10/15/2015

Reporting Period End Date

10/14/2018

Abstract

The objective of this research is to conceptualize, implement, and evaluate a framework called Resilient Autonomic Software Systems (RASS) inspired by the MAPE-K (i.e., Monitor, Analyze, Plan, and Execute based on Knowledge) paradigm for autonomic systems (aka self-managing systems). Sensor data is obtained from a variety of systems (e.g., airplanes, UAVs, cloud systems, and cell-phones) by the Monitor component of RASS. These data are aggregated and sent to the Analyze Adaptation component that determines whether self-adaptation is needed and why. The results of this analysis are passed to the Plan Adaptation component, which determines an optimal or near-optimal plan to automatically adapt the system. This plan is received by the Execution Adaptation component, which sends adaptation commands to the parts of the system that need to be adapted and/or recovered, and orchestrates the adaptation to ensure an appropriate and consistent recovery.

Because the adaptation plan is driven by the system's software architecture, an architecture discovery system called DeSARM recovers the descriptive architecture (i.e., the one that reflects the actual system) or supplements the prescriptive architecture (i.e., the one used to guide the development) when one is not available or has been eroded by system evolution. The Knowledge Base of RASS includes recovery methods and models that capture their performance overhead, failure recovery software patterns, adaptation patterns, a configuration database, an application binary repository to assist in recovering the software architecture of mobile applications, and the descriptive and prescriptive software architectures. A significant challenge of our research is that all the activities described above need to be carried out in a completely decentralized fashion. This is essential for achieving high-resiliency and high-dependability. The RASS group published 42 papers covering the areas of automatic generation of architectural models of mobile applications, robust decentralized discovery of software architectures of distributed systems, analysis and optimization of resiliency mechanisms, decentralized recovery patterns and connectors, automated evaluation of recovery and adaptation frameworks, autonomic fog and cloud resource management and elasticity control, analytic modeling of distributed systems, and moving target defenses. Additionally, we graduated 3 PhD students plus another expected to graduate in May 2019.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

SF_298-Final.pdf

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

AFOSR-Just_Final_Reportpdf.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

1. W. Connell, D.A. Menasce, M. Albanese, Performance Modeling of Moving Target Defenses with Reconfiguration Limits, IEEE Tr. Dependable and Secure Computing, accepted for publication on November 15, 2018.
2. D. Menasce and S. Bardhan, TDQN: Trace-Driven Analytic Queuing Network Modeling of Computer Systems, The Journal of Systems & Software, Elsevier, Vol. 147, January 2019, pp. 162-171.
3. U. Tadakamalla and D.A. Menasce, FogQN: An Analytic Model for Fog/Cloud Computing, Proc. 1st Workshop on Managed Fog-to-Cloud (mF2C), joint with 11th IEEE/ACM Intl. Conf. Utility and Cloud Computing, Zurich, Switzerland, December 17-20, 2018.
4. V. Tadakamalla and D.A. Menasce, Model-Driven Elasticity Control for Multi-Server Queues Under Traffic Surges in Cloud Environments, Proc. 2018 Intl. Conf. Autonomic Computing, Trento, Italy, September 3-7, 2018.
5. M. Hammad, J. Garcia, and S. Malek, SALMA: Self-Protection of Android Systems from Inter-Component Communication Attacks. 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2018), Montpellier, France, September 2018.
6. V. Tzeremes and H. Gomaa, A Software Product Line Approach to Designing End User Applications for the Internet of Things, Proc. 13th Intl. Joint Conf. Software Technologies (ICSOFT 2018), Porto, Portugal, July 2018.
7. M. Shin, H. Gomaa, and D. Pathirage, A Software Product Line Approach for Feature Modeling and Design of Secure Connectors, Proc. 13th Intl. Joint Conf. Software Technologies (ICSOFT 2018), Porto, Portugal, July 2018 (best paper award, invited to submit an extended version to a Spring LNCS publication).
8. V. Tadakamalla and D.A. Menasce, Analytic Model of Traffic Surges for Multi-Server Queues in Cloud Environments, Proc. Workshop: Cloud Performance and Reliability, IEEE CLOUD 2018 Conf. July 2-7, 2018, San Francisco, CA, USA.

9. N. Bajunaid and D.A. Menasce, Efficient Modeling and Optimizing of Checkpointing in Concurrent Component-Based Software Systems, J. Systems and Software, Vol. 139, May 2018, pp. 1-13.
10. Mahmoud Hammad, Joshua Garcia, and Sam Malek. A Large-Scale Empirical Study on the Effects of Code Obfuscations on Android Apps and Anti-Malware Products. Proc. Intl. Conf. Software Engineering (ICSE), May 2018, Gothenburg, Sweden.
11. J. Porter, D.A. Menasce, H. Gomaa, and E. Albassam, TESS: Automated Performance Evaluation of Self-Healing and Self-Adaptive Distributed Software Systems, 9th ACM/SPEC Intl. Conf. Performance Engineering (ICPE), Berlin, Germany, April 9--13, 2018.
12. D.A. Menasce and Noor Bajunaid, Performance Evaluation of Heterogeneous Multi-Queues with Job Replication 2017 Computer Measurement Group Conf. (imPACt 2017), New Orleans, LA, November 6-9, 2017.
13. Joshua Garcia, Mahmoud Hammad, and Sam Malek. Lightweight, Obfuscation-Resilient Detection and Family Identification of Android Malware ACM Transactions on Software Engineering and Methodology (TOSEM), November 2017 (Accepted)
14. W. Connell, D.A. Menasce, M. Albanese, Performance Modeling of Moving Target Defenses, 2017 Moving Target Defense (MTD) Workshop, Oct 30th - Nov 3rd, 2017, Dallas, Texas.
15. M. Carvalho, D.A. Menasce, and F. Brasileiro, Capacity Planning for IaaS Cloud Providers Offering Multiple Service Classes, Future Generation Computer Systems, Elsevier, 77 (2017), pp. 97–111.
16. M. Shin, H. Gomaa, and D. Pathirage, Reusable Secure Connectors for Secure Software Architectures, Proc. 4th International Workshop on Model-Driven and Component-Based Software Engineering (ModComp), MODELS Conference, Austin, Texas, September 2017.
17. H. Gomaa, E. Albassam, and D.A. Menasce, Run-time Software Architectural Models for Adaptation, Recovery and Evolution, Proc. 12th International Workshop on Models@runtime, MODELS Conference, Austin, Texas, September 2017.
18. E. Albassam, H. Gomaa, and D.A. Menasce, Model-Based Recovery and Adaptation Connectors: Design and Experimentation, ICSOFT 2016, CCIS 743, E. Cabello et al. (Eds.), Lecture Notes in Computer Science, Chapter 6, Springer Intl. Publishing AG, pp. 1–24, 2017, DOI: 10.1007/978-3-319-62569-0_6.
19. Joshua Garcia, Mahmoud Hammad, Negar Ghorbani, and Sam Malek. Automatic Generation of Inter-Component Communication Exploits for Android Applications. 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, September 2017.
20. Venkat Tadakamala and D.A. Menasce, Analysis and Autonomic Elasticity Control for Multi-Server Queues Under Traffic Surges, 2017 IEEE Intl. Conf. Cloud and Autonomic Computing (ICCAC), Tucson, AZ, USA, September 18-22, 2017.
21. Alireza Sadeghi, Reyhaneh Jabbarvand, and Sam Malek. PATDroid: Permission-Aware GUI Testing of Android. 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, September 2017.
22. Reyhaneh Jabbarvand and Sam Malek. muDroid: An Energy-Aware Mutation Testing Framework for Android. 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, September 2017.
23. E. Albassam, H. Gomaa, and D.A. Menasce, Variable Recovery and Adaptation Connectors for Dynamic Software Product Lines, 10th Intl. Workshop on Dynamic Software Product Lines (DSPL 2017), Sevilla, Spain, September 25-29, 2017.
24. E. Albassam, J. Porter, H. Gomaa, and D.A. Menasce, DARE: A Distributed Adaptation and Failure Recovery Framework for Software Systems, 14th IEEE Intl. Conf. Autonomic Computing (ICAC 2017), Columbus, OH, July 17-21, 2017.
25. Alireza Sadeghi, Hamid Bagheri, Joshua Garcia, and Sam Malek. A Taxonomy and Qualitative Comparison of Program Analysis Techniques for Security Assessment of Android Software. IEEE Transactions on Software Engineering (TSE), Vol. 43, No. 6, June 2017.
26. Alireza Sadeghi, Naeem Esfahani, and Sam Malek. Ensuring the Consistency of Adaptation through Inter- and Intra-Component Dependency Analysis. ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 26, No. 1, May 2017.
27. Mahmoud Awad and Daniel A. Menasce, Deriving Parameters for Open and Closed QN Models of Operational Systems Through Black Box Optimization, 8th ACM/SPEC Intl. Conf. Performance Engineering, L'Aquila, Italy, April 22--26, 2017.
28. Mahmoud Hammad, Hamid Bagheri, and Sam Malek. DELDroid: Determination and Enforcement of Least-Privilege Architecture in Android. Intl. Conf. Software Architecture (ICSA), Gothenburg, Sweden, April 2017.
29. Noor Bajunaid and Daniel A. Menasce, Analytic Models of Checkpointing for Concurrent Component-Based Software Systems, 8th ACM/SPEC Intl. Conf. Performance Engineering, L'Aquila, Italy, April 22--26, 2017.
30. Hamid Bagheri, Joshua Garcia, Alireza Sadeghi, Sam Malek, Nenad Medvidovic, "Software Architectural Principles in

Contemporary Mobile Software: from Conception to Practice," J. Systems and Software 119, 2016, pp. 31-44.
31. Daniel A. Menasce and Noor Bajunaid, "Modeling and Optimization of Multitiered Server-based Systems," 2016 Computer Measurement Group Conference (imPACT 2016), La Jolla, CA, November 7-10, 2016.
32. Mahmoud Awad and Daniel A. Menasce, "A Knowledge Base to Support the Automatic Derivation of Performance Models of Operational Systems," 2016 Computer Measurement Group Conference (imPACT 2016), La Jolla, CA, November 7-10, 2016.
33. Jason Porter, Daniel A. Menasce, and Hassan Gomaa, "DeSARM: A Decentralized Software Architecture Discovery Mechanism for Distributed Systems." 11th International Workshop on Models@run.time (MODELS 2016), Saint-Malo, France, October 4, 2016.
34. Mahmoud Awad and Daniel A. Menasce, "Performance Model Derivation of Operational Systems Through Log Analysis," IEEE Intl. Symp. Modeling, Analysis and Simulation of Computer Systems and Telecommunication Systems (MASCOTS 2016), Imperial College, London, UK, September 19-21, 2016.
35. Bradley Schmerl, Jeff Gennari, Alireza Sadeghi, Hamid Bagheri, Sam Malek, Javier Camara, and David Garlan, "Architecture Modeling and Analysis of Security in Android Systems," Proc. 10th European Conf. Software Architecture, Istanbul, Turkey, September 2016.
36. Sam Malek, Kyle Canavera, and Naeem Esfahani, "Automated Inference Techniques to Assist with the Construction of Self-Adaptive Software," in Managing Trade-offs in Adaptable Software Architectures, eds. Ivan Mistrik, Nour Ali, John Grundy, Rick Kazman, and Bradley Schmerl, Elsevier.
37. Emad Albassam, Hassan Gomaa, and Daniel A. Menasce, "Model-based Recovery Connectors for Self-Adaptation and Self-Healing," Proc. 11th Intl. Joint Conf. Software Technologies (ICSOFT 2016), July 24-26, 2016, Lisbon, Portugal.
38. Vasilios Tzeremes and Hassan Gomaa, "A Multi-Platform End User Software Product Line Meta-model for Smart Environments," Proc. 11th Intl. Joint Conf. Software Technologies (ICSOFT 2016), July 24-26, 2016, Lisbon, Portugal.
39. Hamid Bagheri, Alireza Sadeghi, Reyhaneh Jabbarvand Behrouz, and Sam Malek. "Practical, Formal Synthesis and Autonomic Enforcement of Security Policies for Android." Proc. 46th Annual IEEE/IFIP Intl. Conf. Dependable Systems and Networks (DSN 2016), Toulouse, France, June 2016.
40. Michael Shin, Hassan Gomaa, and Don Pathirage, "Reusable Secure Connectors for Secure Software Architectures," Proc. 15th Intl. Conf. Software Reuse, Limassol, Cyprus, June 2016.
41. Nariman Mirzaei, Joshua Garcia, Hamid Bagheri, Alireza Sadeghi, and Sam Malek. "Reducing Combinatorics in GUI Testing of Android Applications," Proc. 38th Intl. Conf. Software Engineering (ICSE 2016), Austin, TX, May 2016.
42. Eric Yua and Sam Malek, "Mining Software Component Interactions to Detect Security Threats at the Architectural Level." Proc. 13th Working IEEE/IFIP Conf. Software Architecture (WICSA 2016), Venice, Italy, April 2016.

New discoveries, inventions, or patent disclosures:

Do you have any discoveries, inventions, or patent disclosures to report for this period?

No

Please describe and include any notable dates

Do you plan to pursue a claim for personal or organizational intellectual property?

Changes in research objectives (if any):

None

Change in AFOSR Program Officer, if any:

We started with Dr. Kathleen Kaplan as PO. Then Dr. James Lawton became the PO. Finally, Dr. Tristan Nguyen became the PO.

Extensions granted or milestones slipped, if any:

none

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Nov 17, 2018 18:03:17 Success: Email Sent to: menasce@gmu.edu