

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

AVAGO TECHNOLOGIES INTERNATIONAL SALES PTE. LIMITED,
Patent Owner.

Inter Partes Review No. IPR2020-01582

U.S. Patent No. 8,572,138 B2

DECLARATION OF DAVID ROSENBLUM

TABLE OF CONTENTS

I.	BACKGROUND AND QUALIFICATIONS	1
II.	UNDERSTANDING OF THE LAW	3
	A. Anticipation	3
	B. Obviousness.....	4
	C. Claim Construction	6
III.	PERSON OF ORDINARY SKILL IN THE ART	7
IV.	OVERVIEW OF THE '138 PATENT	8
V.	THE '138 PATENT DISCLOSES VIRTUAL MACHINES ON PHYSICAL NODES.....	12
VI.	THE '138 PATENT DISCLOSES A TWO-PHASE IMAGE CAPTURE PROCESS.....	14
VII.	THE THREE-STEP DEPLOYMENT METHODOLOGY OF THE '138 PATENT	22
VIII.	THE CHALLENGED CLAIMS REQUIRE THE SAME THREE-STEP DEPLOYMENT PROCESS DISCLOSED IN THE SPECIFICATION OF THE '138 PATENT	31
	A. Claim 1	32
	B. Claim 19	40
	C. Claim 26	43
IX.	CLAIM CONSTRUCTION	45
X.	THE ASSERTED PRIOR ART	49
	A. Khandekar	49
	1. Background on Khandekar	49
	2. Application of Khandekar to the Challenged Claims	54
	a) The Three-Step Deployment Process	54
	b) Storage of Image Instances.....	58

c) Autonomic Deployment of the Image Instances
 of Software Applications69

d) Advantages of the Claimed Invention over
 Khandekar70

B. Le72

C. Sidebottom and Stone73

XI. CONCLUSION73

I. BACKGROUND AND QUALIFICATIONS

1. I am qualified by education and experience to testify as an expert in the field of virtual machines and distributed computing systems. Attached as Exhibit 1 is a copy of my curriculum vitae detailing my experience and education. Additionally, I provide the following overview of my background related to my expert testimony.

2. I am a Professor and Department Chair at the Department of Computer Science, at George Mason University, where I also hold an Endowed Chair from the Planning Research Corporation. I have studied and practiced in the field of computer science, generally, for close to 40 years, and have been a professor of computer science since 1996.

3. I received my Ph.D. in Electrical Engineering (Computer Systems) in 1988 from Stanford University, where I also received my Masters of Science (M.S.) degree in Electrical Engineering (Computer Systems) in 1987. Before that, I received another M.S. degree in Computer Science in 1983 from North Texas State University, where I also received my Bachelor of Science (B.S.) degree (*summa cum laude*) in Computer Science in 1982.

4. After receiving my Ph.D., I was a Member of the Technical Staff at AT&T Bell Laboratories from 1988 to 1996, after which I was on the Faculty of the Department of Information and Computer Science at the University of California,

Irvine, until 2002. At that time, I was also the Chief Technology Officer and Principal Architect for PreCache, a startup company developing technology in the area of publish/subscribe networking.

5. I then accepted a position as a Professor of Software Systems in the Department of Computer Science at University College London, which I held from 2004 to 2011. I then accepted a position as a Professor of Computer Science at the National University of Singapore, where I also served as Dean of the NUS School of Computing from 2013 to 2016 and the Director of the NUS-Singtel Cyber Security Research and Development Laboratory from 2016 to 2020. Thereafter, I began my current position at George Mason University.

6. I have been an avid researcher throughout my academic career, and my research has addressed a wide range of problems spanning the breadth of the software development life cycle, including software specification, architecture, design, verification, testing, analysis and maintenance, with particular emphasis on software for distributed systems. My current research focuses on the use of machine learning in the design and testing of mobile, context-aware adaptive systems for ubiquitous computing and the Internet of Things. In 1997, I received a CAREER Award from the US National Science Foundation for my work on distributed component-based software, and I held a Wolfson Research Merit Award from the Royal Society from 2004-2009.

7. I am a Fellow of the ACM and IEEE and have received two test-of-time awards for my research papers. I was previously Editor-in-Chief of the ACM Transactions on Software Engineering and Methodology (ACM TOSEM) and an Associate Editor for the IEEE Transactions on Software Engineering (IEEE TSE). In addition, I have served as Member-at-Large, Vice Chair, Chair and Past Chair of the ACM Special Interest Group on Software Engineering (ACM SIGSOFT), and in 2018, I received the ACM SIGSOFT Distinguished Service Award.

8. Moreover, I have authored and co-authored over 200 technical writings in edited books, international journals, and conference proceedings, focusing on the fields of computer science generally, and I am the named inventor of 11 U.S. and international patents in related fields.

II. UNDERSTANDING OF THE LAW

A. Anticipation

9. I understand that to be valid, a patent claim must be non-obvious and novel. I also understand that a patent claim is not novel if it is anticipated by a single prior art reference – that is, a single prior art reference discloses each and every element of the claim either expressly or inherently. I also understand that a single reference cannot merely disclose each element of a claim to be found to anticipate. Rather, it must disclose all of the elements as arranged in the claim.

10. I further understand that for a reference to "inherently" disclose something, the missing descriptive matter must necessarily be present in the reference, not merely probably or possibly present, and that it would be so recognized by a person of ordinary skill.

11. I understand that if an element of the claim is not disclosed by a prior art reference, then the claim is not anticipated by that reference.

12. I have been informed that a prior art reference that anticipates must enable one of ordinary skill in the art to make the claimed invention. The enablement cannot require undue experimentation.

B. Obviousness

13. I understand that to find a patent invalid for obviousness, Petitioner must prove that the differences between the claimed invention and the prior art taken as a whole would have been obvious to one of ordinary skill in the art at the time the invention was made.

14. I have been informed that to render a claim obvious, a combination of prior art references must disclose each and every claim element of that claim, and that obviousness is a question of law (*i.e.*, for the ALJ to determine) based on the underlying facts. I understand that the underlying factual inquiries are: (1) the scope and content of the prior art, (2) the differences between the prior art and the claims

at issue, (3) the level of ordinary skill in the pertinent art, and (4) secondary considerations of nonobviousness.

15. I am also informed that the fact that all of the elements of a claimed invention can be found separately in more than one prior art reference is not sufficient to warrant a finding that a claimed invention is obvious. There must be a reason why it would have been obvious to modify or combine the prior art to arrive at the claimed invention. I understand that a helpful insight into this determination is whether there is a teaching, suggestion or motivation in the prior art that would lead one of ordinary skill in the art to combine the elements. When there is no suggestion for the proposed combination, or when the prior art suggests something other than the combination, a finding of obviousness is generally not warranted.

16. I understand that the prior art must be considered in its entirety, including disclosures that teach away from the claimed invention. Furthermore, if a proposed modification to a prior art reference would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification.

17. I am also informed that to render a claim obvious there must be a finding that a skilled artisan would have had a reasonable expectation of success in combining the teachings of the prior art references to achieve the claimed invention.

18. I understand that the pertinent obviousness inquiry takes place at the time of the invention. Therefore, the use of hindsight in an obviousness analysis is impermissible.

19. I further understand that, in deciding as to whether the claimed invention would have been obvious to a person of ordinary skill, the fact finder must consider certain objective factors, such as commercial success, long-felt, but unsolved need, unexpected results, copying, and praise by others in the field. The presence of such factors – known as "secondary considerations" – is evidence of non-obviousness. I also understand that a connection, or nexus, must exist between the secondary consideration and the claimed invention.

C. Claim Construction

20. I understand that in *inter partes* review, claims are given their "ordinary and customary meaning" which is the meaning that the term would have, in view of the specification, to a person of ordinary skill in the art in question at the time of the invention. I have applied this standard in my analysis of the Challenged Claims, except where I have expressly stated otherwise.

21. I have been told that the "construction cannot be divorced from the specification and the record evidence" and that the construction must be reasonable in light of the claims and specification.

22. I have been told that the principle of antecedent basis means that, absent indications to the contrary, subsequent uses of a claim term should be understood to refer to the first use of that claim term. For example, if a patent claim recites "an item" and then later recites "the item" or "said item," those subsequent uses generally refer to the same item initially recited.

23. I understand that a patent claim requires an ordering of steps when the claim language, as a matter of logic or grammar, requires that the steps be performed in the order written. I understand that a claim also requires steps to be performed in the order written if the specification directly or implicitly requires an order of steps.

III. PERSON OF ORDINARY SKILL IN THE ART

24. In my opinion, a person of ordinary skill in the art ("POSITA") at the time of the filing of U.S. Patent No. 8,572,138 (the "'138 Patent") would have had a bachelor's degree in Electrical Engineering, Computer Science, or a similar discipline, with one to two years of experience in this or a related field. The POSITA would also have been familiar with computing environments and distributed computing systems.

25. I have also reviewed Dr. Shenoy's definition of a POSITA, which is similar to my own, with the exception that Dr. Shenoy lists a variety of tasks that a POSITA would have been able to perform. *See* Ex. 1003 ¶ 35. I do not necessarily agree that a POSITA would have been required to perform those tasks, but my

opinions set forth in this declaration do not change regardless of which definition is applied.

IV. OVERVIEW OF THE '138 PATENT

26. I understand that the claims challenged in the Petition in this *inter partes* review are claims 1-5, 9, 10, 17, 19-21, and 26 (the "Challenged Claims"). In order to better understand the inventions set forth in the Challenged Claims, the following is an overview of some of the basic components of these various inventions as explained by the '138 Patent.

27. **Virtual Machine:** The inventions of the '138 Patent generally relate to deployment of virtual machines on nodes in a distributed computing system. *See* Ex. 1001 ('138 Patent) at Abs. As used in the '138 Patent, the term virtual machine refers to a virtual environment that emulates a physical computer platform. *See id.* at 32:67-33:2. A distributed computing system refers to a collection of independent networked computers (*i.e.*, nodes) that are configured to function (or to appear to users) as a single coherent computing system. *See id.* at 1:18-25. Each networked computer or node may be a physical machine with physical hardware and ports. *See id.* at 32:65-33:24. The nodes are the target locations for deploying virtual machines. *See id.*

28. The '138 Patent uses the term "virtual machine" to refer **only** to this virtual (emulated) computer platform that is provided (created) by the virtual

machine manager software running on an application node. (As I note below, this usage is very different from the way the term is used in the prior art references cited by Petitioner). This emulated computer platform is merely emulated hardware, without any operating system or applications installed. For example, FIG. 24 of the '138 Patent (reproduced below) illustrates virtual machines 404 separate from the operating system and applications 408. The '138 Patent further explains,

For example, virtual machines [404] provide environments on which guest operating systems (OS) execute as through the operating systems were operating directly on a physical computing platform. In this manner, multiple operating systems and software applications [408A] through [408N] (collectively, operating systems and applications [406]) may execute on virtual machines 404A through 404N, respectively

Id. at 33:2-9.

29. Thus, as used in the '138 Patent, the term "virtual machine" refers to an emulated hardware platform without any software installed; this allows any desired operating system and/or applications to be deployed onto the virtual machine. *See id.* A term used in the art for this type of platform is an emulated "bare metal" machine. *See* Ex. 2004 at 1-2.

30. **Virtual Machine Manager:** A virtual machine manager is computer software, firmware, or hardware that is used to manage and, as specified by the

Challenged Claims, to provide (or create) one or more virtual machines. *See* Ex. 1001 at 32:59-67; *see also id.* at 35:6-9.

31. **Software Applications:** A virtual machine without any executable software applications on it is a non-functional shell or unconfigured machine. *See id.* at 32:65-33:24. To provide a working machine, software applications are deployed or installed on the virtual machines. *See id.* For example, to emulate a home computer, software applications including, but not limited to, Microsoft Windows 10 operating system, Microsoft Office Suite of applications (*e.g.*, WORD, EXCEL, OUTLOOK), and/or an Internet browser, may be deployed on the virtual machine. *See id.*

32. **Image Instances:** In some cases, virtual machines operate by instantiating a software image. *See id.* Generally, an image of a software application is a file that contains a snapshot or "image" of an installed and configured software application. *See id.* at 35:1-24. The file may also contain established or selected settings and/or preferences. *See id.* at 33:25-67. An instance of such an image is a copy of the image deployed onto a particular virtual machine. *See id.* at 32:65-33:24. In this manner, deployment of the image instance of the software application can avoid the time-consuming and tedious task of installing each software application on each virtual machine, and thereafter selecting the desired settings and/or preferences for each installed software application. *See id.* at 33:25-36:20.

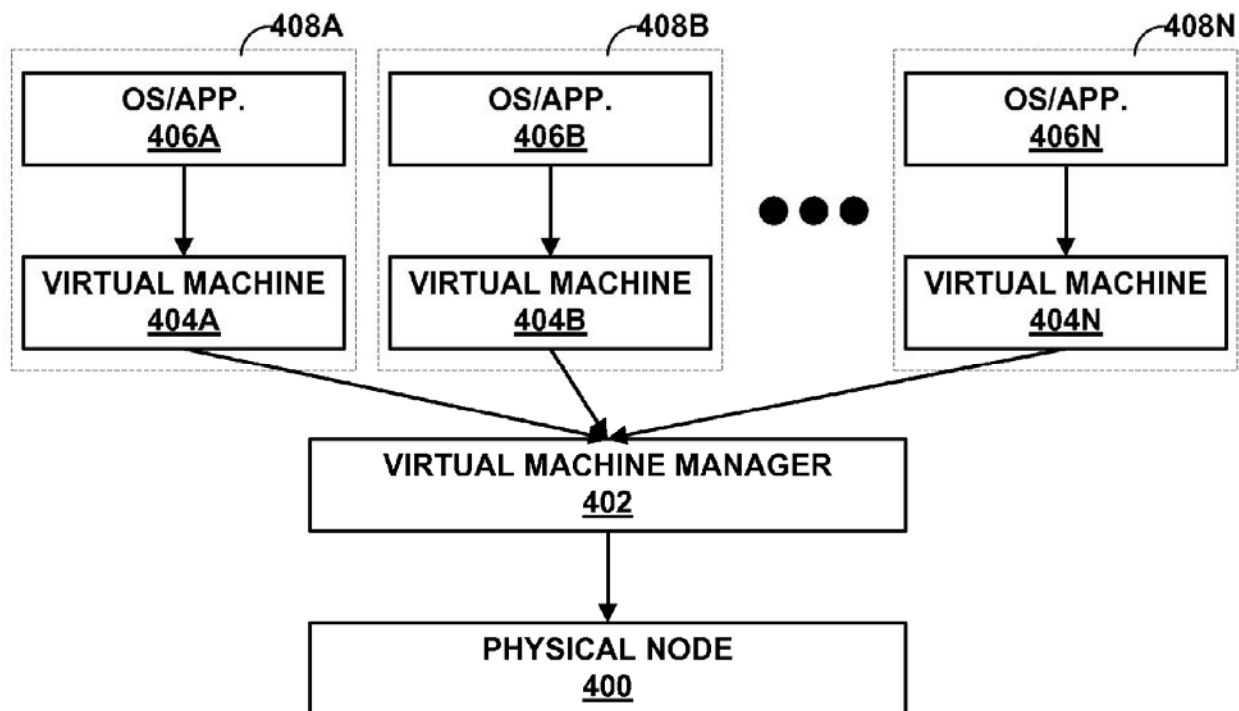
33. For example, a first user, who has already installed and selected optimal settings and/or preferences for the operating system and tailored software applications for performing at least one set of desired functions, can take a "snapshot" of the system to create an image of the operating system with the installed collection of software applications and then send an instance of that image to a second user. *See id.* at 33:25-67, *see also id.* at 35:1-24. Upon deploying (and in some cases, executing) the image instance of the collection of software applications, the second user would immediately be provided with a collection of software applications with the same functionalities, settings, and preferences as the software applications of the first user. *See id.* Accordingly, the second user does not need to install the collection of software applications or to manually set the settings and preferences to match those that were set by the first user. *See id.*

34. The virtual machine deployment technique disclosed by the '138 Patent employs these various components, and the Challenged Claims are directed to this technique as well. *See id.* at 36:26-49.

35. As I will discuss in detail below, these claimed inventions of the '138 Patent are fundamentally different than the techniques taught in the references cited by Petitioner, and the techniques of the '138 Patent provide significant advantages over those references, and all of the prior art of which I am aware.

V. THE '138 PATENT DISCLOSES VIRTUAL MACHINES ON PHYSICAL NODES

36. FIG. 24 of the '138 Patent illustrates the arrangement of these various components after the deployment of image instances of virtual machine managers and software applications:



Ex. 1001, FIG. 24.

37. As illustrated by FIG. 24, an instance image of a virtual machine manager has been deployed to and "is executing on node 400." *Id.* at 32:59-60. When deployed (and executing) on node 400, virtual machine manager 402 creates or provides one or more virtual machines 404. *See id.* at 32:65-67; *accord id.* at 35:33-36 ("After these physical nodes boot with the virtual machine manager image instance, each of the nodes creates a set of virtual machines as specified by the image

instances."). As noted above, the '138 Patent uses the term "virtual machine" to refer to software that creates an environment that emulates a computer platform," or more specifically, "a physical computing platform." *Id.* at 33:1-6.

38. The '138 Patent further discloses, "multiple operating systems and software applications 406A through 406N (collectively, operating systems and applications 406) may execute on virtual machines 404A through 404N, respectively, and the virtual machines 404 may execute on a single physical node 400 or multiple physical nodes." *Id.* at 33:6-11.

39. To summarize these teachings from FIG. 24, when a virtual machine manager is deployed to a physical node (*e.g.*, an application node) and booted (or executed) on that node, it creates or provides one or more virtual machines, which emulate physical computing platforms. Different operating systems and/or applications can execute on these emulated physical computing platforms (for example, after being deployed as image instances), which I will describe in detail below.

40. To enable the arrangement illustrated by FIG 24, the '138 Patent teaches a specific image capture and deployment methodology, which I will first discuss in detail. I will then explain how the Challenged Claims are limited to this specific methodology. While I understand it is not appropriate to import limitations from the description of the '138 Patent into the claims of the patent, I do believe that the claims

should be interpreted in light of the specification. I also believe that the specification of a patent, as a part of the patent's intrinsic record, can aid a POSITA in understanding the inventor's view of the patent's claims.

VI. THE '138 PATENT DISCLOSES A TWO-PHASE IMAGE CAPTURE PROCESS

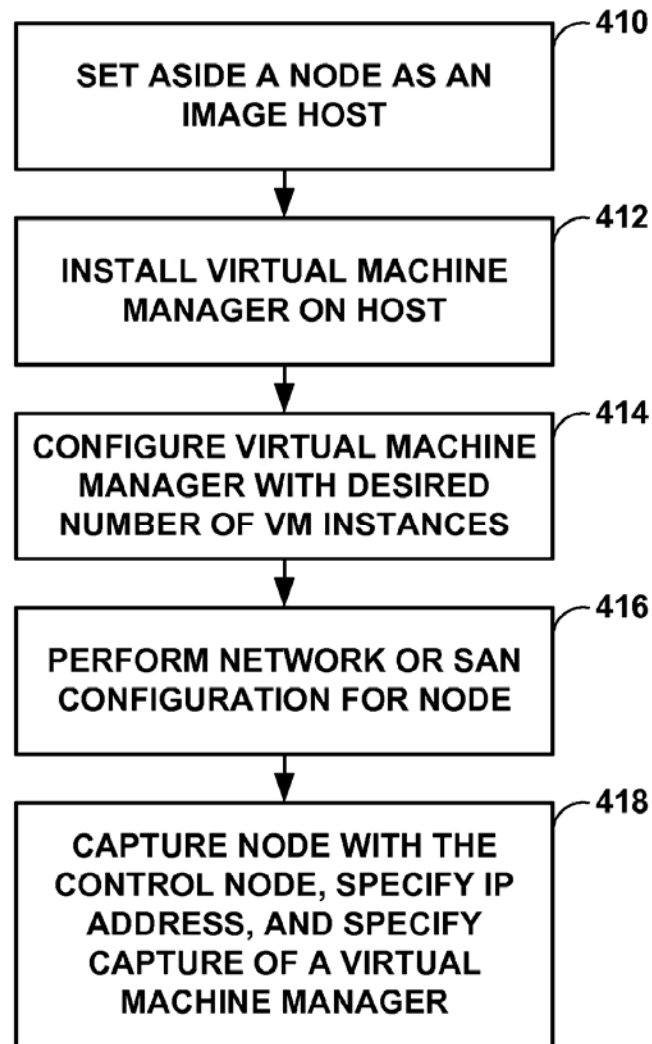
41. Although the Challenged Claims do not directly recite any limitations related to the capture of images of virtual machine managers and software applications, I believe that a review of the image capture process disclosed by the '138 Patent is helpful in understanding the inventions of the Challenged Claims. The '138 Patent refers to this image capture process as a "two-phase capture process," which is used to "provide autonomic control and deployment" of the virtual machine manager image instances and software application image instances. Ex. 1001 at 33:42-45.

42. The '138 Patent teaches that, according to the two-phase capture process, the system first captures an image of a virtual machine manager alone: "control node 12 may first capture an image of virtual machine manager 402 installed on physical node 400 **prior to installation and configuration of any guest operating system and applications 406.**" *Id.* at 33:46-49 (emphasis added). The '138 Patent further discloses, "[c]ontrol node 12 may utilize this captured image of virtual machine manager 402 as a 'golden image' from which instance images can be created for this type virtual machine." *Id.* at 33:49-52.

43. A POSITA would understand that a "golden image" is analogous to another common term in the art, a "gold master," which refers to a version of software that has been certified for release to manufacture, so that duplicates of the gold master can be mass produced (*e.g.*, at the time of invention, on CDs or DVDs). *See* Ex. 2005 at 1; *see also* Ex. 2006 at 69. Based on this analogy, a POSITA would understand the term "golden image" of a virtual machine manager to mean a master copy of an image of a virtual machine manager that could be duplicated to produce image instances of a virtual machine manager.

44. The '138 Patent further discloses, "[t]he instance images may be stored within application matrix 350 for autonomic deployment as other software images (FIG. 21)." Ex. 1001 at 33:52-54. It is important to note that these image instances are built from the gold master; as such they include a virtual machine manager prior to installation of an operating system or any other applications. *See id.* at 33:45-54. This disclosure, therefore, teaches that image instances of virtual machine managers are stored separately from any operating system or other software applications.

45. The '138 Patent teaches that FIG. 25 provides a more detailed description of this process: "FIG. 25 is a flowchart illustrating an exemplary procedure for creating and capturing a golden image for a particular virtual machine manager 402 (FIG. 24)." *Id.* at 34:48-50.



Id., FIG. 25.

46. In describing FIG. 25, the '138 Patent explains, "administrator 20 sets aside a node (e.g., node 400) as an image host (410). Administrator 20 may then install a particular type of virtual machine manager 402 on node 400 (412)." *Id.* at 34:50-53. The '138 Patent further explains,

"[a]fter administrator 20 performs all of the necessary configuration, administrator 20 instructs control node 12

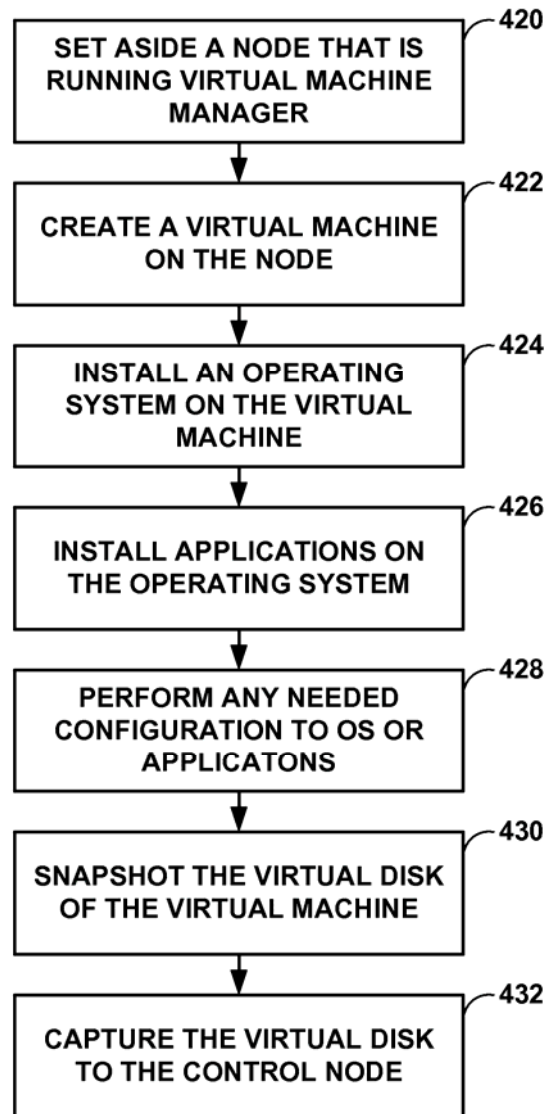
to capture the image of virtual machine manager 402 with the configurations (418). Control node 12 may store the captured image as a golden image in application matrix 350 for use in subsequent autonomic deployment of image instances."

Id. at 34:61-67.

47. Importantly, the '138 Patent teaches that this first phase in the two-phase capture process captures an image of the virtual machine manager before any operating system or applications have been deployed (or otherwise installed) to the virtual machine(s) provided by the virtual machine manager: "control node 12 may first capture an image of virtual machine manager 402 installed on physical node 400 **prior to installation and configuration of any guest operating system and applications 406.**" *Id.* at 33:46-49 (emphasis added). This image, of the virtual machine manager alone, is stored in the application matrix. *Id.* at 34:61-67. This image can be used as a "golden image," which I described above. *See id.* at 33:46-54.

48. I note that Dr. Shenoy agrees with my understanding of this first phase of the two-phase capture process, in which "the control node is capturing an image of the virtual machine manager where there is no OS or applications installed." Ex. 2002 at 42:7-9.

49. The '138 Patent teaches that images of software applications (which can include operating systems and other applications) are captured in the second phase of the two-phase capture process. *See* Ex. 1001 at 33:55-67. This process is described in connection with FIG. 26, which I have reproduced below:



Id., FIG. 26.

50. The '138 Patent describes FIG. 26 as "a flowchart illustrating an exemplary procedure for creating and capturing a virtual disk image file, i.e., a golden image of a guest operating system and applications for execution on a virtual machine." *Id.* at 35:1-4. According to this process, "administrator 20 sets aside a node (e.g., node 400) that is running a virtual machine manager (e.g., virtual machine manager 402) (420). Administrator 20 may then configure the virtual machine manager to create a virtual machine (e.g., virtual machine 404) on node 400 (422)." *Id.* at 35:4-9. I note that this process requires: (1) a virtual machine manager to be deployed to a node and (2) the virtual machine manager to be executing, in order to "create a virtual machine" on the node. *Id.*

51. Once the virtual machine has been created, the '138 Patent teaches that "administrator 20 installs the particular operating system on virtual machine 404 (424). Administrator 20 may then install one or more applications on the operating system (426)." *Id.* at 35:9-12. Once again, I observe that the virtual machine must be created (or provided) by the virtual machine manager before any operating systems or applications can be installed. I note as well that Dr. Shenoy agrees with my observation in this regard. *See* Ex. 2002 at 56:15-57:15.

52. After the desired operating system and applications have been installed, "administrator 20 may configure the operating system and applications as needed (428). In some embodiments, virtual machine manager 402 stores definition data

for the particular virtual machine, operating system, and applications in a virtual disk image file." Ex. 1001 at 35:14-16. I find it noteworthy that the '138 Patent again refers to the "definition data for the . . . virtual machine" as separate from the "operating system" and "applications" when describing the contents of the virtual disk. This teaching reinforces my view that the '138 Patent views the virtual machine as only the software required to emulate a bare metal computer platform.

53. At the end of the two-phase capture process, "administrator 20 takes a 'snapshot' of the data in the virtual disk image file (430). Administrator 20 may then capture this 'snapshot' to control node 12 as a golden image (432)." *Id.* at 35:17-20. This virtual disk image file (which includes the operating system and applications) is separate from the underlying virtual machine manager. Thus, in my opinion the '138 Patent teaches that the two-phase capture process (1) first captures and stores an image of a virtual machine manager without any software applications and (2) then captures and stores an image of the operating system and/or other software applications, independent from the underlying virtual machine manager.

54. The two-phase capture process taught by the '138 Patent provides significant advantages over the prior art, including the references cited by Petitioner. Specifically, the technique of capturing and storing images of virtual machine managers separate from images of operating systems and other applications allows the choice of a virtual machine manager compatible with the underlying hardware

of the physical node and the independent choice of software applications (including operating systems) in accordance with the needs of the user. As the '138 Patent notes,

In this way, distributed computing system 10 may autonomically utilize application nodes, e.g., node 400, to flexibly support a wide variety of operating systems and applications, and provide virtualized execution environment for those operating systems and applications. For instance, node 400 may autonomically configured to support a Macintosh Operating System designed for a PowerPC processor, and a Microsoft Windows Operating System designed for an x86 processor based on the particular goals of the autonomic system. **Thus, control node 12 may not need to distinguish between hardware processor architectures when deploying operating systems and applications.**

Id. at 34:30-41 (emphasis added).

55. This advancement over the prior art can provide significant advantages. For example, in a situation in which there are 100 different hardware configurations of physical nodes and 100 different software configurations (*i.e.*, combinations of operating systems and installed applications), the two-phase capture and storage technique can support these needs with 200 different stored images. Specifically, the control node might store 100 different images of virtual machine managers (to

support the 100 different hardware configurations of physical nodes) and 100 different images of software applications (to support the 100 different combinations of operating systems and installed applications). In contrast, if the virtual machine manager images were not stored separately from the software application images, supporting this environment would require 10,000 different images: 100 different images with the desired operating system/application combination for each of the 100 different hardware configurations of the physical nodes.

56. In this regard, I agree with Dr. Shenoy that the two-phase image capture and storage process provides significant flexibility advantages in the deployment of images to different configurations of physical nodes. *See* Ex. 2002 at 51:14-52:5.

VII. THE THREE-STEP DEPLOYMENT METHODOLOGY OF THE '138 PATENT

57. As I will describe in detail in the following paragraphs, the '138 Patent discloses a three-step deployment methodology, in which a control server first deploys a virtual machine manager to an application node.¹ The virtual machine

¹ For purposes of explanation and without loss of generality, I describe this methodology in terms of a single virtual machine manager creating a single virtual machine that hosts a plurality of software applications, even though the Challenged

manager is deployed from a virtual machine manager image instance (step 1). Next, that virtual machine manager executes on the application node (step 2). The execution of the virtual machine manager creates (or provides) a virtual machine, which as I note above, is an emulated computer platform. Finally, the control node deploys one or more software applications onto this emulated computer platform (step 3). As I explain above, these software applications, which are deployed from software application image instances, can include operating systems but are not so limited. As I will discuss further below, the '138 Patent demonstrates that this process requires the deployment and execution of the virtual machine manager before any software applications can be deployed. In this regard, the deployment process taught by the '138 Patent is similar to the image capture process I describe above, in which the virtual machine manager must be executing to create (or provide) a virtual machine onto which software applications can be installed (in the capture process) or deployed (in the deployment process). For this reason, I refer to the methodology disclosed by the '138 Patent as a "three-step deployment" methodology.

Claims describe this methodology in terms of a plurality of virtual machine managers, each of which may create a plurality of virtual machines.

58. As I describe above, the '138 Patent teaches that the image instances of the software applications are stored separately from the image instances of the virtual machine manager. Similarly, the three-step deployment process of the '138 Patent deploys the virtual machine manager image instances to application nodes separately from deploying the software application image instances to virtual machines created by the deployed virtual machine manager image instances. *See* Ex. 1001 at 34:1-13. In particular, the '138 Patent teaches, "control node 12 may deploy the image instance of virtual machine manager 402 to one or more nodes in free pool 13. Once control node 12 deploys the image instance of virtual machine manager 402 to a node, the node appears to be one or more new, unallocated nodes." *Id.* at 34:2-7.

59. The '138 Patent further teaches, "**Subsequently**, control node 12 may deploy an image instance of virtual disk image files 408 to one of the new unallocated nodes." *Id.* at 34:11-13 (emphasis added). There are two notable aspects to this quotation. First, I would note that the '138 Patent teaches that virtual disk image files are one type of software application image instance, so this passage clearly refers to the deployment of an image instance of a software application. *See id.* at 33:25-41; *accord* Ex. 1003 ¶ 38.

60. The second notable aspect of this passage is the use of the word "subsequently." As I explain below, while the '138 Patent expressly discloses that the third step of the three-step deployment process (deployment of the software

application image instance) is subsequent to the first step (deployment of the virtual machine manager image instance), the nature of the three-step deployment process inherently requires that the software application image instance is deployed separately from, and subsequent to, the deployment of the virtual machine manager image instance.

61. In fact, the only deployment technique disclosed by the '138 Patent expressly employs the three-step deployment process. Specifically, the '138 Patent illustrates the process using FIG. 27A (which illustrates step 1 and step 2) and FIG. 27B (which illustrates step 3):

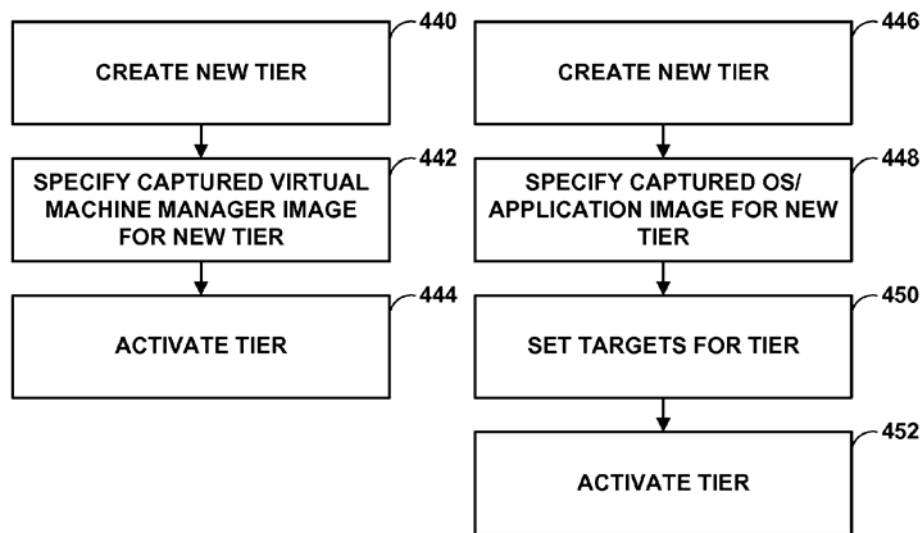


FIG. 27A

FIG. 27B

Ex. 1001, FIGS. 27A, 27B.

62. As the '138 Patent discloses, "FIG. 27A is a flowchart illustrating an exemplary procedure for autonomic deploying a **virtual machine manager** to a

node." *Id.* at 35:25-27 (emphasis added). The '138 Patent describes the deployment process of the virtual machine manager:

Initially, administrator 20 may instruct control node 12 to create a new tier (440). Subsequently, administrator 20 specifies one or more captured virtual machine manager image instances for assignment to the slots of the new tier (442). Administrator 20 may also specify whether the image instances of the tier run in persistent or non-persistent mode. Administrator 20 may then activate the new tier (444). After these physical nodes boot with the virtual machine manager image instance, each of the nodes creates a set of virtual machines as specified by the image instances. Control node 12 may then discover each of these virtual machines and place them in free pool 13 as if they were independent nodes. In this manner, each of the virtual machines appear to be physical nodes available for use as application nodes 14 when needed.

Id. at 35:27-40.

63. According to this disclosure, the virtual machine manager image instance is deployed to an application node (step 1 in the three-step deployment Process) and then executed, by the physical node booting with the virtual machine manager (step 2 in the three-step deployment process). *See id.* When executing, the virtual machine manager creates one or more virtual machines (as that term is used by the '138 Patent, *i.e.*, emulated computer platforms). *Id.* As Dr. Shenoy aptly

explains, "once the virtual machine manager has been installed on the physical node and a virtual machine has been created, the virtual machine appears to be a physical node for use as application nodes." Ex. 2002 at 68:12-69:2. Dr. Shenoy also correctly notes that it is not merely the deployment of the virtual machine manager that creates a virtual machine, but the execution of the virtual machine manager on the physical node: "[I]n the context of figure 27A that I was just describing, the virtual machines are created on the physical node on which the virtual machine manager **runs** in that example." *Id.* at 27:4-7 (emphasis added). Thus, I find that I have substantial agreement with the way in which Dr. Shenoy has characterized the disclosure of the '138 Patent regarding the deployment and execution of the virtual machine manager.

64. FIG. 27B of the '138 Patent (reproduced above) illustrates how the image instances of the software applications are deployed to these virtual machines (which are created, as noted above, only by the execution of the virtual machine manager on the host). In describing that figure, the '138 Patent discloses,

FIG. 27B is a flowchart illustrating an exemplary procedure for **deploying an operating system and application image instance to a node that is executing a virtual machine manager**. To start the procedure, administrator 20 may instruct control node 12 to create a new tier, including defining the number of slots for the tier, or select an existing tier (446). Administrator 20 may

then specify a captured operating system and application image for the tier (448). In addition, administrator 20 may set minimum, maximum, and target values for the image instances to be deployed to the slots new tier (450). Administrator 20 may then activate the tier (452). After administrator 20 activates the tier, based on the current state and goals of distributed computing system 10, **control node 12 autonomically selects a virtual machine from free pool 13, associates the virtual machine with a slot in the tier**, identifies the virtual disk image file (image instance) associated with the particular slot, copies the virtual disk image file to the local hard disk of the physical node on which the virtual node resides or the an associated SAN, and finally boots the virtual machine from the virtual disk image file.

Ex. 1001 at 35:48-67 (emphasis added).

65. I agree with Dr. Shenoy's acknowledgement that this passage "does talk about deploying the virtual machine manager node **and then** deploying image instances of virtual disk image files to an unallocated node." Ex. 2002 at 64:3-6 (emphasis added). Thus, I find that I have substantial agreement with the way in which Dr. Shenoy has characterized the deployment of the image instances of virtual disk image files (*i.e.*, image instances of software applications) as disclosed in the '138 Patent.

66. Thus, as set forth in multiple examples in the specification, the autonomic deployment process disclosed in the '138 Patent includes three separate steps involving: (1) copying an image instance of a virtual machine manager onto a node; (2) executing the virtual machine manager to create a virtual machine on the node; and (3) copying an image instance of a software application onto the virtual machine. *See* Ex. 1001 at 34:1-13, 35:25-47, 35:48-67.

67. Moreover, according to this disclosure (which is the only disclosure of deployment of images of software applications), the logic and grammar of the description in the '138 Patent both dictate that the individual steps in the three-step deployment process must be performed in the order described. Specifically, the control node must first deploy the image instance of the virtual machine manager to an application node (step 1). If the application node does not have a virtual machine manager, none of the rest of the deployment process is possible. This is because the virtual machine manager must execute on the application node to provide (or create) a virtual machine (step 2). As I note above, and Dr. Shenoy agrees, the '138 Patent discloses only one way in which a virtual machine can be provided or created: execution of a virtual machine manager. *See* Ex. 2002 at 33:12-21. Finally, only when the virtual machine has been provided can an image instance of a software application be deployed to that virtual machine (step 3). If the virtual machine had not already been provided by execution of the virtual machine manager on the

application node, there would be no virtual machine to which the image instance of the application could be deployed.

68. The language of the '138 Patent emphasizes this ordering:

Step 1 – To begin the deployment process, the specification uses the phrase "Once control node 12 deploys the image instance of virtual machine manager 402 to a node" (*i.e.*, this step must happen first). Ex. 1001 at 34:5-6.

Step 2 – The specification next states that the virtual machine manager supports the virtual machine(s) on the nodes (*i.e.*, the virtual machine manager provides the virtual machine on the node). *Id.* at 34:7-11. This of course cannot happen as a matter of logic until after the control node deploys the image instance of the virtual machine manager onto the node.

Step 3 – The specification states "**Subsequently**, control node 12 may deploy an image instance of virtual disk image files 408 to one of the new unallocated nodes" (*i.e.*, the image instances of the software applications are deployed onto the virtual machine after the deployment of the image instance of the virtual machine manager). *Id.* at 34:11-13 (emphasis added); *see also id.* at 32:67-33:2.

The use of the term "subsequently" clearly indicates that the execution of the virtual machine manager to create a virtual machine must occur before the deployment of an image instance of a software application.

69. I note that Dr. Shenoy characterizes the specification of the '138 Patent as describing the three-step deployment process. For example, Dr. Shenoy recognizes that the image instance of the virtual machine manager deploys before the image instance of the software applications. *See* Ex. 2002 at 63:10-64:6; 68:12-69:2; and 73:2-74:3. Dr. Shenoy also recognizes that, once the image instance of the virtual machine manager is deployed on the node, it creates the virtual machine. *See id.* at 68:12-69:2, 73:2-74:3. Dr. Shenoy also acknowledges that the image instance of the software application (*e.g.*, virtual disk image) is deployed after the deployment of the virtual machine manager and after the creation of the virtual machine (by execution of the deployed virtual machine manager), because the image instance of the software application is deployed onto the virtual machine (*i.e.*, the unallocated node that is executing the virtual machine manager). *See id.* at 63:10-64:6; 73:2-74:3.

70. I agree with Dr. Shenoy's characterization of the teachings of the '138 Patent with respect to its description of the three-step deployment process.

VIII. THE CHALLENGED CLAIMS REQUIRE THE SAME THREE-STEP DEPLOYMENT PROCESS DISCLOSED IN THE SPECIFICATION OF THE '138 PATENT

71. The Challenged Claims cover inventions where the process of deploying a virtual machine manager and the additional application software is divided into three steps: (1) copying a virtual machine manager image instance onto

a target node; (2) executing the virtual machine manager to create a working virtual machine (as that term is used by the '138 Patent); and (3) copying a software application image into the emulated computer platform created by the virtual machine.

A. Claim 1

72. In this regard, an analysis of Claim 1 is instructive:

1. A distributed computing system comprising:

a software image repository comprising non-transitory, computer-readable media operable to store: (i) a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes, wherein **when executed on the application nodes, the image instances of the virtual machine manager provide a plurality of virtual machines**, each of **the plurality of virtual machine** operable to provide an environment that emulates a computer platform, and (ii) a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines; and

a control node that comprises an automation infrastructure to provide autonomic deployment of the plurality of image instances of the virtual machine manager on the application nodes **by causing the plurality of image instances of the virtual machine manager to be**

copied from the software image repository to the application nodes and to provide autonomic deployment of the plurality of image instances of the software applications on the virtual machines by causing the plurality of image instances of the software applications to be copied from the software image repository to the application nodes.

Ex. 1001 at 36:26-50 (emphasis added).

73. In my opinion, Claim 1 requires the image instances of the software applications to be deployed on the virtual machines that are created (provided) only when the virtual machine manager image instances are executed. In other words, the virtual machines onto which the application software image instances are deployed must be the **same** virtual machines that are provided by the virtual machine managers executing on the application nodes. As such, my opinion is that Claim 1 requires the operations of the three-step deployment process to be performed in the order recited by Claim 1: (1) the automation infrastructure of the control node provide autonomic deployment of the plurality of image instances of the virtual machine manager on the application nodes by causing the plurality of image instances of the virtual machine manager to be copied from the software image repository to the application nodes, (2) the image instances of the virtual machine manager execute on the application nodes to provide virtual machines, and (3) the automation infrastructure of the control node provides autonomic deployment of the

plurality of image instances of the software applications on the virtual machines by causing the plurality of image instances of the software applications to be copied from the software image repository to the application nodes.

74. As I explain below, this opinion is based on both the logic and grammar of Claim 1, as well as the teachings of the description and figures of the '138 Patent, which require the operations to be performed in the order recited by Claim 1.

75. Claim 1 recites several limitations that are entirely consistent with the teachings in the description of the '138 Patent. First, the software repository recited by Claim 1 must store at least two separate sets of image instances: (1) a set of image instances of a virtual machine manager and (2) a set of image instances of software applications. This is clear both from the text of Claim 1 as well as from the image creation and storage process I describe above. I will note that Dr. Shenoy agrees that Claim 1 requires separate storage of these two sets of image instances. *See* Ex. 2002 at 78:2-80:2. Thus, I believe there is no material dispute between Dr. Shenoy's opinion and mine that Claim 1 requires the images of the virtual machine manager to be stored separately from the images of the software applications.

76. Another requirement of Claim 1 is that each of the image instances of the virtual machine manager must be capable of providing one or more virtual machines on the application node(s). In my opinion, this is clear from the text of Claim 1 itself: "when executed on the applications nodes, the image instances of the

virtual machine manager provide a plurality of virtual machines." Ex. 1001 at 36:31-33.

77. In my opinion, the only reasonable reading of this text is that a required characteristic of the virtual machine manager is that it is executed on the application nodes to provide the virtual machines to which the image instances of the software applications are deployed (as I describe in further detail below).

78. Once again, I note that there appears to be no material dispute between my opinion and that of Dr. Shenoy in this regard. *See* Ex. 2002 at 84:22-86:11. I note that Dr. Shenoy argued that, as used in Claim 1, "executable is not the same as executing." *Id.* at 80:14-81:16. Nonetheless, Dr. Shenoy does acknowledge that Claim 1 requires the control node to "provide autonomic deployment of a plurality of image instances of the software applications on the virtual machines." *See* Ex. 1001 at 36:45-47 (emphasis added).

79. As I explain below, this limitation is important to understanding the entirety of Claim 1, for two reasons. First, this limitation requires that the image instances of the software applications are deployed "**on**" the virtual machines. Second, this limitation requires that these image instances are deployed on "**the**" virtual machines.

80. With regard to this limitation, it is important to understand that Claim 1 recites two primary elements: a storage element recited with respect to the

functionality of the software repository, and a deployment element related to the operation of the control node. The storage element describes the storage of image instances, how they are organized, and how the image instances work. *See id.* at 36:27-38. The deployment element describes operations taken by the control node to effect the deployment of these image instances to an application node. *See id.* at 36:39-49. When describing, in the deployment element, the functionality of the control node, Claim 1 relies on the descriptions of the image instances provided in the storage element. *See id.* I will explain further below.

81. The storage element recites, "a plurality of image instances of a virtual machine manager," and that is the first time this term appears in Claim 1. *Id.* at 36:28-29. In describing further characteristics of this plurality of image instances, Claim 1 recites, "**when executed** on the application nodes, the image instances of the virtual machine manager **provide a plurality of virtual machines.**" *Id.* at 36:31-33 (emphasis added). This language is important because (1) it refers to the same plurality of image instances of the virtual machine manager (based on the antecedent basis for that term), and (2) it describes what happens when these image instances execute on an application node. Because this phrase is first found in the storage element of Claim 1, the phrase "when executed" is appropriate. While necessary to describe this characteristic of the plurality of image instances of the virtual machine manager in this element for purposes of clarity, it is clear from the context of the

claim that when the image instances of the virtual machine manager are stored in the software repository, these images obviously cannot be executing on the application nodes.

82. However, this characterization of the image instances of the virtual machine manager in the storage element provides important context for the operation of the deployment element, in which Claim 1 requires that the control node (1) provide autonomic deployment of the plurality of image instances of the virtual machine manager onto the application nodes and (2) "provide autonomic deployment of a plurality of image instances of the software applications **on the virtual machines.**" As I note above (and Dr. Shenoy agrees), the only way to create virtual machines disclosed in the '138 Patent is for a virtual machine manager to execute on an application node. *See* Ex. 2002 at 33:12-21.

83. Accordingly, my opinion is that the logic of Claim 1 requires that the control node first deploy the image of the virtual machine manager, which executes on the application node to which it is deployed, in order to create the virtual machine so that the image instances of the software applications can then be deployed: If the image instances of the virtual machine manager had not been deployed and executed, the control node would be unable to deploy any image instances of the software applications **on** the virtual machines, because no virtual machine would have been provided. The grammar of Claim 1, including in particular the antecedent basis for

the terms "virtual machine managers" and "virtual machines," as well as the use of the term "on the virtual machines," merely reinforces my opinion. Thus, in my opinion, both the logic and the grammar of Claim 1 require the three-step deployment process.

84. Moreover, as described above, this process is exactly the same as the process described by the specification of the '138 Patent with regard to deployment of image instances of virtual machine managers and software applications, as Dr. Shenoy agrees. *See supra* Section VII; *accord* Ex. 2002 at 63:10-64:6, 68:12-69:2, 73:2-74:3. Specifically, the description and figures of the '138 Patent both clearly teach that virtual machines are provided (or created) when a virtual machine manager executes on an application node, and that image instances of software applications are deployed onto these provided (or created) virtual machines. *See supra* Section VII.

85. Consequently, in my opinion, a POSITA would understand the plain and ordinary meaning of the phrase "autonomic deployment of the plurality of image instances of the software applications on the virtual machines" to refer to deployment of those image instances on the virtual machines that are provided when the virtual machine managers are executed on the application nodes, in accordance with the first use of the term "virtual machines."

86. I understand that Dr. Shenoy has not offered an opinion on whether the virtual machines onto which the image instances of software applications are deployed (in Claim 1) are the same as the virtual machines provided by execution of the virtual machine managers on the application nodes. *See* Ex. 2002 at 86:12-93:12, 100:2-101:18. For the reasons I discuss above, I disagree with Dr. Shenoy on this point with respect to Claim 1.

87. It is apparent to me that our differing interpretations of Claim 1 in this regard underlie a part of our difference of opinion on the validity of Claim 1 (and the other Challenged Claims) over the references cited by Petitioner. As I will explain in more detail below, I find no disclosure or suggestion in any of those references of the three-step deployment methodology recited by the Challenged Claims. As I contrast below, I believe Dr. Shenoy's opinion on the validity of the Challenged Claims is founded on the view that the Challenged Claims do not require the three-step deployment process that we both agree the '138 Patent discloses.

88. In comparing the language of the specification of the '138 Patent with the language of Claim 1, I observe that the specification often describes the execution of the virtual machine manager as "creating" the virtual machine, while Claim 1 recites, "when executed on the applications nodes, the image instances of the virtual machine manager **provide** a plurality of virtual machines. *Compare* Ex. 1001 at 35:33-36 *with* Ex. 1001 at 36:32-33. After carefully reviewing the

entirety of the '138 Patent, however, it is my opinion that the '138 Patent uses these terms interchangeably with regard to the relationship between the virtual machine manager and the virtual machine. For example, the '138 Patent states:

The term virtual machine is used herein to refer to software that **creates** an environment that emulates a computer platform. For example, virtual machines 40 **provide** environments on which guest operating systems (OS) execute as though the operating systems were operating directly on a physical computing platform.

Id. at 32:67-33:6 (emphasis added). My understanding is that Dr. Shenoy has not provided a contrary opinion. *See* Ex. 2002 at 22:4-23:2.

89. It therefore is my opinion that a POSITA would read the limitation of Claim 1, "wherein when executed on the applications nodes, the image instances of the virtual machine manager **provide** a plurality of virtual machines," to unambiguously be the same functionality taught in the description, in which the virtual machine manager "**creates**" a virtual machine on an application node. Ex. 1001 at 32:67-33:6, 36:31-33 (emphasis added).

B. Claim 19

90. Claim 19 is a method claim, which I have reproduced below:

19. A method comprising:

storing a plurality of image instances of a virtual machine manager that is executable on a plurality of application

nodes in a distributed computing system, wherein the image instances of the virtual machine manager provide a plurality of virtual machines, each virtual machine operable to provide an environment that emulates a computer platform;

storing a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines;

executing a rules engine to perform operations to autonomically deploy the image instances of the virtual machine manager on the application nodes by causing the image instances of the virtual machine manager to be copied to the application nodes; and

executing the rules engine to perform operations to autonomically deploy the image instances of the software applications on the virtual machines by causing the image instances of the software applications to be copied to the application nodes.

Ex. 1001 at 38:1-21.

91. Claim 19 recites functionality and uses language similar to Claim 1, particularly with respect to limitations that require the three-step deployment process discussed above with respect to Claim 1. For that reason, it is my opinion that Claim 19 requires this three-step deployment process for the same reasons.

92. Because Claim 19 is a method claim, however, it is even more apparent to me that this three-step deployment process is required by the language of Claim 19. For example, Claim 19 expressly recites, in separate limitations, (1) storing a plurality of image instances of virtual machine managers and (2) storing a plurality of image instances of software applications. *See id.* The use of two different storing limitations in Claim 19 provides further evidence that Claim 19, like Claim 1, is intended to cover the embodiments disclosed in the specification, in which the two types of image instances are stored separately.

93. Similarly, Claim 19 expressly recites two "executing" limitations, one of which is directed to autonomically deploying the image instances of the virtual machine managers to application nodes, and one of which is directed to autonomically deploying the image instances of software applications on the virtual machines. *See id.* These express limitations indicate even more clearly that Claim 19, like Claim 1, is intended to cover the three-step deployment process I describe above in the context of both the specification and Claim 1.

94. In particular, it is my opinion that Claim 19 requires the plurality of image instances of the virtual machine manager to be stored separately from the plurality of image instances of the plurality of software applications that are executable on a plurality of virtual machines. It is further my opinion that the first execution step of Claim 19 performs operations to achieve the autonomic

deployment of the image instances of the virtual machine manager and the second execution step of Claim 19 performs operations to autonomically deploy image instances of the software applications on to the virtual machines. It is my understanding that Dr. Shenoy agrees with these opinions with regard to Claim 19. *See* Ex. 2002 at 93:13-103:12.

95. For the reasons stated above with regard to Claim 1, it is my opinion that the term "virtual machines" as used throughout Claim 19 refers to the same virtual machines first recited at line 6 of column 38 in Claim 19. Ex. 1001 at 38:1-21. I understand that Dr. Shenoy does not offer an opinion disputing my opinion in this regard. *See* Ex. 2002 at 103:18-104:4.

96. For all of these reasons, it is my opinion that Claim 19 requires the three-step deployment process described above with respect to Claim 1, based on both the teachings of the specification of the '138 Patent and the logic and grammar of Claim 19 itself.

C. Claim 26

97. Claim 26 is directed to non-transitory, computer-readable medium. I have reproduced Claim 26 below:

26. A non-transitory, computer-readable medium comprising instructions stored thereon, that when executed by a programmable processor:

store a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes in a distributed computing system, wherein the image instances of the virtual machine manager provide a plurality of virtual machines, each virtual machine operable to provide an environment that emulates a computer platform;

store a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines;

perform operations to autonomically deploy the image instances of the virtual machine manager on the application nodes by causing the image instances of the virtual machine manager to be copied to the application nodes; and

perform operations to autonomically deploy the image instances of the software applications on the virtual machines by causing the image instances of the software applications to be copied to the application nodes.

Ex. 1001 at 38:61-39:15.

98. Claim 26 recites a set of instructions that are executable by a processor to perform the same operations recited by the method of Claim 19. *Compare id. with* Ex. 1001 at 38:1-21. Accordingly, my analysis of Claim 19 above applies with

equal force to Claim 26, and it is my opinion that Claim 26 requires the same three-step deployment process as Claim 19.

99. In summary, each of these independent Challenged Claims requires, in accordance with the three-step deployment process discussed above, that virtual machine manager image instances be deployed, and then that software application image instances be deployed onto the virtual machines created by the virtual machine manager image instances executing on the application nodes. In other words, the virtual machine manager image instances must be deployed and executed before the software application image instances can be deployed onto the virtual machines.

IX. CLAIM CONSTRUCTION

100. With regard to whether the Challenged Claims require the operations of the three-step deployment process, it is my opinion, as stated above, that a POSITA would understand the plain and ordinary meaning of independent Claims 1, 19, and 26 to require the recited steps to be performed in the order stated. It is further my opinion that the Challenged Claims should be accorded this plain and ordinary meaning.

101. With regard to the claim construction proposed by Petitioner, my opinion is that the proposed construction is inconsistent with the specification of the '138 Patent and the language of the Challenged Claims themselves. Specifically, I

understand that Petitioner proposes a construction of the phrase "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines" that "is satisfied by, but not limited to, a plurality of virtual disk image files." Pet. at 13; *accord* Ex. 1003 ¶ 38.

102. This proposed construction appears to be founded on dependent Claim 2, which recites, "wherein the image instances of the plurality of software applications comprise virtual disk image files." Ex. 1001 at 36:50-52; *see* Pet. at 13 ("Dependent claim 2 specifies that 'the image instances of the plurality of software applications comprise virtual disk image files.' A teaching sufficient to satisfy claim 2 must also be sufficient to satisfy claim 1.") (citing Ex. 1003 ¶ 38).

103. In my view, however, this proposed construction draws a false parallel between the language of Claim 1 and Claim 2. The disputed term of Claim 1 recites, "a plurality of image instances of a plurality of software applications **that are executable on the plurality of virtual machines.**" Ex. 1001 at 36:35-37 (emphasis added). Claim 2, as noted above, recites, "wherein the image instances of the plurality of software applications comprise virtual disk image files." *Id.* at 36:50-52. A simple comparison of these two terms reveals that the term "image instances of a plurality of software applications" must be broad enough to include "virtual disk files." I do not dispute this proposition. Indeed both the specification and Claim 2

indicate that a virtual disk file can be a type of image instance of a software application. *See id.* at 33:25-41, 36:50-52.

104. The mere phrase "image instances of a plurality of software applications," however, is not the phrase for which Petitioner proposes a construction. If it were, no construction would be required, because there would be no dispute. In contrast, Petitioner seeks a construction of the entire term, "a plurality of image instances of a plurality of software applications **that are executable on the plurality of virtual machines.**" Pet. at 13 (emphasis added).

105. Thus, Petitioner asks the Board to make the clear substitution: "a plurality of image instances of a plurality of software applications **that are executable on the plurality of virtual machines**" for "virtual disk image files." Ex. 1001 at 36:35-37 (emphasis added); *see also id.* at 36:52. While Claim 2 indicates that "a plurality of image instances of a plurality of software applications" can comprise "virtual disk image files," Claim 2 does not address how (or where) the control node deploys the virtual disk image files.

106. Petitioner's proposed construction essentially disregards the emphasized language in this term, without any explanation or justification. In the absence of any rationale from Petitioner or Dr. Shenoy, I cannot agree that a POSITA would understand the disputed claim term to exclude the language emphasized above.

107. In my opinion, Petitioner's proposed construction would read out of the Challenged Claims not only the emphasized language above, but also the three-step deployment process. As I explain above, the emphasized language in the disputed term, "a plurality of image instances of a plurality of software applications **that are executable on the plurality of virtual machines,**" (and the similar phrase "on the virtual machines" in the deployment element) play a significant role in dictating the three-step deployment process of the Challenged Claims. In fact, as I explain above, this language is part of the logic and grammar of Claim 1 that requires the three-step deployment process.

108. The reason for this is simple – "on the plurality of virtual machines" dictates where the image instances of the software applications must be deployed (*i.e.*, "on" the virtual machines that are on the application nodes after the second step of the three-step deployment process described above). *See supra* Section VIII.A.

109. Moreover, the phrase "that are executable on the plurality of virtual machines" refers to "a plurality of virtual machines" that provides antecedent basis for (and thus defines) "virtual machines." If adopted by the Board, Petitioner's disregard for the principles of antecedent basis in its proposed construction would render the phrase "that are executable on the plurality of virtual machines" superfluous. I understand that a construction that renders claim language superfluous is disfavored under the law, and this is an additional reason I cannot

agree with Petitioner's proposed construction. Therefore, for these reasons, including the many reasons discussed above with respect to the ordered sequence of steps required by the Challenged Claims, I disagree with Petitioner's proposed construction and instead believe a POSITA would understand this term as written, without any further construction necessary.

X. THE ASSERTED PRIOR ART

110. It is my opinion that there is no combination of the cited references (Khandekar, Le, Sidebottom, and Stone) that teaches each and every limitation of any Challenged Claim, for the reasons I discuss below.

A. Khandekar

111. Dr. Shenoy identifies Khandekar as the primary reference for each of his four Grounds. *See* Ex. 1003 ¶ 23. With respect to Ground 1, Dr. Shenoy alleges that Khandekar renders independent Claims 1, 19, and 26, and dependent Claims 2-5, 9, 20, and 21, obvious under 35 U.S.C. § 103. *See id.* I will first provide a description of Khandekar's teachings, and I will then explain why each of the Challenged Claims is valid over the disclosure of Khandekar.

1. Background on Khandekar

112. In general, Khandekar discloses "a method and system implementation for creating a virtualized computer system based on input information identifying a desired computer configuration." *See* Ex. 1004 at 4:30-32. More specifically,

Khandekar discloses inventions related to a provisioning server that a user can interact with to obtain access to a virtual computer with user-requested features. *See* Ex. 1004 at 9:26-27 (the "main feature of the invention is a VM provisioning server"). As Khandekar discloses:

A provisioning server automatically configures a virtual machine (VM) according to user specifications and then deploys the VM on a physical host. The user may either choose from a list of pre-configured, ready-to-deploy VMs, or he may select which hardware, operating system and application(s) he would like the VM to have. The provisioning server then configures the VM accordingly, if the desired configuration is available, or it applies heuristics to configure a VM that best matches the user's request if it isn't.

Ex. 1004, Abs.; *see also id.* at 9:26-27; 9:65-10:1.

113. At the outset, it is important to note that the term "virtual machine (VM)," used by Khandekar, does not have the same meaning as "virtual machine," as used by the '138 Patent. As described above, the term "virtual machine" as used in the '138 Patent refers to "software that creates an environment that emulates a computer platform" on an application node. Ex. 1001 at 32:67-33:2 and 36:34-36. In this regard, a "virtual machine" in the '138 Patent is an emulated "bare-metal" computer without any operating system or applications installed. *See supra* ¶¶ 29,

51-52. A "virtual machine" in the '138 Patent is created by a virtual machine manager on an application node. Ex. 1001 at 35:6-10 (emphasis added); *see also supra* Section V. In contrast, Khandekar teaches that a "a 'model virtual machine' is a virtual machine (VM) that contains a certain configuration of guest operating system 320, service pack, and applications 360, where the applications have been tested and are known to work correctly." Ex. 1004 at 8:51-56. In other words, Khandekar uses the term VM (or virtual machine) in a similar manner as what the '138 Patent describes as a virtual disk image file. *See* Ex. 1001 at 33:25-41 ("The operating system data and application data, as well as instance-specific configuration data for the underlying virtual machines 404, is stored as respective virtual disk image files 408. From the perspective of the guest operating system, each of virtual disk image files 408 appears as a bootable hard disk of the host computer. In other words, the virtual disk image file provides a bootable software image of the operating system and applications, and includes all necessary configuration data and file structures.").

114. While Khandekar discloses various types of VMs (pre-built, custom-built, and instantly deployable) (*see* Pet. at 41), none of these VMs are "software that creates an environment that emulates a computer platform" on a node, which is how the '138 Patent uses the term. Ex. 1001 at 35:6-10. Rather, each type of VM disclosed by Khandekar is created in the provisioning server with an operating

system and (in most cases) software applications. Ex. 2002 at 108:17-109:13; *see infra* ¶¶ 132-37. As I note above, both FIGS. 24 and 26 indicate that the virtual machine is separate from, and underlies, the operating system and software applications. *See supra* ¶¶ 29, 51-52; *accord* Ex. 1001, FIG. 24 (illustrating the virtual machines 404 separately from the operating system and applications 408), FIG. 26 (showing the installation of the operating system (step 424) and applications (step 426) in the process of capturing an image instance of software applications/virtual disk image).

115. Khandekar also discloses a "'virtual machine monitor' (VMM)," which Khandekar describes as "usually a thin piece of software that runs directly on top of the host and virtualizes all, or at least some of, the resources of the machine, or at least of some machine." Ex. 1004 at 7:4-9. Khandekar provides little disclosure of how the VMM is deployed, instead presuming almost universally that each host already has a VMM installed before the deployment process begins. With regard to how the VMM arrives on the host, Khandekar discloses only the following:

In the following description of the invention, merely for the sake of simplicity, only one VM/VMM pair is discussed. The discussion applies equally, however, to all such VM/VMM pairs that may be included in any given implementation of the invention. Moreover, **it will be assumed below that when a VM is installed on a host, then a corresponding VMM is either already installed**

on the host, or is included and installed together with the VM.

Id. at 8:36-43 (emphasis added).

As is mentioned above, a VMM is typically included for each VM in order to act as its software interface with the underlying host system. **In the preferred embodiment of the invention, a VMM is already pre-installed on each host 1000, 1001 so as to ensure compatibility and proper configuration.** Several products are available from VMware, Inc. of Palo Alto, Calif., that configure conventional computers, both stand-alone desktop computers and multi-user servers, to load and run multiple VMs. **It would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known. An advantage of pre-installing a VMM, without a VM, on a plurality of hosts, such as hosts 1000, 1001, 1003, is that it will then be possible to deploy a given VM on any arbitrary one (or more) of these hosts. Moreover, as long as the VMM on each host exports a known hardware interface to the VM deployed on it, then VM deployment will be substantially independent of the actual physical hardware configuration.**

Id. at 13:17-35 (emphasis added).

116. In my opinion, it is important to recognize the differences between the meaning and use of the terms VM and VMM in Khandekar as opposed to "virtual machine," "virtual machine manager," and "software applications" in the '138 Patent when analyzing the Khandekar reference and attempting to apply its teachings to the Challenged Claims, as I explain further below.

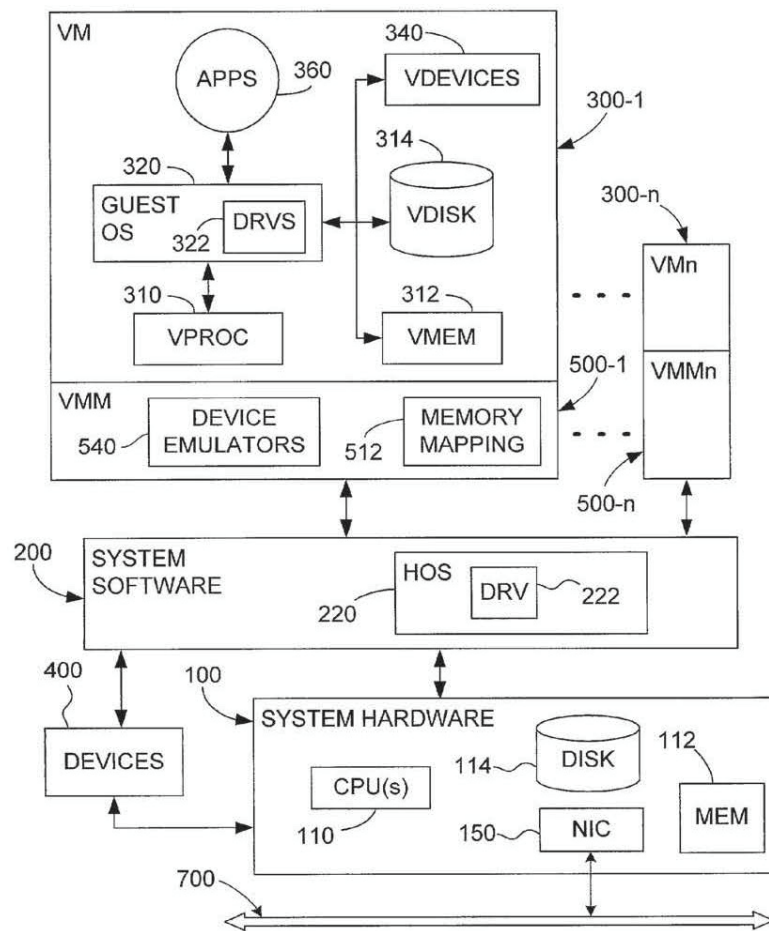
2. Application of Khandekar to the Challenged Claims

a) The Three-Step Deployment Process

117. Khandekar does not disclose the three-step deployment process required by the Challenged Claims. *See supra* Section VIII. Instead, as Dr. Shenoy correctly acknowledges, Khandekar teaches that the VMs are assembled on the provisioning server. *See* Ex. 2002 at 108:17-109:13. Rather than a three-step process, Khandekar expressly states that the deployment process is a single step: "'Deployment' is the step of taking a staged VM, copying it to a host machine and letting it configure with a new identity." Ex. 1004 at 8:64-67 (emphasis added).

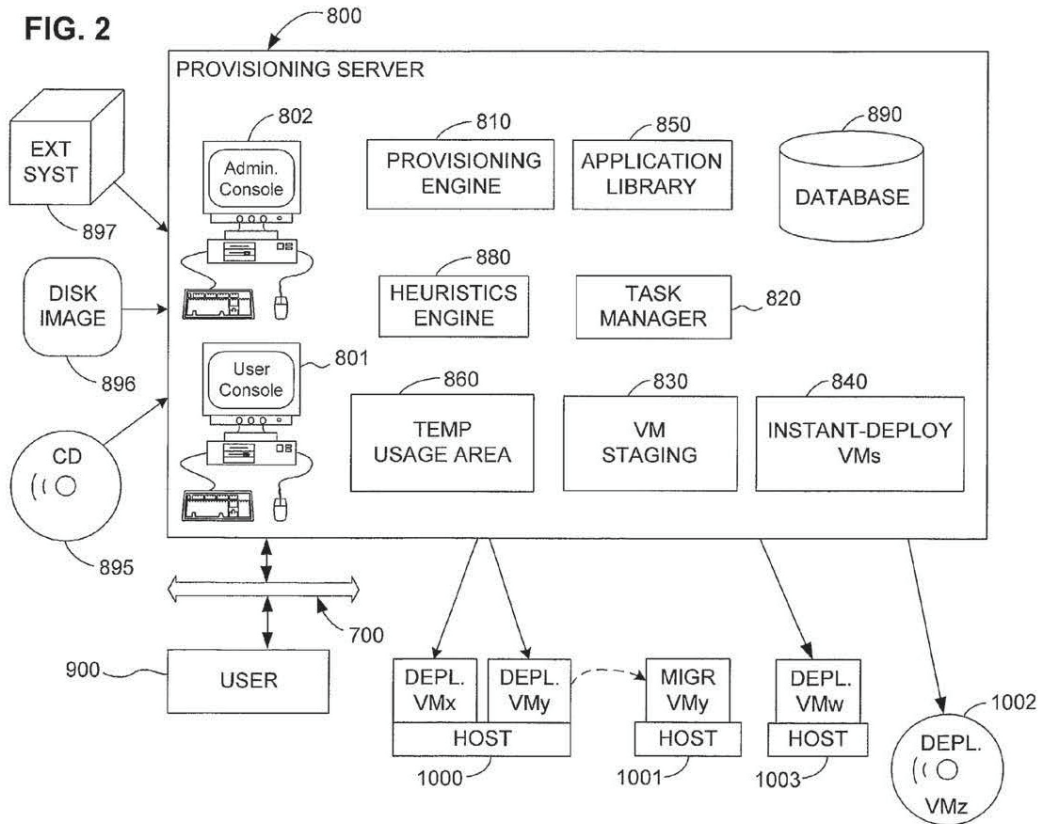
118. An analysis of Petitioner's assertions with regard to what it describes as limitations [1d] and [1e] is revealing. *See* Pet. at 37-47. Limitation [1d] requires, among other things, "when executed on the applications nodes, the image instances of the virtual machine manager provide a plurality of virtual machines." In alleging Khandekar discloses this portion of Claim 1 (limitation [1d]), Petitioner relies heavily on FIG. 1 of Khandekar, which I have reproduced below:

FIG. 1
(Prior Art)



Ex. 1004, FIG. 1; *see* Pet. at 37-38.

119. In contrast, when discussing limitation [1e] ("plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines"), Petitioner relies on Khandekar's disclosure of different types of VMs, which are illustrated on FIG. 2 of Khandekar, which I have also reproduced below:



Ex. 1004, FIG. 2; *see* Pet. at 41-48.

120. I note that FIG. 1 of Khandekar is labeled "Prior Art" and thus is not Khandekar's invention. It certainly is not part of the invention disclosed in FIG. 2. In describing FIG. 1, Khandekar offers no explanation (and indeed no disclosure at all) of how the VMMs (500-1 and 500-n) might have arrived on the hosts (whether through deployment or any other method). In my opinion, therefore, Petitioner does not explain how FIG. 1 teaches that VMMs (500-1 and 500-n) can provide "a plurality of virtual machines" on the application nodes. In my opinion, element [1d] of Claim 1 is significant because it recites how the virtual machine is created. Neither FIG. 1 nor any of the accompanying description (all of which are "Prior Art"

within the disclosure of Khandekar) provide any disclosure of how the disclosed VMM creates or provides the VM. It is my opinion, therefore, that Petitioner has not demonstrated that Khandekar satisfies limitation [1d].

121. Turning to limitation [1e], Petitioner proposes that Khandekar's disclosure related to pre-built, custom-built, and instantly deployable VMs satisfies that limitation, which requires "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." Pet. at 41-42. As noted above, however, Khandekar discloses that VMs are deployed from the provisioning server in a single step. *See* Ex. 1004 at 8:64-67. Moreover, as Petitioner notes, Khandekar teaches that these applications are part of the VM itself, and are installed into the VM on the provisioning server before deployment of the VM. *See* Pet. at 42-47. Petitioner does not explain how these software applications, which already have been installed on the VMs prior to deployment of the VMs, can be deployed on "the virtual machines" disclosed in the "Prior Art" portion of Khandekar's disclosure.

122. In my opinion, as noted above, the language of the Challenged Claims requires (under the principle of antecedent basis) that the virtual machines onto which the image instances of software application are deployed must be the same virtual machines that are created/provided when the virtual machine manager executes on the application node. *See supra* Section VIII.A. Petitioner's analysis

appears to require that the virtual machines onto which the software applications are deployed (virtual disk files stored on the provisioning server) are different from the virtual machines disclosed (in Khandekar's prior art FIG. 1). For the reasons I discuss above, I find this interpretation of the Challenged Claims untenable. *See id.*

123. As such, there really is no dispute that Khandekar fails to disclose or suggest the three-step deployment process that, as I discuss above, is required by the Challenged Claims. *See supra* Section VIII. Petitioner does not argue otherwise. Instead, Petitioner selects different portions of the Khandekar reference to apply against each limitation (or each part of a limitation) of the Challenged Claims in isolation, without regard to the overall claim structure. As I noted, I do not believe that the logic or grammar of the Challenged Claims (including the principle of antecedent basis) permits this piecewise analysis. Nor do I believe that the ordering required by the specification permits the three steps of the three-step deployment process to be analyzed outside of their context together in the Challenged Claims.

b) Storage of Image Instances

124. Claim 1 recites in part that:

a software image repository comprising non-transitory, computer-readable media operable to store: (i) a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes, wherein when executed on the applications

nodes, the image instances of the virtual machine manager provide a plurality of virtual machines, each of the plurality of virtual machine operable to provide an environment that emulates a computer platform, and (ii) a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines

Ex. 1001 at 36:27-37. As I discuss below, it is my opinion that neither Khandekar nor any other reference cited by Petitioner discloses or suggests this limitation, for multiple reasons.

125. First, Khandekar does not even suggest a "software image repository" that stores "a plurality of image instances of a virtual machine manager" as set forth in Claim 1. *See id.* at 36:27-29. Rather, Khandekar teaches that its preferred embodiment presumes that the VMM is "pre-installed on each host." Ex. 1004 at 13:17-35. Khandekar does not provide any disclosure of how this pre-installation might occur. A POSITA would have understood that any number of possibilities would have existed, including factory installation of the VMM software, manual installation of the VMM on each host, *etc.* There is no suggestion in Khandekar that this pre-installation might be performed by deployment of an image instance.

126. I note that Petitioner disregards this preferred embodiment relying instead on a single sentence in Khandekar that teaches, "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, as

long as the characteristics of the host on which the VM is to be installed are known." Pet. at 27 (quoting Ex. 1004 at 13:25-28). From this sentence, Petitioner argues that "a POSITA would have found it obvious to likewise store a plurality of VMM images in the provisioning server so they may be deployed together to hosts, as taught by Khandekar." Pet. at 27.

127. I note, however, that, in the very next sentence, Khandekar refers back to its preferred embodiment: "An advantage of pre-installing a VMM, without a VM, on a plurality of hosts. such as hosts 1000, 1001, 1003, is that it will then be possible to deploy a given VM on any arbitrary one (or more) of these hosts." Ex. 1004 at 13:28-32.

128. In support of its position that a POSITA would have found it obvious to store image instances of virtual machine managers in the software repository, Petitioner relies on a single paragraph from Dr. Shenoy's declaration. *See* Pet. at 27 (citing Ex. 1003 ¶ 83). In the preceding paragraph of this declaration, Dr. Shenoy quotes Khandekar's speculation that "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known." Ex. 1003 ¶ 82 (quoting Ex. 1004 (Khandekar) at 13:25-28). From this text, Dr. Shenoy opines, "A POSITA would have found it obvious to follow these teachings from Khandekar, with VMM images being stored in the provisioning server and provided

to hosts together with VM images." *Id.*, ¶ 83. I respectfully disagree with this opinion, which I believe finds no support in the actual disclosure of Khandekar.

129. Even assuming that an isolated, single sentence, which Khandekar immediately disregards, might lead a POSITA to try deploying a VMM together with a VM (an opinion I do not share), my opinion is that the passage provides no teaching or suggestion whatsoever about how a VMM might be stored, let alone teaching "VMM images being stored in the provisioning server," as Dr. Shenoy opines. *Id.*

130. In my opinion, Petitioner identifies no disclosure in any prior art, including the references cited as part of the grounds for review (Khandekar, Le (Ex. 1005), Sidebottom (Ex. 1006), and Stone (Ex. 1007)), as well as other references mentioned in the Petition (such as Powell (Ex. 1014) and French (Ex. 1016)), that suggest storing an image of a virtual machine manager in a software repository, let alone storing a plurality of such images. Petitioner quotes the Powell reference as teaching that a hypervisor can be loaded as part of an operating system or firmware image. *See* Pet. at 20 (quoting Ex. 1014, ¶ 0062). I find nothing in this disclosure that suggests virtual machine manager images might be stored on a software repository. Similarly, I find Petitioner's reliance on French's teaching that a "bootable core and the virtual machine manager image must be installed in a secure partition," to be unavailing; once again, Petitioner identifies nothing in the reference

that suggests this image might be stored on a software repository. *Id.* (quoting Ex. 1016, ¶ 0049). Moreover, I note that neither of these references was listed as part of the grounds asserted against the Challenged Claims.

131. Further, even assuming (contrary to my opinion) that a POSITA might have found it obvious to store an image of a virtual machine manager in a software repository, I see no evidence in the record that image instances of such a virtual machine manager might be capable of being executed on an application node to create (provide) a virtual machine, in the sense Khandekar uses that term. Again, in the context of the Challenged Claims, a virtual machine is a bare metal platform onto which operating systems and applications can be deployed, but it does not include the software itself. *See supra* ¶¶ 29, 52. Petitioner identifies nothing in Khandekar that suggests that an image of a virtual machine manager stored in a software repository might have such a capability. Rather, as I note above, the VMs deployed (in a single step) according to Khandekar's teachings already include an operating system and applications. *See supra* ¶ 114; *accord infra* ¶¶ 132-37. Thus, these VMs cannot also serve as the virtual machines provided/created by execution of the virtual machine manager on an application.

132. The storage limitation of Claim 1 also requires the software repository to store "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." Ex. 1001 at 36:35-37. As I explain

above, the VMs disclosed by Khandekar already include installed operating systems and software applications. *See supra* ¶¶ 114, 131. As such, Petitioner's reliance on Khandekar's VMs to disclose both "virtual machines" and "image instances of software applications" is inconsistent. *See Pet.* at 41-47 (citing only various disclosures related to the pre-built, custom-built, and instantly deployable VMs disclosed in Khandekar to teach both virtual machines and software application images). This limitation of Claim 1 requires both "virtual machines" created by the execution of a virtual machine manager and "image instances of software applications" that deploy "on the virtual machines." The VMs disclosed by Khandekar cannot serve as both of these entities. In effect, the term "VM" in Khandekar eliminates any distinction between the "image instance of a software application" and the "virtual machine" in the '138 Patent by combining them into a single concept that is neither wholly one nor wholly the other.

133. In my opinion, this inconsistency prevents the Petitioner from proving that any Challenged Claim is invalid, regardless of whether the Challenged Claims require a three-step deployment process.

134. Petitioner argues that Khandekar's disclosure related to pre-built, custom-built, and instantly deployable VMs satisfies element [1e] that requires a "plurality of image instances of a plurality of software applications that are executable on **the** plurality of virtual machines." *Id.* at 41-42 (emphasis added). My

opinion is that none of those different types of VMs disclose the limitation 1[e], even apart from the inconsistency I note above between the use of VM to mean both an image instance of an application and a virtual machine. As I explain below, Petitioner's argument that these types of virtual machines teach deployment of an instance of a software application is mistaken, and any such argument is controverted by Khandekar itself as well as Petitioner's own concessions regarding the proper construction of claim element [1e].

135. First, with regard to the "instant deploy" VMs, the Khandekar specification explains that this disclosure does not describe deployment of an image instance at all, but rather the reallocation of already-running VMs. *See* Ex. 1004 at 9:23-25 ("Instantly provision a VM by resuming it from a suspended state. Such a VM is referred to below as a [sic] an 'instant deploy' VM."). As described by Khandekar, instant-deploy VMs are fully configured and thus do not include the step of "provid[ing] the autonomic deployment of the plurality of image instances of the software applications on the virtual machines . . . ," as required by the '138 Patent. *See id.* at 17:31-33 ("This component of the provisioning server 800 contains fully configured VMs that are suspended and are ready to be instantly deployed.").

136. Second, with regard to the "pre-built" VMs, Khandekar discloses that, "[i]n the preferred embodiment of the invention, pre-built VMs will be stored with their operating systems and applications installed" *Id.* at 15:43-45. This

directly contradicts the Challenged Claims, which require that the software application image instances be deployed on the virtual machines as the final step of the three-stage deployment process.

137. Third, with regard to the "custom built" VMs, Khandekar discloses that "custom built VMs will be stored with just their operating systems installed." *Id.* at 15:45-46. To provision a custom built VM, Khandekar discloses that a provisioning engine "creates the custom answer file 864 and the application installation software 862." *Id.* at 21:46-50. As Khandekar explains, this "involves running an executable installation file 851 [provided by the application vendor] that takes input from a text file, called an 'answer file,' that specifies the parameters for installation for that application (for example, the folder to install it in) and the installation files/data 853 that need to be copied to the machine (here, virtual machine) that the application is to be installed on." *Id.* at 18:61-19:2. This "executable installation file," *i.e.*, the file used to install an application for the first time, is not equivalent to the "image instances of software applications" required by the '138 Patent. As Netflix admits, an "image instance" as used in the '138 Patent refers to "a snapshot, or copy, **installed and configured software.**" Ex. 2003 Joint Claim Construction and Prehearing Statement, *Broadcom Corp. et al. v. Netflix, Inc.*, No. 3:20-cv-04677-JD (N.D. Cal. May 18, 2021), ECF No. 112 at 2 (emphasis added). A software application image instance as required by the '138 Patent thus provides a snapshot

of a pre-installed and configured application at a particular state, thereby allowing a user to recreate or resume an operation without the need for any additional configuration details. By contrast, an "executable installation file" that an application vendor provides merely performs a basic installation of an application without specific configuration details and application state data. This is not the same as an application image instance, and, in my opinion, no POSITA would recognize them as equivalent. Thus, it is my opinion that Khandekar's disclosures related to pre-built, custom-built, and instantly deployable VMs do not satisfy element [1e]'s requirement for a "plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines."

138. Finally, with regard to the storage of image instances, I note above that the Challenged Claims require the image instances of the virtual machine managers to be stored separately from the image instances of the software applications. *See supra* ¶¶ 43-44, 53-54. To the extent Khandekar possibly could be read as storing a virtual machine manager image at all, any assertion by Petitioner that such storage might be separate from the storage of Khandekar's VMs themselves is pure speculation. Indeed, the only justification for the proposition that Khandekar's VMMs could be stored as images at all is based on Khandekar's bare mention that the VM and VMM can be deployed together in a single step. I find nothing in that teaching that would indicate the VMM and VM might be stored separately in that

situation. Any inference that they might be stored separately would seem to be driven by hindsight, in light of the disclosure of the '138 Patent itself.

139. Not only does Khandekar lack an explicit teaching that images of a VMM would be stored separately from images of VMs (*see supra* ¶¶ 43-44, 53-54, 138), I believe that Khandekar teaches that including a VMM with a VM requires including the VMM *within the same image* as the VM. As stated in Khandekar, "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, **as long as the characteristics of the host on which the VM is to be installed are known.**" Ex. 1004 at 13:25-28 (emphasis added). The highlighted phrase teaches an understanding whereby the included VMM *depends directly* on the VM with which it is to be associated, and that dependence thereby constrains the choice of VMM to include with the VM. If the VMM were imaged and stored separately from the VM, then there would be no dependence between the VMM and the VM and therefore no need to know *a priori* the characteristics of the host on which the VM is to be installed.

140. I believe that this teaching of Khandekar is further reinforced by the fact that Khandekar teaches that VMMs and VMs are always to be considered in pairs:

In the following description of the invention, merely for the sake of simplicity, **only one VM/VMM pair** is discussed. The discussion applies equally, however, to **all**

such VM/VMM pairs that may be included in any given implementation of the invention. Moreover, it will be assumed below that when a VM is installed on a host, **then a corresponding VMM is either already installed on the host, or is included and installed together with the VM.**

Id. at 8:36-43 (emphasis added). Khandekar thus teaches that any VMM included with a VM is included in such a way that the VMM and VM are bound together as a pair. Furthermore, logic and grammar dictate that the phrase "included and installed together" quoted above must be read as "*included together* and installed together", thereby teaching that whenever a VMM is included with a VM, it is *included together with the VM in the same image*, since the only VMs described by Khandekar as being "installed" on a host are those VMs that have been captured in an image. This view of a VMM and VM being bound in pairs is further reinforced by Khandekar's teaching that "from the perspective of the system designer **the VM and VMM can be considered to form a single cooperating system.**" *Id.* at 7:40-42 (emphasis added). In other words, a VM/VMM pair forms a single cooperating system.

141. In light of these teachings of Khandekar, I believe that not only is it unfounded speculation for the Petitioner to assert that a POSITA would store images of a VMM separately from images of a VM, but in order to do so, the POSITA must

perform a creative step that *contradicts* the teachings of Khandekar. This creative step – the storage of images of a virtual machine manager for subsequent autonomic deployment – is disclosed only in the '138 Patent.

**c) Autonomic Deployment of the Image Instances of
Software Applications**

142. Claim 1 further requires a control node with an automation infrastructure "to provide autonomic deployment of the plurality of image instances of the software applications on the virtual machines by causing the plurality of image instances of the software applications to be copied from the software image repository to the application nodes." Ex. 1001 at 36:45-59. In my opinion, Petitioner fails to establish that Khandekar satisfies this limitation. This limitation, like the storing limitation discussed above, requires both "virtual machines" created by the execution of a virtual machine manager and "image instances of software applications" that deploy "on the virtual machines." As I note above, the VMs disclosed by Khandekar cannot serve as both of these entities. *See supra* ¶¶ 132-37.

143. In my opinion, this further inconsistency prevents the Petitioner from proving that any Challenged Claim is invalid, regardless of whether the Challenged Claims require a three-step deployment process.

d) Advantages of the Claimed Invention over Khandekar

144. As I describe above, the techniques disclosed by the '138 Patent and captured by the Challenged Claims provide significant advantages over the prior art, including in particular Khandekar. *See supra* ¶ 55.

145. As I note above, the technology of the Challenged Claims could accommodate 100 different configurations of physical nodes and 100 different configurations of operating systems and other applications with 200 total images (100 virtual machine manager images and 100 software application images). *See id.*

146. Khandekar, like all of the other prior art of which I am aware, would require storage of 10,000 images under the same circumstances. *See id.* The advantages provided by the Challenged Claims would not be possible with any of the embodiments described by Khandekar. Specifically, Khandekar acknowledges that it is better to pre-install a VMM on each host (which is highly inefficient) than attempt to deploy a VMM image instance with a software application image instance, because, according to Khandekar, the latter technique cannot accommodate differing configurations of the underlying nodes. *See Ex. 1004 at 13:28-32.* Even if Khandekar's disclosure of VMMs could be construed to argue that a VMM would be stored in the provisioning server (which I believe it could not), Khandekar's teaching would suggest that each VM/VMM pair would be stored together, which cannot provide the flexibility of the Challenged Claims, which require storage of the

virtual machine images separately from the software application images. *See supra* ¶ 55.

147. In light of this multiplicative growth in the number of images Khandekar requires to be stored in the case where a VMM is included with a VM, a further advantage of the '138 Patent over Khandekar is that the '138 Patent enables more efficient routine maintenance of its much lower number of image instances. As recited in Challenged Claim 1, illustrated in FIG. 24, and explained above (*e.g., supra* Section X.A.2.b), the '138 Patent teaches imaging and deployment of individual virtual machine managers and individual software applications, thereby allowing the image instances of each virtual machine manager and each software application to be maintained independently of the others. Updating or "patching" a virtual machine manager or software application requires re-imaging only that virtual machine manager or software application, without having to re-image any other virtual machine manager or software application. Furthermore, only the updated virtual machine manager image or software application image needs to be deployed to any application node currently using the non-updated version, without having to re-deploy instance images of any other virtual machine manager or software application. In contrast, in Khandekar's embodiments, any update to a VMM or software application stored inside a VM image requires all such VM

images to be deployed to an imaging server and then rebuilt in full once the updated version has been substituted for the old.

B. Le

148. As I have described above, Khandekar fails to disclose or suggest several limitations of the Challenged Claims, including but not limited to the three-step deployment process required by those claims. I find no disclosure in Le that would remedy any of the deficiencies in Khandekar.

149. With regard to the installation of a VMM image instance, Le teaches only that, "[t]o power on and run the cloned virtual machine, all that the host computer needs is to have the appropriate VMM software installed." *See* Ex. 1005 (Le) at 14:28-30. This neither teaches nor suggests any particular technique for installing the VMM, let alone the deployment required by the Challenged Claims.

150. In other words, Le, like Khandekar, presumes that a VMM has already been installed when discussing deployment of disk images, and that reference provides no disclosure of how the pre-existing installation of a VMM might have occurred. Le also appears to teach that the "cloned virtual machine," which would correspond to the software application image instance, is copied to the host before the VMM is executed. *See id.* at 54:14-21.

151. In any case, Petitioner fails to identify any teaching or suggestion of the three-step deployment process required by the independent Challenged Claims.

Likewise, my careful review of the reference does not reveal any such teaching or suggestion.

152. For these reasons, Le fails to supplement Khandekar in any material way with regard to independent Claims 1, 19, and 26, and, in my opinion, those claims are valid over the combination of Khandekar and Le asserted by the Petitioner in ground 2 of the Petition.

C. Sidebottom and Stone

153. My analysis of the Petition indicates that Petitioner employs Sidebottom only against dependent Claims 9 and 10 in Ground 3. Similarly, Petitioner employs Stone only against Claim 17 in Ground 4.

154. In particular, Petitioner does not argue that either Sidebottom or Stone provides any material disclosure, or remedies any of the deficiencies of Khandekar and Le with regard to independent Challenged Claims 1, 19, and 26. As such, neither Sidebottom nor Stone has any effect on my opinion of the validity of Claims 1, 19, and 26. In my opinion, for the reasons stated above, those claims are valid against any combination of Khandekar, Le, Sidebottom, and Stone.

XI. CONCLUSION

155. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct and that these statements were made with the knowledge that willful false statements and the like so made are punishable

by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

156. Executed on this 21st day of June, 2021 in Fairfax, Virginia.

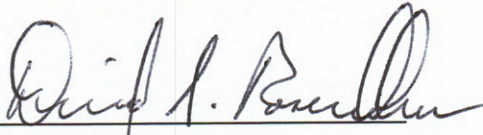
By: 
David Rosenblum

Exhibit 1

CURRICULUM VITAE

David S. Rosenblum

June 2021

Planning Research Corporation Professor and Chair
Department of Computer Science
George Mason University
4305 Nguyen Engineering Building, MSN 4A5
4400 University Drive
Fairfax, VA 22030
USA

+1 (703) 993-1530 (office)
+1 (703) 993-1710 (fax)
dsr@gmu.edu (work)
dsrosenblum@alumni.stanford.edu (personal)
<https://cs.gmu.edu/~dsr>
<https://www.linkedin.com/in/dsrosenblum>

h-index = 54, *i10-index* = 113 (Google Scholar, June 20, 2021); *Erdős number* ≤ 4

Summary of Key Accomplishments and Research Interests

Fellow of the ACM and IEEE
Recipient of the ACM SIGSOFT Distinguished Service Award
Recipient of two significant test-of-time awards for research papers
Previously Editor-in-Chief of one of the leading archival journals in software engineering (ACM TOSEM)
Previously Chair of the leading international professional group in software engineering (ACM SIGSOFT)
Produced significant, highly cited research results in formal methods, software testing, software architecture, distributed systems, ubiquitous computing, and applied machine learning
Secured tens of millions of dollars in extramural research funding
Graduated 14 PhD students to date with strong placements in both academia and industry
Held professorships and leadership positions in academia in three countries on three continents
Held research and leadership positions in industry
Current research is on novel machine learning techniques for ubiquitous computing and software testing

Honors and Awards

ACM SIGSOFT Distinguished Service Award, 2018
Fellow of the Association for Computing Machinery (ACM), 2010
"For contributions to software testing and distributed systems, and for service to the software engineering community"
Fellow of the Institute of Electrical and Electronics Engineers (IEEE), 2006
"For contributions to scalable, distributed component- and event-based software systems"
Chartered Fellow of the Institution of Engineering and Technology (IET), 2005
Chartered Fellow of the British Computer Society (BCS), 2005
Senior Member of the Singapore Computer Society (SCS), 2013
Featured in "People of ACM", Jan. 5, 2016,
<http://www.acm.org/articles/people-of-acm/2016/david-s-rosenblum>
2008 ACM SIGSOFT Impact Paper Award, ESEC/FSE 1997 paper (#C16)
2002 ICSE Most Influential Paper Award, ICSE 1992 paper (#C8)
2nd Prize, ACM SIGSOFT FSE 2014 Visions & Challenges paper (#C72)
Best Demo Award, IoT 2015 Demo (presented by Max Pagel) (#C75)
Provost's Chair Professor, Dept. of Computer Science, National University of Singapore, 2016
Wolfson Research Merit Award, The Royal Society, UK, 2004–2009
CAREER Award, US National Science Foundation, 1997–2001
Best Graduate Student Paper Award, ACM Special Interest Group on Ada (SIGAda), 1986 (#J2)
L.F. Connell Award, Best Undergraduate Student, Dept. of Physics, North Texas State Univ., 1982

Education

- Ph.D. *Electrical Engineering (Computer Systems)*, 1988
Stanford University, Stanford, CA, USA
- M.S. *Electrical Engineering (Computer Systems)*, 1987
Stanford University, Stanford, CA, USA
- M.S. *Computer Sciences*, 1983
North Texas State University, Denton, TX, USA
- B.S. (summa) *Computer Sciences (major), Mathematics (minor)*, 1982
North Texas State University, Denton, TX, USA

Professional Certifications

- Senior Management Programme, National University of Singapore, 2012
- Chartered Engineer (CEng), Engineering Council UK, 2006–2013 (registered with IET)

Academic Positions

- 2020– *Planning Research Corporation Endowed Chair (2020–present)*
Department Chair (2020–present)
Department of Computer Science, George Mason University, Fairfax, Virginia, USA
- 2011–2020 *Provost's Chair Professor, Department of Computer Science (2016–2020)*
Professor, Department of Computer Science (2011–2016)
Director, NUS-Singtel Cyber Security Research & Development Laboratory (2016–2020)
Dean, School of Computing (2013–2016)
Director of the Felicitous Computing Institute (2012–2020)
Chair, School of Computing Ethical Review Committee (2019–2020)
School of Computing, National University of Singapore, Republic of Singapore
- 2004–2011 *Professor of Software Systems*
Graduate Tutor (2008–2011)
Course Coordinator, MSc in Software Systems Engineering (2004–2007)
Director, London Software Systems (2004–2008)
Head of Software Systems Engineering (2008–2011)
Deputy Head of Department (2010–2011)
Department of Computer Science, University College London, UK
- 1998–2002 *Associate Professor (1998), Assistant Professor (1996)*
Faculty Affiliate, University of California Institute for Software Research (ISR)
Associate Chair for Computing (1998–2001)
Acting Manager, Computing Support Group (1999)
Dept. of Information & Computer Science, University of California, Irvine, CA, USA
- 1996 *Part-Time Lecturer*
Department of Computer Science, Rutgers University, Piscataway, NJ, USA
- 1983–1988 *Research Assistant*
Computer Systems Laboratory, Stanford University, Stanford, CA, USA
- 1982–1983 *Teaching Fellow*
Department of Computer Sciences, North Texas State University, Denton, TX, USA
- 1981 *Programmer/Analyst*
University of Texas Health Science Center at Dallas, , TX, USA

Industry Positions

- 2001–2003 *Chief Technology Officer and Principal Architect*
PreCache Inc., Orange, CA, and Bridgewater, NJ, USA
- 1988–1996 *Member of the Technical Staff*

AT&T Bell Laboratories, Murray Hill, NJ, USA

Other Academic Appointments

- 2020– *Strategic Advisory Board*
Schaffhausen Institute of Technology (SIT), Schaffhausen, Switzerland
- 2019–2020 *Chair, Management Board*
Cryptocurrency Strategy, Techniques, and Algorithms (CRYSTAL) Centre, National University of Singapore, Singapore
- 2018–2021 *Deputy Convenor, Computer Science/Information Technology Panel*
2020 Research Assessment Exercise (RAE 2020), University Grants Committee, Hong Kong
- 2017–2019 *External Examiner, Bachelor of Computer Science (Hons) (Software Engineering)*
Faculty of Computing and Informatics, Multimedia University, Malaysia
- 2017–2019 *External Assessor, Bachelor of Software Engineering*
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia
- 2014–2016 *Executive Committee, Services & Digital Economy Research & Development*
National Research Foundation (NRF), Singapore
- 2013–2014 *Computer Science/Information Technology Panel*
2014 Research Assessment Exercise (RAE 2014), University Grants Committee, Hong Kong
- 2015–2016 *Management Board*
Institute of Systems Science (ISS), National University of Singapore, Singapore
- 2015–2016 *Management Board*
Institute for Mathematical Sciences (IMS), National University of Singapore, Singapore
- 2013–2018 *Visiting Professor*
Department of Computer Science, University College London, London, UK
- 2013–2016 *Advisory Committee*
School of Infocomm, Republic Polytechnic, Singapore
- 2013–2016 *Management Board*
Temasek Laboratories, National University of Singapore, Singapore
- 2007–2009 *External Member, Faculty Selection Committee*
Faculty of Informatics, Università della Svizzera Italiana, Lugano, Switzerland
- 2006–2010 *External Examiner, BEng/MEng in Computer Science and Software Engineering*
External Examiner, BSc Computer Science with Business Management
External Examiner, BSc/MEng Electronic and Software Engineering
School of Computer Science, University of Birmingham, UK
- 2005– *Advisory Council*
Dept. of Computer Science & Engineering, Univ. of North Texas, Denton, TX, USA
- 2004–2010 *Honorary Professorial Research Fellow*
Department of Computing, Imperial College London, London, UK
- 2004–2007 *External Examiner, MSc in Software Engineering for the e-Economy*
External Examiner, MSc in Advanced Software Engineering
External Examiner, MSc in Advanced Distributed Systems
Department of Computer Science, University of Leicester, Leicester, UK

Other Industry Appointments

- 2019–2020 *Subcommittee on Smart Nation/Industry Transformation Maps/Emerging Technologies*
Coordinating Committee for Cybersecurity, Enterprise Singapore, Singapore

1999–2001 *Scientific Advisory Board*
HomeGopher, Inc., Tustin, CA, USA

Consulting, Review Committees, and Other Professional Service

2018 *Reviewing Committee*
Computer Science Discipline, Tianjin University, China

2017– *ACM Distinguished Member Committee*
Deputy Chair, 2019
Chair, 2020

2012 *External Reviewer, Faculty Hiring Committee*
Department of Computer Science and Engineering, Aalto University, Jul. 2012

2011–2015 *ACM-IEEE Computer Society Ken Kennedy Award Committee*

2007 *Expert Reviewer, Software Product Lines Research Programme*
Lero—The Irish Software Engineering Research Centre, Univ. of Limerick, Ireland

2006 *Ex-Post Evaluation Panelist on behalf of Technopolis Ltd*
Industry PhD programme, IDA Department, Linköping University, Sweden

2004 *Senior Advisor, Shanghai Software Testing Center, Shanghai, China*

2004 *Expert Witness, individual claim in Brent County Court, London, UK*

2004–2005 *Expert Consultant, Intelligence in Future Imaging Technology (IFIT) Consortium*
Kodak Research, UK

2000–2011 *Co-Lead (with Lori A. Clarke), Software Testing and Analysis Team*
Software Engineering Impact Project

1993–1994 *Expert Witness, US Department of Labor Alien Employment Certification cases*

1990–1991 *Representative from AT&T, Formal Methods Transition Study*
Microelectronics and Computer Technology Corporation (MCC), Austin, TX, USA

Elective Offices

2001–2015 *Past Chair (2012–2015)*
Chair (2009–2012)
Vice Chair (2005–2009)
Member-at-Large (2001–2005)
ACM Special Interest Group on Software Engineering (ACM SIGSOFT)

Editorial Boards

ACM Transactions on Software Engineering and Methodology, 2005–2018
Editor-in-Chief, 2013–2018

IEEE Transactions on Software Engineering, 1998–2002
Co-Editor, Outstanding Papers of ICSE 1999, Vol. 27, No. 2, Feb. 2001
Guest Co-Editor, Best Papers of ISSTA 2007, Vol. 34, No. 5, Sep./Oct. 2008

Software Process Improvement and Practice, 1997–1998

Current Professional Memberships

Association for Computing Machinery (ACM) (Fellow)
Special Interest Group on Data Communication (SIGCOMM)
Special Interest Group on Mobility of Systems, Users, Data and Computing (SIGMOBILE)
Special Interest Group on Software Engineering (SIGSOFT)
Institute of Electrical and Electronics Engineers (IEEE) (Fellow)
UK Computing Research Committee (UKCRC)

Grants and Contracts

George Mason University

Cornell University, Break Through Tech, *Inclusive Computing at Mason: Get Cities Proposal*, subaward 91229-21014. US\$774,663 (Career Access), Feb. 2021–Dec. 2025. Co-PI with PI Huzefa Rangwala, co-PIs Özlem Uzuner and Mark Snyder.

Cornell University, Break Through Tech, *Inclusive Computing at Mason: Get Cities Proposal*, subaward 90620-21012. US\$2,028,235 (Curriculum, Community, and General Administration), Feb. 2021–Dec. 2025. Co-PI with PI Huzefa Rangwala, co-PIs Özlem Uzuner and Mark Snyder.

National University of Singapore

Singapore National Research Foundation (NRF), National Satellite of Excellence in Trustworthy Software Systems (NSoE-TSS) Core Technologies, *Evaluating the Trustworthiness of Deep Learning Systems*, NSOE-TSS2019-05. S\$418,400, Oct. 2019–Sep. 2021. Sole PI.

Singapore National Research Foundation (NRF), *NUS-Singtel Corporate Research Laboratory in Cyber Security*, S\$42,768,245 (funded with S\$14,129,194 each from NRF and Singtel and S\$14,509,857 from NUS), Jul. 2016–Jun. 2021. Lead PI and Lab Director.

National University of Singapore – Humboldt Universität zu Berlin Profile Partnership, *Efficient Probabilistic Verification of Reconfigurable Software System*. S\$23,550.00, Jul. 2016–Jun. 2017. PI for NUS, with Lars Grunske as PI for HU.

Singapore Agency for Science, Technology and Research (A*STAR), Science and Engineering Research Council (SERC), Public Sector Research Funding (PSF). *Ambient Web: Towards Write Once, Run Everywhere Mobile IoT Applications*, 1521200084. S\$648,750.00, Apr. 2016–Mar. 2019. Sole PI.

Singapore Ministry of Education (MOE), Academic Research Fund (AcRF) Tier 2. *Modeling and Analysis of Stochastic Systems with Perturbed Parameters*, MOE2015-T2-1-137. S\$417,705.60, Dec. 2015–Feb. 2019. Sole PI.

Microsoft Research Asia. *Inferring People's Mood and Context: Methods and Framework*, 754074. S\$25,037.00, Apr. 2013–Apr. 2014. Co-PI with PI Stanislaw Jarzabek.

National University of Singapore, School of Computing and Office of the Deputy President for Research & Technology. *Felicitous Computing Institute*, R-252-000-473-133 and R-252-000-473-750. S\$2,000,000.00, Jan. 2012–Dec. 2015. PI and Institute Director.

National University of Singapore, Singapore Ministry of Education Academic Research Fund Tier 1. *Advances in Probabilistic Modeling and Analysis*, R-252-000-458-133, S\$300,000.00, Jun. 2011–Nov. 2014. Sole PI.

University College London

European Office of Aerospace Research and Development (EOARD), US Air Force Office of Scientific Research (AFOSR). *Enhancing the Dependability of Complex Missions through Automated Analysis*, FA8655-10-1-3007, US\$194,295.00, Jan. 2010–Jul. 2013. Sole PI.

UK Engineering and Physical Sciences Research Council (EPSRC). *UbiComp Grand Challenge*, EP/F013442/1, travel grant for Sebastian Elbaum. £1,660, Jun. 2008–Sep. 2008. Sole PI.

UK Engineering and Physical Sciences Research Council (EPSRC). *Improving Medical Safety using Software Engineering Technology*, EP/E062245/1, visiting researcher grant for Leon J. Osterweil and Lori A. Clarke. £71,417.81, Apr. 2007–Aug. 2007. Sole PI.

UK Engineering and Physical Sciences Research Council (EPSRC). *Testing Techniques for Context-Aware Ubiquitous Systems*, EP/E006191/1, visiting researcher grant for Sebastian Elbaum. £54,992.99, Aug. 2006–Dec. 2006. Sole PI.

UK Engineering and Physical Sciences Research Council (EPSRC), Wired and Wireless Intelligent Networked Systems (WINES) Programme. *UbiVal: Fundamental Approaches to Validation of Ubiquitous Computing Applications and Infrastructures*, EP/D077273/1, project at University College London, University of Birmingham, Imperial College London, Cambridge University and University of Oxford. £1,610,781.58 (including £634,475.93 for University College London), Oct. 2006–Sep. 2010. Project Director.

European Commission, Mobile and Wireless Services Beyond 3G Strategic Objective, Information Society Technologies (IST), Sixth Framework Programme (FP6). *PLASTIC: Providing dependable and Adaptive Service Technology for pervasive Information and Communication*, EU-IST-2005-026955, specific targeted research project (STREP) of 11 partners coordinated by Valérie Issarny of INRIA. €2,500,000 (including €285,000 for University College London), Feb. 2006–Jul. 2008. Co-PI.

European Commission, Global Computing II Programme, Future and Emerging Technologies (FET), Information Society Technologies (IST), Sixth Framework Programme (FP6). *SENSORIA: Software Engineering for Service-Oriented Overlay Computers*, EU-IST-3-016004-IP-09, integrated project (IP) of 18 partners coordinated by Martin Wirsing of LMU München. €8,150,000 (including €375,748 for University College London), Sep. 2005–Aug. 2009. Site Leader for UCL, Work Package Leader (WP6, Deployment and Reengineering).

The Royal Society, Wolfson Research Merit Award, £200,000, Apr. 2004–Mar. 2009. Sole PI.

University of California, Irvine

US Defense Advanced Research Projects Agency (DARPA), Information Systems Office (ISO), Program in Dynamic Assembly for System Adaptability, Dependability, and Assurance (DASADA). *Proteus: Assessment and Adaptation through Dynamic Architecture Technology*, F30602-00-2-0607. US\$1,400,000, Jun. 2000–Jun. 2003. Co-PI with PI Richard N. Taylor, co-PIs David F. Redmiles and Adriaan W. van der Hoek.

US Defense Advanced Research Projects Agency (DARPA), Information Technology Office (ITO), Program in Runtime Environments for Networked, Embedded Systems (RENES). *SensOS: Architecture-centric Dynamic Adaptation of Deeply Networked Embedded Systems*, F30602-99-C-0174. US\$980,400 (including US\$443,313 for UC Irvine), Sep. 1999–Jan. 2001. Co-PI with PI Richard N. Taylor.

US Air Force Office of Scientific Research, New World Vistas Program. *Specification and Dynamic Checking of Composition Constraints in Distributed Component-Based Systems*, F49620-98-1-0061. US\$278,767, Nov. 1997–Oct. 2001. Sole PI.

US National Science Foundation, Software Engineering & Languages Program, Faculty Early Career Development (CAREER) Award. *CAREER: Mechanisms for Ensuring the Integrity of Distributed Object Systems*, CCR-9701973. US\$200,000, Jul. 1997–Jun. 2001. Sole PI.

University of California MICRO Program, *Specifying and Checking Integrity Constraints in Distributed Avionics Systems*, 97-151/UCM-110-720842AK. US\$7,959, Jul. 1997–Dec. 1998. Co-sponsored by Northrop Grumman Corporation. Sole PI.

University of California, Irvine, Department of Information and Computer Science, Research & Travel Award, US\$2,623, 1998. Sole PI.

University of California, Irvine, Department of Information and Computer Science, Research & Travel Award, US\$2,300, 1997. Sole PI.

Patents

- P11. Y. Huang, A.W.P. Fung, D.S. Rosenblum, S. Jainik, R. Teodorescu, T.W. Chen, C.-M. Lin, C.Y. Wang, P.F. Yang and R.C. Leng, *Packing routing via payload inspection for alert services, for digital content delivery and for quality of service management and caching with selective multicasting in a publish-subscribe network*, Korea Patent, registration no. 1009852370000, Sep. 28, 2010.
- P10. P.F. Yang, T.W. Chen, A.W.P. Fung, D.S. Rosenblum, S. Jainik, C.Y. Wang, C.M. Lin, Y. Huang and R. Teodorescu, *Method and apparatus for reliable and efficient content-based routing and query and response in a publish-subscribe network*, Korea Patent, registration no. 1009715060000, Jul. 14, 2010.
- P9. T.-W. Chen, A.W.P. Fung, D.S. Rosenblum and P.-F. Yang, *Method and apparatus for content-based packet routing using compact filter storage and off-line pre-computation*, US Patent 7,653,753, Jan. 26, 2010.
- P8. D.S. Rosenblum and R. Teodorescu, *Method and apparatus for implementing query-response interactions in a publish-subscribe network*, US Patent 7,627,603, Dec. 1, 2009.
- P7. T.-W. Chen, A.W.P. Fung, D.S. Rosenblum and P.-F. Yang, *Method and apparatus for propagating content filters for a publish-subscribe network*, US Patent 7,551,629, Jun. 23, 2009.

- P6. P.-F. Yang, T.-W. Chen, A.W.P. Fung, D.S. Rosenblum, S. Yajnik, C.-Y. Wang, C.-M. Lin, Y. Huang and R. Teodorescu, *Method and apparatus for reliable and efficient content-based routing and query and response in a publish-subscribe network*, China Patent, publication no. 1656474, Feb. 4, 2009.
- P5. D.S. Rosenblum, *Method for sending and receiving a Boolean function over a network*, US Patent 7,117,270, Oct. 3, 2006.
- P4. T.-W. Chen, A.W. Fung, P.-F. Yang, Y. Huang, C.-M. Lin, S. Yajnik, C.-Y. Wang and D.S. Rosenblum, *Packet routing via payload inspection and subscription processing in a publish-subscribe network*, Hong Kong Patent 1082858, Jun. 16, 2006.
- P3. D.S. Rosenblum, *Method for storing Boolean functions to enable evaluation, modification, reuse, and delivery over a network*, US Patent 6,910,033, Jun. 21, 2005.
- P2. T.-W. Chen, A.W. Fung, P.-F. Yang, Y. Huang, C.-M. Lin, S. Yajnik, C.-Y. Wang and D.S. Rosenblum, *Packet routing via payload inspection and subscription processing in a publish-subscribe network*, Taiwan Patent 571531, Jan. 11, 2004.
- P1. Y.F.R. Chen, D.S. Rosenblum and K.-P. Vo, *System and method for selecting test units to be re-run in software regression testing*, US Patent 5,673,387, Sep. 30, 1997.

Publications

Edited Books

- EB4. D. Bianculli, N. Medvidović and D.S. Rosenblum (editors), *40 Editions of ICSE—the ruby anniversary celebration*, limited edition book distributed at the 40th Int'l Conf. Software Engineering, Gothenburg, Sweden, May–Jun. 2018. Available online at <https://www.icse2018.org/info/50-years-of-se> (last accessed Jun. 3, 2018).
- EB3. D.S. Rosenblum and G. Taentzer (co-editors), *Proc. 13th Int'l Conf. Fundamental Approaches to Software Engineering* (FASE 2010), Springer LNCS 6013, Mar. 2010.
- EB2. J. Estublier and D.S. Rosenblum (co-editors), *Proc. 26th Int'l Conf. Software Engineering* (ICSE 2004), IEEE CS Press, May 2004.
- EB1. D.S. Rosenblum (editor), *Proc. ACM SIGSOFT 2000 Eighth Int'l Symp. Foundations of Software Engineering* (FSE 2000), ACM Press, Nov. 2000.

Chapters in Edited Books

- BC8. A. Mukhija, D.S. Rosenblum, H. Foster and S. Uchitel, “Runtime Support for Dynamic and Adaptive Service Composition”, in Part VII of *Rigorous Software Engineering for Service-Oriented Systems: Results of the SENSORIA Project on Software Engineering for Service-Oriented Computing*, Springer LNCS 6582, 2011, pp. 585–603.
- BC7. H. Foster, A. Mukhija, D.S. Rosenblum and S. Uchitel, “Specification and Analysis of Dynamically-Reconfigurable Service Architectures”, in Part V of *Rigorous Software Engineering for Service-Oriented Systems: Results of the SENSORIA Project on Software Engineering for Service-Oriented Computing*, Springer LNCS 6582, 2011, pp. 428–446.
- BC6. A. Carzaniga, D.S. Rosenblum and A.L. Wolf, “Design and Evaluation of a Wide-Area Event Notification Service”, in *Foundations of Intrusion Tolerant Systems*, DARPA Program on Organically Assured and Survivable Information Systems (OASIS), IEEE Computer Society, 2003, pp. 283–334. Invited reprint of #J16.
- BC5. A.L. Wolf and D.S. Rosenblum, “A Study in Software Process Data Capture and Analysis”, in *Process-Centered Software Engineering Environments*, IEEE Computer Society Press, 1995, pp. 361–370. Invited reprint of #C9.
- BC4. Y.-F. Chen, G.S. Fowler, D.G. Korn, E. Koutsosifos, S.C. North, D.S. Rosenblum and K.-P. Vo, “Intertool Connections”, Chapter 11 of *Practical Reusable UNIX Software*, Wiley, 1995, pp. 299–336.
- BC3. D.S. Rosenblum and B. Krishnamurthy, “Generalized Event-Action Handling”, Chapter 9 of *Practical Reusable UNIX Software*, Wiley, 1995, pp. 247–274.
- BC2. D.S. Rosenblum, “Self-Checking Programs and Program Instrumentation”, Chapter 5 of *Practical Reusable UNIX Software*, B. Krishnamurthy (ed.), Wiley, 1995, pp. 159–176.

- BC1. D.S. Rosenblum, "A Methodology for the Design of Ada Transformation Tools in a DIANA Environment", in M.V. Zelkowitz (ed.), *Reprints in Software*, third ed., IEEE Computer Society Press, 1987, pp. 84–93. Invited reprint of paper #J1.

Refereed Journal Papers

- J31. K. Ouyang, Y. Liang, Y. Liu, Z. Tong, S. Ruan, Y. Zheng and D.S. Rosenblum, "Fine-Grained Urban Flow Inference", *IEEE Trans. Knowledge and Data Engineering*, 2020, 16pp. to appear.
- J30. G. Su, L. Liu, M. Zhang and D.S. Rosenblum, "Quantitative Verification for Monitoring Event-Streaming Systems", *IEEE Trans. Software Engineering*, 2020, 13pp., to appear.
- J29. Y. Liu, Y. Liang, K. Ouyang, S. Liu, D.S. Rosenblum and Y. Zheng, "Predicting Urban Water Quality with Ubiquitous Data—A Data-driven Approach", *IEEE Trans. Big Data*, 2020, 14pp., to appear.
- J28. H. Li, Y. Liu, N. Mamoulis and D.S. Rosenblum, "Translation-Based Sequential Recommendation for Complex Users on Sparse Data", *IEEE Trans. Knowledge and Data Engineering*, Vol. 32, No. 8, Aug. 2020, pp. 1639–1651.
- J27. O. Bataineh, D.S. Rosenblum and M. Reynolds, "Efficient Decentralized LTL Monitoring Framework using Tableau Technique", *ACM Trans. Embedded Computing Sys.*, Special Issue for ESWEEK 2019, Vol. 18, No. 5S, Oct. 2019, Article 87, 12pp. Presented at EMSOFT 2019, New York, NY, USA, Oct. 2019.
- J26. L. Liu, S. Wang, B. Hu, Q. Qiong, J. Wen and D.S. Rosenblum, "Learning Structures of Interval-Based Bayesian Networks in Probabilistic Generative Model for Human Complex Activity Recognition", *Pattern Recognition*, Vol. 81, Sep. 2018, pp. 545–561.
- J25. G. Su, Y. Feng, T. Chen and D.S. Rosenblum, "Asymptotic Perturbation Bounds for Probabilistic Model Checking with Empirically Determined Probability Parameters", *IEEE Trans. Software Engineering*, Vol. 42, No. 7, Jul. 2016, pp. 623–639.
- J24. Y. Liu, L. Nie, L. Liu and D.S. Rosenblum, "From Action to Activity: Sensor-Based Activity Recognition", *Neurocomputing*, Vol. 181, 12 Mar. 2016, pp. 108–115.
- J23. J. Lü, D.S. Rosenblum, T. Bultan, V. Issarny, S. Dustdar, M.-A. Storey and D. Zhang, "Roundtable: The Future of Software Engineering for Internet Computing", *IEEE Software*, invited contribution for Focus Section on Internetware and Beyond, Vol. 32, No. 1, Jan./Feb. 2015, pp. 91–97.
- J22. L. Duboc, E. Letier and D.S. Rosenblum, "Systematic Elaboration of Scalability Requirements through Goal-Obstacle Analysis", *IEEE Trans. Software Engineering*, Vol. 39, No.1, Jan. 2013, pp. 119–140.
- J21. M. Sama, S. Elbaum, F. Raimondi, D.S. Rosenblum and Z. Wang, "Context-Aware Adaptive Applications: Fault Patterns and Their Automated Identification", invited paper for Special Issue on the Best Papers of FSE 2008, *IEEE Trans. Software Engineering*, Vol. 36, No. 5, Sep./Oct. 2010, pp. 644–661.
- J20. M. Sama, D.S. Rosenblum, Z. Wang and S. Elbaum, "Multi-Layer Faults in the Architectures of Mobile, Context-Aware Adaptive Applications", invited paper for Special Issue on Software Architecture and Mobility, *Journal of Systems and Software*, Vol. 83, Issue 6, Jun. 2010, pp. 906–914.
- J19. L.A. Clarke and D.S. Rosenblum, "Runtime Assertion Checking", Sidebar in "Determining the Impact of Software Engineering Research on Practice", *Computer*, Vol. 41, No. 3, Mar. 2008, p.48.
- J18. A. Orso, H. Do, G. Rothermel, M.J. Harrold and D.S. Rosenblum, "Using Component Metadata to Regression Test Component-Based Software", *Software Testing, Verification & Reliability*, Vol. 17, No. 2, Jun. 2007, pp. 61–94.
- J17. N. Medvidovic, D.S. Rosenblum, J.E. Robbins and D.F. Redmiles, "Modeling Software Architectures in the Unified Modeling Language", *ACM Trans. Software Engineering and Methodology*, Vol. 11, No. 1, Jan. 2002, pp. 2–57.
- J16. A. Carzaniga, D.S. Rosenblum and A.L. Wolf, "Design and Evaluation of a Wide-Area Event Notification Service", *ACM Trans. Computer Systems*, Vol. 19, No. 3, Aug. 2001, pp. 332–383.
- J15. J. Bible, G. Rothermel and D.S. Rosenblum, "A Comparative Study of Coarse- and Fine-Grained Safe Regression Test Selection Techniques", *ACM Trans. Software Engineering and Methodology*, Vol. 10, No. 2, Apr. 2001, pp. 149–183.
- J14. M.J. Harrold, D. Rosenblum, G. Rothermel and E. Weyuker, "Empirical Studies of a Prediction Model for Regression Test Selection", *IEEE Trans. Software Engineering*, Vol. 27, No. 3, Mar. 2001, pp. 248–263.

- J13. R. Natarajan and D.S. Rosenblum, "Supporting Architectural Concerns in Component Interoperability Standards", *IEEE Proceedings – Software*, Special Issue on Component-based Software Engineering, Vol. 147, No. 6, Dec. 2000, pp. 215–223.
- J12. P. Oreizy, M.M. Gorlick, R.N. Taylor, D. Heimbigner, G. Johnson, N. Medvidovic, A. Quilici, D.S. Rosenblum and A.L. Wolf, "An Architecture-based Approach to Self-Adaptive Software", *IEEE Intelligent Systems*, Vol. 14, No. 3, May / Jun. 1999, pp. 54–62.
- J11. D.S. Rosenblum and E.J. Weyuker, "Using Coverage Information to Predict the Cost-Effectiveness of Regression Testing Strategies", *IEEE Trans. Software Engineering*, Vol. 23, No. 3, Mar. 1997, pp. 146–156.
- J10. D.S. Rosenblum and E.J. Weyuker, "Lessons Learned from a Regression Testing Case Study", *Empirical Software Engineering*, Vol. 2, No. 2, Jun. 1997, pp. 188–191.
- J9. P.T. Devanbu, D.S. Rosenblum, A.L. Wolf, "Generating Testing and Analysis Tools with Aria", *ACM Trans. Software Engineering and Methodology*, Vol. 5, No. 1, Jan. 1996, pp. 42–62.
- J8. B. Krishnamurthy and D.S. Rosenblum, "Yeast: A General Purpose Event-Action System", *IEEE Trans. Software Engineering*, Vol. 21, No. 10, Oct. 1995, pp. 845–857.
- J7. N.S. Barghouti, D.S. Rosenblum, D.G. Belanger and C. Alliegro, "Two Case Studies in Modeling Real, Corporate Processes", *Software Process Improvement and Practice*, Vol. 1, Pilot Issue, Aug. 1995, pp. 17–32.
- J6. D.S. Rosenblum, "A Practical Approach to Programming with Assertions", *IEEE Trans. Software Engineering*, Vol. 21, No. 1, Jan. 1995, pp. 19–31. Minor correction published in Vol. 21, No. 3, Mar. 1995, p. 265.
- J5. D.S. Rosenblum, "Specifying Concurrent Systems with TSL", *IEEE Software*, Vol. 8, No. 3, May 1991, pp. 52–61.
- J4. S. Sankar and D.S. Rosenblum, "Runtime Checking and Debugging of Formally Specified Programs", Technical Correspondence, *ACM Computing Surveys*, Vol. 23, No. 1, Mar. 1991, pp. 125–127.
- J3. D.C. Luckham, R.B. Neff and D.S. Rosenblum, "An Environment for Ada Software Development Based on Formal Specification", *ACM SIGAda Ada Letters*, Vol. VII, No. 3, May–Jun. 1987, pp. 94–106.
- J2. D.S. Rosenblum, "An Efficient Communication Kernel for Distributed Ada Runtime Tasking Supervisors", *ACM SIGAda Ada Letters*, Vol. VII, No. 2, Mar.–Apr. 1987, pp. 102–117. **Winner of 1986 ACM SIGAda Best Graduate Student Paper Award.**
- J1. D.S. Rosenblum, "A Methodology for the Design of Ada Transformation Tools in a DIANA Environment", *IEEE Software*, Vol. 2, No. 2, Mar. 1985, pp. 24–33.

Other Journal Papers

- JO39. D.S. Rosenblum, "Farewell Editorial from the Outgoing Editor-in-Chief", *ACM Trans. Software Engineering and Methodology*, Vol. 28, No. 1, Feb. 2019, Article 1e, 2pp.
- JO38. D.S. Rosenblum, "Editorial", *ACM Trans. Software Engineering and Methodology*, Vol. 27, No. 1, Sep. 2018, Article 9, 2pp.
- JO37. D.S. Rosenblum, "Editorial", *ACM Trans. Software Engineering and Methodology*, Vol. 27, No. 1, Jun. 2018, Article 1e, 2pp.
- JO36. D.S. Rosenblum, "Editorial", *ACM Trans. Software Engineering and Methodology*, Vol. 25, No. 2, May 2016, Article 11, 2pp.
- JO35. M.B. Dwyer and D.S. Rosenblum, "Editorial: Journal-First Publication for the Software Engineering Community", *ACM Trans. Software Engineering and Methodology*, Vol. 25, No.1, Nov. 2015, Article 1, 2pp. Also appeared in *IEEE Trans. Software Engineering*, Vol. 42, No. 1, Jan. 2016, p.1.
- JO34. D.S. Rosenblum, "The Pros and Cons of the 'PACM' Proposal: Counterpoint", *Communications of the ACM*, Vol. 58, No. 9, Sep. 2015, pp. 44–45.
- JO33. D.S. Rosenblum, "Editorial", *ACM Trans. Software Engineering and Methodology*, Vol. 23, No. 4, Aug. 2014, Article 27, 1pp.
- JO32. D.S. Rosenblum, "Editorial", *ACM Trans. Software Engineering and Methodology*, Vol. 23, No. 2, Mar. 2014, Article 11, 4pp.

- JO31. D.S. Rosenblum, "Editorial", *ACM Trans. Software Engineering and Methodology*, Vol. 23, No. 1, Feb. 2014, Article 1, 2pp.
- JO30. G.A. Lewis, J. Gray, H. Muccini, N. Nagappan, D. Rosenblum and E. Shihab, "Report of the 2013 ICSE 1st International Workshop on Engineering Mobile-Enabled Systems (MOBS 2013)", *ACM SIGSOFT Software Engineering Notes*, Vol. 38, No. 5, Sep. 2013, pp. 55–58.
- JO29. D.S. Rosenblum, "Editorial—In Memoriam: David Notkin (1955–2013)", *ACM Trans. Software Engineering and Methodology*, Vol. 22, No. 3, Jul. 2013, Article 16, 2pp.
- JO28. D.S. Rosenblum, "Editorial—Looking Forward", *ACM Trans. Software Engineering and Methodology*, Vol. 22, No. 1, Feb. 2013, Article 2, 3pp.
- JO27. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 37, No. 4, Jul. 2012, p. 1.
- JO26. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 37, No. 3, May 2012, p. 1.
- JO25. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 37, No. 2, Mar. 2012, p. 1.
- JO24. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 37, No. 1, Jan. 2012, p. 1.
- JO23. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 36, No. 6, Nov. 2011, p. 1.
- JO22. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 36, No. 5, Sep. 2011, p. 1.
- JO21. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 36, No. 4, Jul. 2011, p. 1.
- JO20. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 36, No. 3, May 2011, p. 1.
- JO19. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 36, No. 1, Jan. 2011, p. 1.
- JO18. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 35, No. 5, Sep. 2010, p. 1.
- JO17. L. Duboc, D.S. Rosenblum and E. Letier, "Death, Taxes, & Scalability", invited column for Requirements Department, *IEEE Software*, Vol. 27, No. 4, Jul. / Aug. 2010, pp. 20–21.
- JO16. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 35, No. 4, Jul. 2010, p. 1.
- JO15. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 35, No. 3, May 2010, p. 1.
- JO14. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 35, No. 1, Jan. 2010, p. 1.
- JO13. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 34, No. 6, Nov. 2009, p. 1.
- JO12. D.S. Rosenblum, "Letter from the Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 34, No. 5, Sep. 2009, p. 1.
- JO11. S. Elbaum and D.S. Rosenblum, "Guest Editors' Introduction: [2007] International Symposium on Software Testing and Analysis", *IEEE Trans. Software Engineering*, Vol. 34, No. 5, Sep./Oct. 2008, pp. 577–578.
- JO10. D.S. Rosenblum, "Letter from the Vice-Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 33, No. 5, Sep. 2008, p. 1.
- JO9. L.A. Clarke and D.S. Rosenblum, "A Historical Perspective on Runtime Assertion Checking in Software Development", *ACM SIGSOFT Software Engineering Notes*, Vol. 31, No. 3, May 2006, pp. 25–37.

- JO8. D.S. Rosenblum, "Challenges in Software Testing", Guest Editorial in *System Design Frontier*, Vol. 3, No. 4, Apr. 2006, pp. 5–6. Published online at <http://www.hwswworld.com/pdfs/frontier26.pdf.gz> (last accessed Mar. 28, 2006).
- JO7. D.S. Rosenblum, "Letter from the Vice-Chair", *ACM SIGSOFT Software Engineering Notes*, Vol. 30, No. 6, Nov. 2005, pp. 1–2.
- JO6. D. Rosenblum, "Technology Update: Real-Time Delivery Gets Efficiency Boost", invited contribution to *Network World*, Vol. 20, No. 19, May 12, 2003, p. 37. Also published online in *Network World Fusion*, <http://www.nwfusion.com/news/tech/2003/0512techupdate.html> (last accessed Aug. 2, 2010).
- JO5. J. Kramer, D. Garlan and D.S. Rosenblum, "Guest Editors' Introduction: 1999 Int'l Conference on Software Engineering", *IEEE Trans. Software Engineering*, Vol. 27, No. 2, Feb. 2001, pp. 97–98.
- JO4. D.S. Rosenblum, "CAREER: Mechanisms for Ensuring the Integrity of Distributed Object Systems", NSF Software Engineering and Language Program Summaries, *ACM SIGSOFT Software Engineering Notes*, Vol. 25, No. 1, Jan. 2000, pp. 78–79.
- JO3. G.M. Karam and D.S. Rosenblum, "Succeedings of the 8th Int'l Workshop on Software Specification and Design: Concurrency and Distribution Group Report", *ACM SIGSOFT Software Engineering Notes*, Vol. 21, No. 5, Sep. 1996, pp. 29–31.
- JO2. D.S. Rosenblum, "Formal Methods and Testing: Why the State-of-the-Art Is Not the State-of-the-Practice (ISSTA '96/FMSP '96 Panel Summary)", *ACM SIGSOFT Software Engineering Notes*, Vol. 21, No. 4, Jul. 1996, pp. 64–66.
- JO1. N.S. Barghouti and D.S. Rosenblum, "Modeling Software and Business Processes in Large Organizations—Experience Report", invited paper, *IEEE Technical Council on Software Engineering Software Process Newsletter*, No. 4, Fall 1995, pp. 5–7.

Refereed Conference and Workshop Papers

- C102. Y. Xiao, I. Beschastnikh, D.S. Rosenblum, C. Sun, S. Elbaum, Y. Lin and J.S. Dong, "Self-Checking Deep Neural Networks in Deployment", *Proc. 43rd Int'l Conf. Software Engineering (ICSE 2021)*, virtual conference, May 2021, pp. 372–384.
- C101. T. Thanthriwatta and D.S. Rosenblum, "Instance Selection for Online Updating in Dynamic Recommender Environments", *Proc. 25th Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD 2021)*, May 2021, Delhi, India, pp. 612–624.
- C100. Y. Liang, K. Ouyang, J. Sun, Y. Wang, J. Zhang, Y. Zheng, D. Rosenblum and R. Zimmermann, "Fine-Grained Urban Flow Prediction", *Proc. 30th The Web Conf. (The Web Conference 2021)*, Apr. 2021, Ljubljana, Slovenia, pp. 1833–1845.
- C99. Z. Tong, Y. Liang, C. Sun, X. Li, D.S. Rosenblum and A. Lim, "Digraph Inception Convolutional Networks", *Proc. 34th Conf. Neural Information Processing Systems (NeurIPS 2020)*, virtual conference, Dec. 2020, 12pp., to appear.
- C98. T. Thanthriwatta and D.S. Rosenblum, "Improving Dynamic Recommendation using Network Embedding for Context Inference", Short Paper, *Proc. 32nd Int'l Conf. Tools with Artificial Intelligence (ICTAI 2020)*, virtual conference, Nov. 2020, pp. 205–210.
- C97. Y. Liang, K. Ouyang, Y. Wang, Y. Liu, J. Zhang, Y. Zheng and D.S. Rosenblum, "Revisiting Convolutional Neural Networks for Urban Flow Analytics", *Proc. 2020 European Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2020)*, virtual conference, Sep. 2020, pp. 578–594.
- C96. K. Ouyang, Y. Liang, W. Yang, Y. Liu and D.S. Rosenblum, "Unsupervised Learning of Disentangled Location Embeddings", *Proc. 2020 Int'l Joint Conf. Neural Networks (IJCNN 2020)*, Glasgow, Scotland, UK, Jul. 2020, 8pp., pp. 1–8.
- C95. N. Wanigasekara, Y. Liang, S.T. Goh, Y. Liu, J.J. Williams and D.S. Rosenblum, "Learning Multi-Objective Rewards and User Utility Function in Contextual Bandits for Personalized Ranking", *Proc. 28th Int'l Joint Conf. Artificial Intelligence (IJCAI 2019)*, Macao, China, Aug. 2019, pp. 3835–3841.
- C94. Y. Liang, K. Ouyang, L. Jing, S. Ruan, Y. Liu, J. Zhang, D.S. Rosenblum and Y. Zheng, "UrbanFM: Inferring Fine-Grained Urban Flows", Applied Data Science Track, *Proc. 25th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD 2019)*, Anchorage, AK, USA, Aug. 2019, pp. 3132–3142.

- C93. O. Bataineh, D.S. Rosenblum and M. Reynolds, “A Novel Decentralized LTL Monitoring Framework Using Formula Progression Table”, *Proc. 26th Int’l Symp. Model Checking of Software (SPIN 2019)*, Beijing, China, Jul. 2019, pp. 38–55.
- C92. Y. Liu, H. Li, A. Garcia Duran, M. Niepert, D. Oñoro-Rubio and D.S. Rosenblum, “MMKG: Multi-Modal Knowledge Graphs”, Resources Track, *Proc. 16th Extended Semantic Web Conf. (ESWC 2019)*, Springer LNCS 11503, Portorož, Slovenia, Jun. 2019, pp. 459–474.
- C91. D. Shriver, S. Elbaum, M.B. Dwyer and D.S. Rosenblum, “Evaluating Recommender System Stability with Influence-Guided Fuzzing”, *Proc. 33rd AAAI Conf. Artificial Intelligence (AAAI 2019)*, Honolulu, HI, USA, Feb. 2019, pp. 4934–4942.
- C90. Y.R. Serrano Llerena, M. Böhme, M. Brünink, G. Su and D.S. Rosenblum, “Verifying the Long-Run Behavior of Probabilistic System Models in the Presence of Uncertainty”, *Proc. 26th ACM Joint European Software Engineering Conf. and Symp. Foundations of Software Engineering (ESEC/FSE 2018)*, Orlando, FL, USA, Nov. 2018, pp. 587–597.
- C89. M. Grechanik, C.W. Mao, A. Baisal, B.M.M. Hossain and D.S. Rosenblum, “Differencing Graphical User Interfaces”, *Proc. 18th IEEE Int’l Conf. Software Quality, Reliability and Security (QRS 2018)*, Lisbon, Portugal, Jul. 2018, pp. 203–214.
- C88. K. Ouyang, R. Shokri, D.S. Rosenblum and W. Yang, “A Non-Parametric Generative Model for Human Trajectories”, *Proc. 27th Int’l Joint Conf. Artificial Intelligence and 23rd European Conf. Artificial Intelligence (IJCAI-ECAI 2018)*, Stockholm, Sweden, Jul. 2018, pp. 3812–3817.
- C87. O. Al-Bataineh, M. Reynolds and D. Rosenblum, “A Comparative Study of Decision Diagrams for Real-time Model Checking”, *Proc. 25th Int’l Symp. Model Checking of Software (SPIN 2018)*, Springer LNCS 10869, Málaga, Spain, Jun. 2018, pp. 216–234.
- C86. M. Brünink and D.S. Rosenblum, “Using Branch Frequency Spectra to Evaluate Operational Coverage”, *Proc. 24th Asia-Pacific Software Eng. Conf. (APSEC 2017)*, Nanjing, China, Dec. 2017, pp. 150–159.
- C85. Y. Serrano, G. Su and D.S. Rosenblum, “Probabilistic Model Checking of Perturbed MDPs with Applications to Cloud Computing”, *Proc. 11th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symp. Foundations of Software Engineering (ESEC/FSE 2017)*, Paderborn, Germany, Sep. 2017, pp. 454–464.
- C84. G. Su, T. Chen, Y. Feng and D.S. Rosenblum, “ProEva: Runtime Proactive Performance Evaluation Based on Continuous-Time Markov Chains”, *Proc. 39th Int’l Conf. Software Engineering (ICSE 2017)*, Buenos Aires, Argentina, May 2017, pp. 484–495.
- C83. M. Brünink and D.S. Rosenblum, “Mining Performance Specifications”, *Proc. 24th ACM SIGSOFT Int’l Symp. Foundations of Software Engineering (FSE 2016)*, Seattle, WA, USA, Nov. 2016, pp. 39–49.
- C82. N. Wanigasekara, J. Schmalfuss, D. Carlson and D.S. Rosenblum, “A Bandit Approach for Intelligent IoT Service Composition across Heterogeneous Smart Spaces”, *Proc. 6th Int’l Conf. Internet of Things (IoT 2016)*, Stuttgart, Germany, Nov. 2016, pp. 121–129.
- C81. Y. Liu, Y. Zheng, Y. Liang, S. Liu and D.S. Rosenblum, “Urban Water Quality Prediction Based on Multi-Task Multi-View Learning”, *Proc. 25th Int’l Joint Conf. Artificial Intelligence (IJCAI 2016)*, New York, NY, USA, Jul. 2016, pp. 2576–2582.
- C80. G. Su, D.S. Rosenblum and G. Tamburrelli, “Reliability of Run-Time Quality-of-Service Evaluation using Parametric Model Checking”, *Proc. 38th Int’l Conf. Software Engineering (ICSE 2016)*, Austin, TX, USA, May 2016, pp. 73–84.
- C79. G. Su, T. Chen, Y. Feng, D.S. Rosenblum and P.S. Thiagarajan, “An Iterative Decision-Making Scheme for Markov Decision Processes and Its Application to Self-Adaptive Systems”, *Proc. ETAPS 2016 19th Int’l Conf. Fundamental Approaches to Software Engineering (FASE 2016)*, Springer LNCS 9633, Eindhoven, The Netherlands, Apr. 2016, pp. 269–286.
- C78. L. Liu, L. Cheng, Y. Liu, Y. Jia and D.S. Rosenblum, “Recognizing Complex Activities by a Probabilistic Interval-based Model”, *Proc. 30th AAAI Conf. Artificial Intelligence (AAAI 2016)*, Phoenix, AZ, USA, Feb. 2016, pp. 1266–1272.
- C77. Y. Liu, L. Zhang, L. Nie, Y. Yan and D.S. Rosenblum, “Fortune Teller: Predicting Your Career Path”, *Proc. 30th AAAI Conf. Artificial Intelligence (AAAI 2016)*, Phoenix, AZ, USA, Feb. 2016, pp. 201–207.

- C76. Y. Jia, X. Song, J. Zhou, L. Liu, L. Nie and D.S. Rosenblum, "Fusing Social Networks with Deep Learning for Volunteerism Tendency Prediction", *Proc. 30th AAAI Conf. Artificial Intelligence* (AAAI 2016), Phoenix, AZ, USA, Feb. 2016, pp. 165–171.
- C75. D. Carlson, M. Mögerle, M. Pagel, S. Verma and D.S. Rosenblum, "A Visual Design Toolset for Drag-and-drop Smart Space Configuration", *Proc. 5th Int'l Conf. Internet of Things* (IoT 2015), Demo Track, Seoul, Korea, Oct. 2015, pp. 193–194. **Winner of Best Demo Award.**
- C74. D. Carlson, M. Mögerle, M. Pagel, S. Verma and D.S. Rosenblum, "Ambient Flow: A Visual Approach for Remixing the Internet of Things", *Proc. 5th Int'l Conf. Internet of Things* (IoT 2015), Seoul, Korea, Oct. 2015, pp. 114–121.
- C73. Y. Liu, L. Nie, L. Han, L. Zhang and D.S. Rosenblum, "Action2Activity: Recognizing Complex Activities from Sensor Data", *Proc. 24th Int'l Joint Conf. Artificial Intelligence* (IJCAI 2015), Buenos Aires, Argentina, Jul. 2015, pp. 1617–1623.
- C72. S. Elbaum and D.S. Rosenblum, "Known Unknowns: Testing in the Presence of Uncertainty", *Proc. 22nd ACM SIGSOFT Int'l Symp. Foundations of Software Engineering* (FSE 2014), Visions and Challenges Track, Hong Kong, China, Nov. 2014, pp. 833–836. **Winner of 2nd Prize in the Visions and Challenges Track.**
- C71. G. Su and D.S. Rosenblum, "Nested Reachability Approximation for Discrete-Time Markov Chains with Univariate Parameters", *Proc. 12th Int'l Symp. Automated Technology for Verification and Analysis* (ATVA 2014), Springer LNCS 8837, Sydney, Australia, Nov. 2014, pp. 364–379.
- C70. T. Chen, Y. Feng, D.S. Rosenblum and G. Su, "Perturbation Analysis in Verification of Discrete-Time Markov Chains", *Proc. 25th Int'l Conf. Concurrency Theory* (CONCUR 2014), Springer LNCS 8704, Rome, Italy, Sep. 2014, pp. 218–233.
- C69. G. Su and D.S. Rosenblum, "Perturbation Analysis of Stochastic Systems with Empirical Distribution Parameters", *Proc. 36th Int'l Conf. Software Engineering* (ICSE 2014), Hyderabad, India, Jun. 2014, pp. 311–321.
- C68. G. Su and D.S. Rosenblum, "Asymptotic Bounds for Quantitative Verification of Perturbed Probabilistic Systems", *Formal Methods and Software Engineering: Proc. 15th Int'l Conf. Formal Engineering Methods* (ICFEM 2013), Springer LNCS 8144, Queenstown, New Zealand, Oct.–Nov. 2013, pp. 297–312.
- C67. F. Zervoudakis, D.S. Rosenblum, S. Elbaum and A. Finkelstein, "Cascading Verification: An Integrated Method for Domain-Specific Model Checking", *Proc. 9th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symp. Foundations of Software Engineering* (ESEC/FSE 2013), St. Petersburg, Russia, Aug. 2013, pp. 400–410.
- C66. X. Wang, D.S. Rosenblum and Y. Wang, "A Daily, Activity-Aware, Mobile Music Recommender System", Technical Demo Paper, *Proc. ACM Multimedia 2012* (ACMMM 2012), Nara, Japan, Oct.–Nov. 2012, pp. 1313–1314.
- C65. X. Wang, D.S. Rosenblum and Y. Wang, "Context-Aware Mobile Music Recommendation for Daily Activities", Full Paper, *Proc. ACM Multimedia 2012* (ACMMM 2012), Nara, Japan, Oct.–Nov. 2012, pp. 91–108.
- C64. C. Lucas, S. Elbaum and D.S. Rosenblum, "Detecting Problematic Message Sequences and Frequencies in Distributed Systems", *Proc. 27th ACM SIGPLAN Conf. Object-Oriented Programming, Systems, Languages and Applications* (SPLASH/OOPSLA 2012), Tucson, AZ, USA, Oct. 2012, pp. 915–926.
- C63. P. Papakos, L. Capra and D.S. Rosenblum, "VOLARE: Context-Aware Adaptive Cloud Service Discovery for Mobile Systems", *Proc. 9th Int'l Workshop on Adaptive and Reflective Middleware* (ARM 2010), Bangalore, India, Nov. 2010, pp. 32–38.
- C62. H. Foster, A. Mukhija, D.S. Rosenblum and S. Uchitel, "Engage: Engineering Service Modes with WS-Engineer and Dino", Demo Track, *Proc. 7th Int'l Joint Conf. Service Oriented Computing* (ICSOC/ServiceWave 2009), Stockholm, Sweden, Nov. 2009, pp. 641–642.
- C61. J. Cubo, F. Raimondi, M. Sama and D. Rosenblum, "A Model to Design and Verify Context-Aware Adaptive Service Composition", *Proc. IEEE Int'l Conf. Services Computing* (SCC 2009), Springer LNCS 5900, Bangalore, India, Sep. 2009, pp. 184–191.
- C60. C. Raiciu, F. Huici, M. Handley and D.S. Rosenblum, "ROAR: Increasing the Flexibility and Performance of Distributed Search", *Proc. ACM SIGCOMM 2009 Conf. Data Communication* (SIGCOMM 2009), Barcelona, Spain, Aug. 2009, pp. 291–302.

- C59. P. Papakos, D.S. Rosenblum, A. Mukhija and L. Capra, "VOLARE: Adaptive Web Service Discovery Middleware for Mobile Systems", *Proc. 2nd Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS 2009)*, Vol. 19 of the *Electronic Communications of the EASST*, Lisbon, Portugal, Jun. 2009, pp. 35–40.
- C58. H. Foster, A. Mukhija, D.S. Rosenblum and S. Uchitel, "A Model-Driven Approach to Dynamic and Adaptive Service Brokering using Modes", Short Research Paper, *Proc. 6th Int'l Conf. Service Oriented Computing (ICSOC 2008)*, Sydney, Australia, Dec. 2008, pp. 558–564.
- C57. M. Sama, D.S. Rosenblum, Z. Wang and S. Elbaum, "Model-Based Fault Detection in Context-Aware Adaptive Applications", *Proc. 16th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (FSE 2008)*, Atlanta, GA, USA, Nov. 2008, pp. 261–271.
- C56. M. Sama, F. Raimondi, D. Rosenblum and W. Emmerich, "Algorithms for Efficient Symbolic Detection of Faults in Context-Aware Applications", *Proc. 1st Int'l Workshop on Automated Engineering of Autonomous and Run-Time Evolving Systems (ARAMIS 2008)*, L'Aquila, Italy, Sep. 2008, pp. 1–8.
- C55. L. Duboc, E. Letier, D.S. Rosenblum and T. Wicks, "A Case Study in Eliciting Scalability Requirements", Industrial Practice and Experience Track, *Proc. 16th Int'l Requirements Engineering Conf. (RE 2008)*, Barcelona, Spain, Sep. 2008, pp. 247–252.
- C54. A. Maule, W. Emmerich and D.S. Rosenblum, "Impact Analysis of Database Schema Changes", *Proc. 30th Int'l Conf. Software Engineering (ICSE 2008)*, Leipzig, Germany, May 2008, pp. 451–460.
- C53. M. Sama, D.S. Rosenblum, Z. Wang and S. Elbaum, "Multi-Layer Faults in the Architectures of Mobile, Context-Aware Adaptive Applications: A Position Paper", *Proc. ICSE 2008 Workshop on Software Architectures and Mobility (SAM 2008)*, Leipzig, Germany, May 2008, pp. 47–49.
- C52. A. Mukhija, A. Dingwall-Smith and D.S. Rosenblum, "QoS-Aware Service Composition in Dino", *Proc. 5th IEEE European Conf. Web Services (ECOWS 2007)*, Halle, Germany, Nov. 2007, pp. 3–12.
- C51. L. Duboc, D.S. Rosenblum and T. Wicks, "A Framework for Characterization and Analysis of Software System Scalability", *Proc. Joint 11th European Software Engineering Conf. and 15th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (ESEC/FSE 2007)*, Dubrovnik, Croatia, Sep. 2007, pp. 375–384.
- C50. H. Foster, W. Emmerich, J. Kramer, J. Magee, D.S. Rosenblum and S. Uchitel, "Model Checking Service Compositions under Resource Constraints", *Proc. Joint 11th European Software Engineering Conf. and 15th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (ESEC/FSE 2007)*, Dubrovnik, Croatia, Sep. 2007, pp. 225–234.
- C49. P. Bunyakiati, A. Finkelstein and D. Rosenblum, "The Certification of Software Tools with respect to Software Standards", *Proc. 2007 IEEE Int'l Conf. Information Reuse and Integration (IRI-2007)*, Las Vegas, NV, USA, Aug. 2007, pp. 724–729.
- C48. G.N. Rodrigues, D. Rosenblum and J. Wolf, "Reliability Analysis of Concurrent Systems using LTSA", Informal Demonstration, *Proc. 29th Int'l Conf. Software Engineering Companion Volume (ICSE 2007)*, Minneapolis, MN, USA, May 2007, pp. 63–64.
- C47. Z. Wang, S. Elbaum and D.S. Rosenblum, "Automated Generation of Context-Aware Tests", *Proc. 29th Int'l Conf. Software Engineering (ICSE 2007)*, Minneapolis, MN, USA, May 2007, pp. 406–415.
- C46. L. Popa, C. Raiciu, I. Stoica and D.S. Rosenblum, "Reducing Congestion Effects by Multipath Routing in Wireless Networks", *Proc. 14th Int'l Conf. Network Protocols (ICNP 2006)*, Santa Barbara, CA, USA, Nov. 2006, pp. 96–105.
- C45. C. Raiciu, M. Handley and D.S. Rosenblum, "Exploit Hijacking: Side Effects of Smart Defenses", *Proc. SIGCOMM 2006 Workshop on Large Scale Attack Defense (LSAD 2006)*, Pisa, Italy, Sep. 2006, pp. 123–130.
- C44. C. Raiciu and D.S. Rosenblum, "Enabling Confidentiality in Content-Based Publish/Subscribe Infrastructures", *Proc. Second IEEE Communications Society/CreateNet Int'l Conf. Security and Privacy in Communication Networks (SecureComm 2006)*, Baltimore, MD, USA, Aug.–Sep. 2006, 11 pp.
- C43. C. Raiciu, D.S. Rosenblum and M. Handley, "Revisiting Content-Based Publish/Subscribe", *Proc. 5th Int'l Workshop on Distributed Event-Based Systems (DEBS 2006)*, Lisbon, Portugal, Jul. 2006, 6 pp.
- C42. L. Duboc, D.S. Rosenblum and T. Wicks, "A Framework for Modelling and Analysis of Software Systems Scalability", Doctoral Symposium Paper, *Proc. 28th Int'l Conf. Software Engineering (ICSE 2006)*, Shanghai, China, May 2006, pp. 949–952.

- C41. G.N. Rodrigues, D.S. Rosenblum and S. Uchitel, "Reliability Prediction in Model-Driven Development", *Proc. ACM/IEEE 8th Int'l Conf. Model Driven Engineering Languages and Systems (MoDELS/UML 2005)*, Montego Bay, Jamaica, Oct. 2005, pp. 339–354.
- C40. G.N. Rodrigues, D.S. Rosenblum and S. Uchitel, "Sensitivity Analysis for a Scenario-Based Reliability Prediction Model", *Proc. ICSE 2005 Workshop on Architecting Dependable Systems (WADS 2005)*, St. Louis, MO, USA, May 2005, pp. 73–77.
- C39. G. Rodrigues, D.S. Rosenblum and S. Uchitel, "Using Scenarios to Predict the Reliability of Concurrent Component-Based Software Systems", *Proc. ETAPS 2005 8th Int'l Conf. Fundamental Approaches to Software Engineering (FASE 2005)*, Edinburgh, Scotland, UK, Apr. 2005, pp. 111–126.
- C38. G.N. Rodrigues, D. Rosenblum and W. Emmerich, "A Model Driven Approach for Software Systems Reliability", Doctoral Symposium Paper, *Proc. 26th Int'l Conf. Software Engineering (ICSE 2004)*, Edinburgh, Scotland, UK, May 2004, pp. 30–32.
- C37. A. Orso, M.J. Harrold, D. Rosenblum, G. Rothermel, M.L. Soffa and H. Do, "Using Component Metacontent to Support the Regression Testing of Component-Based Software", *Proc. IEEE Int'l Conf. Software Maintenance 2001 (ICSM 2001)*, Florence, Italy, Nov. 2001, pp. 716–725.
- C36. C. Lüer and D.S. Rosenblum, "WREN—An Environment for Component-Based Development", *Proc. Joint 8th European Software Engineering Conf. and 9th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (ESEC/FSE 2001)*, Vienna, Austria, Sep. 2001, pp. 207–217.
- C35. C. Lüer, D.S. Rosenblum and A. van der Hoek, "The Evolution of Software Evolvability". *Proc. 4th Int'l Workshop on Principles of Software Evolution (IWPSE 2001)*, Vienna, Austria, Sep., 2001, pp. 134–137.
- C34. C. Lüer and D.S. Rosenblum, "UML Component Diagrams and Software Architecture—Experiences from the WREN Project", *Proc. 1st ICSE Workshop on Describing Software Architecture with UML*, Toronto, Ontario, May 2001, pp. 79–82.
- C33. A. Orso, M.J. Harrold and D.S. Rosenblum, "Component Metadata for Software Engineering Tasks", *Proc. 2nd Int'l Workshop on Engineering Distributed Objects (EDO 2000)*, Springer LNCS 1999, Davis, CA, Nov. 2000, pp. 129–144.
- C32. A. Carzaniga, D.S. Rosenblum and A.L. Wolf, "Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service", *Proc. Nineteenth ACM Symp. Principles of Distributed Computing (PODC 2000)*, Portland, OR, Jul. 2000, pp. 219–227.
- C31. N. Medvidovic, R.F. Gamble and D.S. Rosenblum, "Towards Software Multioperability: Bridging Heterogeneous Software Interoperability Platforms", *Proc. Fourth Int'l Software Architecture Workshop (ISAW-4)*, Limerick, Ireland, Jun. 2000, pp. 77–83.
- C30. M.S. Guntersdorfer, D.G. Kay and D.S. Rosenblum, "Using Software Patents To Support The Business Model Of Software Components", *Proc. ICSE 2000 Second Workshop on Commercial Off-The-Shelf Software: Continuing Collaborations for Successful COTS Development*, Limerick, Ireland, Jun. 2000, pp. 32–37.
- C29. N. Medvidovic, A. Egyed and D.S. Rosenblum, "Round-Trip Software Engineering Using UML: From Architecture to Design and Back", *Proc. ESEC/FSE '99 2nd Workshop on Object-Oriented Reengineering*, Toulouse, France, Sep. 1999, Technical University Vienna Technical Report TUV-1841-99-13, pp. 1–8.
- C28. N. Medvidovic, D.S. Rosenblum and R.N. Taylor, "A Language and Environment for Architecture-Based Software Development and Evolution", *Proc. 21st Int'l Conf. Software Engineering (ICSE '99)*, Los Angeles, CA, May 1999, pp. 44–53.
- C27. E. Di Nitto and D.S. Rosenblum, "Exploiting ADLs to Specify Architectural Styles Induced by Middleware Infrastructures", *Proc. 21st Int'l Conf. Software Engineering (ICSE '99)*, Los Angeles, CA, May 1999, pp. 13–22.
- C26. E. Di Nitto and D.S. Rosenblum, "On the Role of Style in Selecting Middleware and Underwear", *Proc. ICSE '99 Workshop on Engineering Distributed Objects (EDO '99)*, Los Angeles, CA, May 1999, pp. 78–83.
- C25. A. Carzaniga, D.S. Rosenblum and A.L. Wolf, "Challenges for Distributed Event Services: Scalability vs. Expressiveness", *Proc. ICSE '99 Workshop on Engineering Distributed Objects (EDO '99)*, May 1999, pp. 72–77.
- C24. N. Medvidovic and D.S. Rosenblum, "Assessing the Suitability of a Standard Design Method for Modeling Software Architectures", *Proc. First Working IFIP Conf. Software Architecture (WICSA-1)*, Feb. 1999, pp. 161–182.

- C23. R. Natarajan and D.S. Rosenblum, "Merging Component Models and Architectural Styles", *Proc. Third Int'l Software Architecture Workshop (ISAW-3)*, Lake Buena Vista, FL, Nov. 1998, pp. 109–111.
- C22. A. Carzaniga, E. Di Nitto, D.S. Rosenblum and A.L. Wolf, "Issues in Supporting Event-based Architectural Styles", *Proc. Third Int'l Software Architecture Workshop (ISAW-3)*, Lake Buena Vista, FL, Nov. 1998, pp. 17–20.
- C21. J.E. Robbins, N. Medvidovic, D.F. Redmiles and D.S. Rosenblum, "Integrating Architecture Description Languages with a Standard Design Method", *Proc. 20th Int'l Conf. Software Engineering (ICSE '98)*, Kyoto, Japan, Apr. 1998, pp. 209–218.
- C20. N. Medvidovic, D.S. Rosenblum and R.N. Taylor, "An Architecture-Based Approach to Software Evolution", *Proc. ICSE '98 Int'l Workshop on the Principles of Software Evolution (IWPSE '98)*, Kyoto, Japan, Apr. 1998, pp. 11–15.
- C19. J.E. Robbins, D.F. Redmiles and D.S. Rosenblum, "Integrating C2 with the Unified Modeling Language", *Proc. 1997 California Software Symposium (CSS '97)*, Irvine, CA, Nov. 1997, pp. 11–18.
- C18. N. Medvidovic and D.S. Rosenblum, "Domains of Concern in Software Architectures and Architecture Description Languages", *Proc. USENIX Conf. Domain-Specific Languages (DSL '97)*, Santa Barbara, CA, Oct. 1997, pp. 199–212.
- C17. D.S. Rosenblum and G. Rothermel, "A Comparative Study of Regression Test Selection Techniques", *Proc. IEEE Computer Society 2nd Int'l Workshop on Empirical Studies of Software Maintenance (WESS-2)*, Bari, Italy, Oct. 1997, pp. 89–94.
- C16. D.S. Rosenblum and A.L. Wolf, "A Design Framework for Internet-Scale Event Observation and Notification", *Proc. Sixth European Software Engineering Conf./ACM SIGSOFT '97 Fifth Symp. Foundations of Software Engineering (ESEC/FSE '97)*, Zürich, Switzerland, Sep. 1997, pp. 344–360. **Winner of the 2008 ACM SIGSOFT Impact Paper Award.**
- C15. D.S. Rosenblum and E.J. Weyuker, "Lessons Learned from a Regression Testing Case Study", *Proc. IEEE Computer Society 1st Int'l Workshop on Empirical Studies of Software Maintenance (WESS-1)*, Monterey, CA, Nov. 1996, pp. 41–43.
- C14. D.S. Rosenblum and E.J. Weyuker, "Predicting the Cost-Effectiveness of Regression Testing Strategies", *Proc. ACM SIGSOFT '96 Fourth Symp. Foundations of Software Engineering (FSE-4)*, San Francisco, CA, Oct. 1996, pp. 118–126.
- C13. N.S. Barghouti and D.S. Rosenblum, "A Case Study in Modeling a Human-Intensive, Corporate Software Process", *Proc. 3rd Int'l Conf. Software Process (ICSP-3)*, Reston, VA, Oct. 1994, pp. 99–110.
- C12. P.T. Devanbu, D.S. Rosenblum and A.L. Wolf, "Automated Construction of Testing and Analysis Tools", *Proc. 16th Int'l Conf. Software Engineering (ICSE-16)*, Sorrento, Italy, May 1994, pp. 241–250.
- C11. Y.-F. Chen, D.S. Rosenblum and K.-P. Vo, "TestTube: A System for Selective Regression Testing", *Proc. 16th Int'l Conf. Software Engineering (ICSE-16)*, Sorrento, Italy, May 1994, pp. 211–220.
- C10. A.L. Wolf and D.S. Rosenblum, "Process-Centered Environments (Only) Support Environment-Centered Processes", *Proc. 8th Int'l Software Process Workshop (ISPW-8)*, Schloß Dagstuhl, Germany, Mar. 1993, pp. 148–149.
- C9. A.L. Wolf and D.S. Rosenblum, "A Study in Software Process Data Capture and Analysis", *Proc. 2nd Int'l Conf. Software Process (ICSP-2)*, Berlin, Germany, Feb. 1993, pp. 115–124.
- C8. D.S. Rosenblum, "Towards a Method of Programming with Assertions", *Proc. 14th Int'l Conf. Software Engineering (ICSE-14)*, Melbourne, Australia, May 1992, pp. 92–104. **Winner of the 2002 ICSE Most Influential Paper Award.**
- C7. D.S. Rosenblum and B. Krishnamurthy, "An Event-Based Model of Software Configuration Management", *Proc. 3rd ACM Int'l Workshop on Software Configuration Management (SCM-3)*, Trondheim, Norway, Jun. 1991, pp. 94–97.
- C6. D.S. Rosenblum and A.L. Wolf, "Representing Semantically Analyzed C++ Code with REPRISE", *Proc. Third USENIX C++ Conf.*, Washington, DC, Apr. 1991, pp. 119–134.
- C5. B. Krishnamurthy and D.S. Rosenblum, "An Event-Action Model of Computer-Supported Cooperative Work: Design and Implementation", *Proc. IFIP Int'l Workshop on Computer Supported Cooperative Work*, Berlin, Germany, Apr. 1991, pp. 132–145.

- C4. D.S. Rosenblum and D.C. Luckham, "Testing the Correctness of Tasking Supervisors with TSL Specifications", *Proc. ACM SIGSOFT '89 3rd Symp. Software Testing, Analysis and Verification (TAV '89)*, Key West, FL, Dec. 1989, pp. 187–196.
- C3. D.S. Rosenblum, S. Sankar and D.C. Luckham, "Concurrent Runtime Checking of Annotated Ada Programs", *Proc. 6th Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS-6)*, Delhi, India, Dec. 1986, pp. 10–35.
- C2. S. Sankar, D.S. Rosenblum and R.B. Neff, "An Implementation of Anna", *Ada in Use: Proc. 1985 Ada Int'l Conf.*, Paris, France, May 1985, Cambridge University Press, pp. 285–296.
- C1. D.S. Rosenblum, "A Methodology for the Design of Ada Transformation Tools in a DIANA Environment", *Proc. 1984 IEEE Computer Society Conf. Ada Applications and Environments*, St. Paul, MN, Oct. 1984, pp. 63–70.

Other Conference and Workshop Papers

- CO18. D.S. Rosenblum, "The Challenges of Probabilistic Thinking", Abstract of Invited Keynote Talk, *Proc. 19th Int'l Conf. Formal Engineering Methods (ICFEM 2017)*, Springer LNCS 10610, Xi'an, China, Nov. 2017, p.xi.
- CO17. D.S. Rosenblum, "The Power of Probabilistic Thinking", Abstract of Invited Keynote Talk, *Proc. 31st IEEE/ACM Int'l Conf. Automated Software Engineering (ASE 2016)*, Singapore, Sep. 2016, p.3.
- CO16. G.A. Lewis, J. Gray, H. Muccini, N. Nagappan, D. Rosenblum and E. Shihab, "1st International Workshop on the Engineering Mobile-Enabled Systems (MOBS 2013)", Workshop Summary, *Proc. 2013 Int'l Conf. Software Engineering (ICSE 2013)*, San Francisco, CA, USA, May 2013, pp. 1533–1534.
- CO15. J. Cubo, F. Raimondi, M. Sama and D. Rosenblum, "A Model to Design and Verify Context-Aware Adaptive Service Composition", Relevant Work Dissemination Track, *Actas de las XV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2010)*, 2010 Congreso Español de Informática (CEDI 2010), Valencia, Spain, Sep. 2010, p. 259. Invited contribution based on #C61.
- CO14. A. Bertolino, W. Grieskamp, R. Hierons, Y. Le Traon, B. Legeard, H. Muccini, A. Paradkar, D. Rosenblum and J. Tretmans, "Model-Based Testing for the Cloud", *Proc. Dagstuhl Seminar 10111: Practical Software Testing: Tool Automation and Human Factors*, Schloß Dagstuhl, Germany, Jun. 26, 2010, 11 pp.
- CO13. D.S. Rosenblum, "Software System Scalability: Concepts and Techniques", Abstract of Invited Keynote Talk, *Proc. 2nd India Software Engineering Conf. (ISEC 2009)*, Pune, India, Feb. 2009, p.1.
- CO12. D.S. Rosenblum and A.L. Wolf, "ACM SIGSOFT Impact Paper Award: Reflections and Prospects", Abstract of Invited Plenary Talk, *Proc. 16th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (FSE 2008)*, Atlanta, GA, USA, Nov. 2008, p. 284. **Invited for presentation of ACM SIGSOFT Impact Paper Award for publication #C16.**
- CO11. M. Jazayeri and D.S. Rosenblum, "Foreword to State-of-the-Art Presentations", *Proc. Joint 11th European Software Engineering Conf. and 15th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (ESEC/FSE 2007)*, Companion Volume, Dubrovnik, Croatia, Sep. 2007, p. 437.
- CO10. C. Raiciu, D.S. Rosenblum, M. Handley, "Distributed Online Filtering", Poster, *SIGCOMM 2007*, Kyoto, Japan, Aug. 2007, 2 pp.
- CO9. M. Sama and D.S. Rosenblum, "ContextNotifier & TestingEmulator: A Toolkit for Developing Adaptive, Context-Aware Applications", Demonstration, *5th Int'l Conf. Mobile Systems, Applications, and Services (MobiSys 2007)*, San Juan, Puerto Rico, Jun. 2007, 1 pp. Accessible from <http://www.sigmobile.org/mobisys/2007/index.html>
- CO8. D.S. Rosenblum, "Fundamental Research Issues in Software Testing", *Proceedings of a European Commission Workshop on Fundamental Research in Software Engineering*, Brussels, Belgium, Jul. 2005, 2 pp.
- CO7. E.J. Whitehead, Jr., R. Khare, R.N. Taylor, D.S. Rosenblum and M.M. Gorlick, "Architectures, Protocols, and Trust for Info-Immersed Active Networks", *Digest of the DARPA/NIST/NSF Workshop on Research Issues in Smart Computing Environments*, Atlanta, GA, Jul. 1999, 3 pp.
- CO6. D.S. Rosenblum, "Challenges in Exploiting Architectural Models for Software Testing", invited submission, *Proc. NSF/CNR Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA '98)*, Marsala, Sicily, Jul. 1998, pp. 49–53.

- CO5. D.S. Rosenblum, A.L. Wolf and A. Carzaniga, "Critical Considerations and Designs for Internet-Scale, Event-Based Compositional Architectures", *Digest of the OMG-DARPA-MCC Workshop on Compositional Software Architectures*, Monterey, CA, Jan. 1998, 4 pp.
- CO4. P. Oreizy, N. Medvidovic, R.N. Taylor and D.S. Rosenblum, "Software Architecture and Component Technologies: Bridging the Gap", *Digest of the OMG-DARPA-MCC Workshop on Compositional Software Architectures*, Monterey, CA, Jan. 1998, 3 pp.
- CO3. D.S. Rosenblum, "Automated Monitoring of Component Integrity in Distributed Object Systems", *Digest of the Advanced Topics Workshop of the 3rd USENIX Conf. Object-Oriented Technologies and Systems*, Portland, OR, Jun. 1997, 3 pp.
- CO2. D.S. Rosenblum and A.L. Wolf, "REPRISE: A Representation for Semantically Analyzed C++ Code", *Digest of the ECOOP/OOPSLA '90 Workshop on Object-Oriented Program Development Environments*, Ottawa, Canada, Oct. 1990, 8 pp.
- CO1. D.S. Rosenblum, "Specification-Based Testing of a Concurrent Software System: An Experience Report", *Digest of the ICSE-12 Workshop on Industrial Experience Using Formal Methods*, Nice, France, May 1990, 7 pp.

Media Features

- M29. Irene Tham, "\$43M for New Lab to Bolster Nation's Cyber Security", *The Straits Times*, Oct. 25, 2016, p. B5. Available online at <http://www.straitstimes.com/singapore/43m-for-new-lab-to-bolster-nations-cyber-security> (last accessed Oct. 31, 2016).
- M28. Neo Chai Chin, "NUS, Singtel Set Up New Lab for Cybersecurity Research", *TODAY Online*, Oct. 24, 2016, <http://www.todayonline.com/singapore/nus-singtel-set-new-lab-cybersecurity-research> (last accessed Oct. 31, 2016).
- M27. Kimberly Spykerman, "NUS and Singtel Launch Cybersecurity Lab", with video clip, *Channel NewsAsia*, Oct. 24, 2016, <http://www.channelnewsasia.com/news/singapore/nus-and-singtel-launch-cybersecurity-lab/3230366.html> (last accessed Oct. 31, 2016).
- M26. Eileen Yu, "Singapore University Partners Singtel to Launch \$30M Cybersecurity Lab", *By the Way, ZDNet*, Oct. 24, 2016, <http://www.zdnet.com/article/singapore-university-partners-singtel-to-launch-30m-cybersecurity-lab/> (last accessed Oct. 31, 2016).
- M25. "People of ACM – David S. Rosenblum", Jan. 5, 2016, <http://www.acm.org/articles/people-of-acm/2016/david-s-rosenblum> (last accessed Jan. 6, 2016).
- M24. T.C. Seow, "New Master's Programme for Regional IT Leaders Launched by NUS-ISS", *MIS Asia*, Jul. 27, 2015, <http://mis-asia.com/tech/industries/new-masters-programme-for-regional-it-leaders-launched-by-nus-iss/> (last accessed Dec. 11, 2015).
- M23. T.C. Seow, "New Master's Programme for Regional IT Leaders Launched by NUS-ISS", *Computerworld Malaysia*, Jul. 27, 2015, <http://www.computerworld.com.my/resource/careers/new-masters-programme-for-regional-it-leaders-launched-by-nus-iss/> (last accessed Dec. 11, 2015).
- M22. T.C. Seow, "New Master's Programme for Regional IT Leaders Launched by NUS-ISS", *Computerworld Singapore*, Jul. 27, 2015, <http://www.computerworld.com.sg/tech/careers/new-masters-programme-for-regional-it-leaders-launched-by-nus-iss/> (last accessed Dec. 11, 2015).
- M21. "NUS-ISS, School of Computing Jointly Launch IT Leadership Master's Degree", *Networks Asia*, Jun. 15, 2015, <http://www.networksasia.net/article/nus-iss-school-computing-jointly-launch-it-leadership-masters-degree.1434335275> (last accessed Jun. 16, 2015).
- M20. Felicia Choo, "NUS Ties Up with Swiss Reinsurance Company", *AsiaOne Business*, Jun. 12, 2015, <http://business.asiaone.com/news/nus-ties-swiss-reinsurance-company> (last accessed Dec. 11, 2015).
- M19. Felicia Choo, "NUS Ties Up with Swiss Reinsurance Company", *The Straits Times*, Money Section, Jun. 11, 2015, p. B10.
- M18. "SAP-NUS Collaboration to Help Students Prepare for the Digital Economy", *Networks Asia*, Jun. 10, 2015, <http://www.networksasia.net/article/sap-nus-collaboration-help-students-prepare-digital-economy.1433899583> (last accessed Jun. 10, 2015).

- M17. "NUS, IBM Partner to Offer Watson Cognitive Systems Education in Southeast Asia", *eGov Innovation*, Oct. 16, 2014, <http://enterpriseinnovation.net/article/nus-ibm-partner-offer-watson-cognitive-systems-education-southeast-asia-2119616538> (last accessed Mar. 20, 2015).
- M16. "My Dear Watson", *Convergence Asia*, Oct. 15, 2014, <http://www.convergenceasia.com/story/my-dear-watson> (last accessed Mar. 20, 2015).
- M15. Gabey Goh, "Singapore Deepens Relationship with IBM's Watson", *Digital News Asia*, Oct. 15, 2014, <http://www.digitalnewsasia.com/digital-economy/singapore-deepens-relationship-with-ibm-watson> (last accessed Mar. 20, 2015).
- M14. "IBM and National University of Singapore Partner to Offer Watson Cognitive Systems Education in Southeast Asia", *The Tech Revolutionist*, Oct. 14, 2014, <http://www.thetechrevolutionist.com/2014/10/ibm-and-national-university-of.html> (last accessed Mar. 20, 2015).
- M13. Mohd Hisham, "NUS to Offer Southeast Asia First IBM Watson-Based Cognitive Education System", *The Rojak Place*, Oct. 14, 2014, <http://mhisham.org/2014/10/14/nus-to-offer-southeast-asia-first-ibm-watson-based-cognitive-education-system/> (last accessed Mar. 20, 2015).
- M12. Nurdianah Md Nur, "IBM Watson to Help Singapore Deliver Better Advice to Its Citizens", *MIS Asia*, Oct. 14, 2014, <http://www.mis-asia.com/resource/applications/ibm-watson-to-help-singapore-deliver-better-advice-to-its-citizens/> (last accessed Mar. 20, 2015).
- M11. Joy Tang, "IBM and NUS to Offer Watson Cognitive Systems Education in Southeast Asia", *TechTrade Asia*, Oct. 14, 2014, <http://www.techtradeasia.info/2014/10/ibm-and-nus-to-offer-watson-cognitive.html> (last accessed Mar. 20, 2015).
- M10. Nurdianah Md Nur, "NUS to Offer IBM Watson Cognitive Systems Education from Next Year", *Computerworld Singapore*, Oct. 14, 2014, <http://www.computerworld.com.sg/resource/applications/nus-to-offer-ibm-watson-cognitive-systems-education-from-next-year/> (last accessed Mar. 20, 2015).
- M9. Nurdianah Md Nur, "NUS to Offer IBM Watson Cognitive Systems Education from Next Year", *Computerworld Malaysia*, Oct. 14, 2014, <http://www.computerworld.com.my/resource/management-and-careers/nus-to-offer-ibm-watson-cognitive-systems-education-from-next-year/> (last accessed Mar. 20, 2015).
- M8. Nurdianah Md Nur, "NUS to Offer IBM Watson Cognitive Systems Education from Next Year", *CIO Asia*, Oct. 14, 2014, <http://www.cio-asia.com/resource/applications/nus-to-offer-ibm-watson-cognitive-systems-education-from-next-year/> (last accessed Mar. 20, 2015).
- M7. Sandra Davie, "Computing Is Cool Again—More and Better Students Apply for Courses in NUS", *The Straits Times*, Jul. 22, 2014, pp. A2–A3. Available online at <http://www.straitstimes.com/singapore/education/computing-is-cool-again-more-and-better-students-apply-for-courses-in-nus> (last accessed Dec. 11, 2015).
- M6. "NUS Collaborates with Japanese Infocomm Company Fuji Xerox on Research Projects", *AsiaOne*, publication of NUS press release, Jan. 27, 2014, <http://news.asiaone.com/news/singapore/nus-collaborates-japanese-infocomm-company-fuji-xerox-research-projects> (last accessed Dec. 11, 2015).
- M5. Joanna Seow, "NUS Collaborates with Japanese Infocomm Company Fuji Xerox on Research Projects", *The Straits Times*, Jan. 27, 2014, <http://www.straitstimes.com/singapore/nus-collaborates-with-japanese-infocomm-company-fuji-xerox-on-research-projects> (last accessed Dec. 11, 2015).
- M4. Jamie Yap, "Software Design Cannot Be Neglected", *ZDNet Asia*, May 9, 2012, <http://www.zdnetasia.com/software-design-cannot-be-neglected-62304732.htm> (last accessed May 14, 2012).
- M3. Don Udell, "Playing the Internet Scales", *InfoWorld*, Mar. 7, 2003, <http://www.infoworld.com/article/2681353/application-development/playing-the-internet-scales.html> (last accessed Dec. 11, 2015).
- M2. Don Clark, "PreCache Survives, With Help From Sony", *The Wall Street Journal*, Vol.CCXLI, No. 1, Jan. 2, 2003, pp. A11–A12. Available online at <http://www.wsj.com/articles/SB1041462709120112473> (last accessed Dec. 11, 2015).

- M1. William E. Suydam, Jr., "Need for Software Testing Moves into Spotlight", *Computer Design*, Vol. 24, No. 15, Nov. 1, 1985, pp. 54–62. Available online at <http://www.wesuydam.com/needForSoftwareTestingNew.htm> (last accessed Dec. 11, 2015).

Theses and Selected Technical Reports

- T13. Z. Tong, Y. Liang, C. Sun, D.S. Rosenblum and A. Lim, *Directed Graph Convolutional Network*, arXiv e-Print arXiv:2004.13970v1 [cs.LG], Apr. 19, 2020, 10 pp. Accessible from <https://arxiv.org/abs/2004.13970>
- T12. L. Liu, Y. Yang, L.N. Govindarajan, S. Wang, B. Hu, L. Cheng and D.S. Rosenblum, *An Interval-Based Bayesian Generative Model for Human Complex Activity Recognition*, arXiv e-Print arXiv:1701.00903v1 [stat.ML], Jan. 4, 2017, 12 pp. Accessible from <https://arxiv.org/abs/1701.00903>
- T11. A. Carzaniga, D.S. Rosenblum and A.L. Wolf, *Content-Based Addressing and Routing: A General Model and its Application*, Department of Computer Science, University of Colorado at Boulder, Technical Report CU-CS-902-00, Jan. 2000.
- T10. N. Medvidovic, D.S. Rosenblum and R.N. Taylor, *A Type Theory for Software Architectures*, Department of Information and Computer Science, University of California, Irvine, Technical Report UCI-ICS-98-14, Apr. 1998, 10 pp.
- T9. D.S. Rosenblum, *Adequate Testing of Component-Based Software*, Department of Information and Computer Science, University of California, Irvine, Technical Report UCS-ICS-97-34, Aug. 1997, 9 pp.
- T8. D.S. Rosenblum, *Visualization of Informally Defined Software Processes*, AT&T Bell Laboratories Technical Report, Nov. 1992, 3 pp.
- T7. D.S. Rosenblum, *Petri Net Case Study*, in J.J. Kenney and W. Mann (eds.), *Anna Package Specification: Case Studies*, Technical Report CSL-TR-91-496, Computer Systems Laboratory, Stanford University, Oct. 1991, pp. 87–94.
- T6. D.S. Rosenblum, *Towards Petri Net-Based Analysis of Behavior Specifications for Concurrent Programs*, AT&T Bell Laboratories Technical Report, Jan. 1991, 7 pp.
- T5. P. Frankl and D.S. Rosenblum, *Using Annotations for Selection and Evaluation of Software Test Data*, AT&T Bell Laboratories Technical Report, Oct. 1989, 6 pp.
- T4. D.S. Rosenblum, *Event-Based Semantics of Concurrent Programming Languages*, AT&T Bell Laboratories Technical Report, Sep. 1988, 24 pp.
- T3. D.S. Rosenblum, *Design and Verification of Distributed Tasking Supervisors for Concurrent Programming Languages*, **Ph.D. Dissertation**, Technical Report CSL-TR-88-357, Computer Systems Laboratory, Stanford University, Mar. 1988, 218 pp.
- T2. D.S. Rosenblum and E.W. Mayr, *Simulation of an Ultracomputer with Several 'Hot Spots'*, Technical Report STAN-CS-86-1119, Dept. of Computer Science, Stanford University, Jun. 1986, 33 pp.
- T1. D.S. Rosenblum, *An Implementation of the IEEE Standard for Binary Floating Point Arithmetic for the Motorola 6809 Microprocessor*, **M.S. Thesis**, Dept. of Computer Sciences, North Texas State University, Aug. 1983, 83 pp.

Presentations and Tutorials

Keynote Talks and Other Plenary Conference Talks

- "The Challenges of Probabilistic Thinking", 19th International Conference on Formal Engineering Methods (ICFEM 2017), Xi'an, China, Nov. 2017
- "The Power of Probabilistic Thinking", 31st ACM/IEEE International Conference on Automated Software Engineering (ASE 2017), Singapore, Sep. 2016
- "Career Management", ICSE 2014 New Faculty and Researcher Symp. (ICSE NFRS 2014), Hyderabad, India, Jun. 2014
- "Probability and Uncertainty in Software Engineering", 2013 National Software Application Conf. (NASAC 2013), Tianjin, China, Nov. 2013
- "Whither Software Engineering Research?", 19th Asia-Pacific Software Engineering Conf. (APSEC 2012), Hong Kong, China, Dec. 2012

- "Software System Scalability: Concepts and Techniques", 2nd India Software Engineering Conference (ISEC 2009), Pune, India, Feb. 2009
- "ACM SIGSOFT Impact Paper Award: Reflections and Prospects" (with Alexander L. Wolf), 2008 ACM SIGSOFT Impact Paper Award, 16th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (FSE 2008), Atlanta, GA, USA, Nov. 2008
- "Scalability: What It Is and How to Analyse It", XXI Sympósio Brasileiro de Engenharia de Software (SBES 2007), João Pessoa, Paraíba, Brazil, Oct. 2007
- "Analysis of Software Architectures: Achievements and Challenges", 18th Int'l Workshop on Algebraic Development Techniques (WADT '06), La Roche en Ardenne, Belgium, Jun. 2006
- "Wide-Area Publish/Subscribe: A Re-Assessment", 2005 Int'l Symp. Distributed Objects and Applications (DOA 2005), Agia Napa, Cyprus, Oct. 2005
- "Some Open Problems in Publish/Subscribe Networking", Second Int'l Workshop on Distributed Event-Based Systems (DEBS '03), San Diego, CA, USA, Jun. 2003
- "Assertions a Decade Later", ICSE '92 Most Influential Paper Award, 2002 Int'l Conf. Software Engineering, Orlando, FL, May 2002
- "Internet Scale Event Notification", Keynote Talk at Workshop on Internet Scale Event Notification, Irvine, CA, Jul. 1998
- "An Efficient Communication Kernel for Distributed Ada Runtime Tasking Supervisors", Best Graduate Student Paper Award, ACM SIGAda/AdaJUG Joint Meeting., Hollywood, FL, Jan. 1987

Other Invited Conference and Workshop Talks and Panels

- Panelist, "ICSE 2016 Town Hall on ICSE 2017 Submission Limit", 38th Int'l Conf. Software Engineering (ICSE 2016), Austin, TX, USA, May 2016
- "A Brief Look at Probabilistic Model Checking", Dagstuhl Seminar #15017 on Formal Foundations for Networking, Schloß Dagstuhl, Germany, Feb. 2015
- "How Is Computer Science Changed by Other Disciplines?", Panel "Interdisciplinarity: The Future of Computer Science?", Microsoft Research Asia Faculty Summit 2014, Beijing, China, Oct. 2014
- Panelist, "Does Infrastructure Really Matter When Moving to the Front Lines?", IBM Technology Conference & Expo 2014, Singapore, Jul. 2014
- "The Mother of All Disciplines?", Pre-Summit Workshop for Deans, Department Chairs and Research Directors, 9th European Computer Science Summit (ECSS 2013), Amsterdam, The Netherlands, Oct. 2013
- "A Random Walk Down Memory Lane", IEEE Computer Society Fellows' Distinguished Public Forum, Biopolis, Singapore, Oct. 2013
- "Felicitous Computing", NUS School of Computing Annual Research Workshop, NUS Research Institute, Suzhou, China, Jan. 2013
- "Applications and Abstractions: A Cautionary Tale", DIMACS Working Group on Abstractions for Network Services, Architecture, and Implementation, Rutgers University, Piscataway, NJ, USA, May 2012
- "Felicitous Computing", 5th Annual NUS School of Computing-China Research Workshop, National University of Singapore, Singapore, Jan. 2012
- "High Confidence Ubiquitous Computing Systems", 2011 Int'l Symp. High Confidence Software (ISHCS 2011), Beijing, China, Dec. 2011
- Panelist, "Future of Software Research in Asia-Pacific/China Region", 2011 Int'l Symp. High Confidence Software (ISHCS 2011), Beijing, China, Dec. 2011
- Panelist, "Regression Testing", Dagstuhl Seminar #10111 on Practical Software Testing: Tool Automation and Human Factors, Schloß Dagstuhl, Germany, Mar. 2010
- Panelist, "Academic-Industrial Cooperation in Software Engineering", XXI Sympósio Brasileiro de Engenharia de Software (SBES 2007), João Pessoa, Paraíba, Brazil, Oct. 2007

- "The Software Engineering Impact Project: Testing & Analysis" (with Lori A. Clarke), 2006 Int'l Symp. Software Testing and Analysis (ISSTA 2006), Portland, ME, USA, Jul. 2006
- "Testability and Reliability of Component-Based Software", Pre-ICSE 2006 Workshop on Int'l Software Engineering Cooperation, Shanghai, China, Oct. 2004
- Panellist on Panel "Distributed Event-Base Processing in a Bigger Context", Second Int'l Workshop on Distributed Event-Based Systems (DEBS '03), San Diego, CA, Jun. 2003
- "Publish/Subscribe Architectures: Attractions, Problems and Lessons", Panel "Event-Based Systems and Software Architectures: Out of the Shadows and Into the Mainstream", 1st Annual Research Forum, UC Irvine Institute for Software Research, Irvine, CA, Jun. 2002
- Panellist, "Who Needs Doctors? Reporting from the Doctoral Workshop", 22nd Int'l Conf. Software Engineering (ICSE 2000), Limerick, Ireland, Jun. 2000
- "Challenges in Exploiting Architectural Models for Software Testing", NSF/CNR Int'l Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA), Marsala, Sicily, Italy, Jul. 1998
- "An Overview of Selective Regression Testing", Software Engineering Tools and Technologies (SETT) Java Testing Workshop, Irvine, CA, Nov. 1996

Inaugural Lecture as Professor of Software Systems, University College London

- "Scalability in Software Systems Engineering: The Good, the Bad, and the Ugly", 13 Dec. 2004 (with Vote of Thanks by Alexander L. Wolf)

Other Invited Talks

- "The Power and Challenges of Probabilistic Reasoning for Software Engineering", Faculty of Computer and Information Technology, Universiti Putra Malaysia, Oct. 2018
- "The Power and Challenges of Probabilistic Reasoning for Software Engineering", Huawei Research & Development Competence Centre (Shenzhen), May 2018
- "The Power and Challenges of Probabilistic Reasoning for Software Engineering", Faculty of Computer and Information Sciences, Hosei University, Feb. 2018
- "Uncertainty in Computer Systems: Problems and Results", Department of Computer Science, University of Southern California, Jan. 2017
- "Jogging While Driving, and Other Software Engineering Research Problems", Department of Computer Science Distinguished Lecture Series, Univ. of Illinois at Chicago, Apr. 2014
- "Felicitous Computing", Institute for Software Research (ISR) Distinguished Speaker Series, University of California, Irvine, Apr. 2013
- "Scalability of Software Systems: Concepts and Techniques", Univ. de Buenos Aires, Jul. 2012
- "Automated Fault Identification in Context-Aware Adaptive Applications", National University of Singapore, Aug. 2010
- "Automated Fault Identification in Context-Aware Adaptive Applications", University of Texas at Dallas, Jun. 2010
- "Scalability: What It Is and How to Analyse It", Univ. of Massachusetts-Amherst, Mar. 2008
- "Predicting the Reliability of Concurrent Component-Based Software", Università della Svizzera Italiana, Lugano, Switzerland, Dec. 2005
- "Predicting the Reliability of Concurrent Component-Based Software", Brunel University, London, UK, Nov. 2005
- "Predicting the Reliability of Concurrent Component-Based Software", Georgia Institute of Technology, Atlanta, GA, USA, Jun. 2005
- "Publish/Subscribe Networking: Open Problems and Recent Results", University of Nebraska-Lincoln, Lincoln, NE, USA, Mar. 2005
- "Predicting the Reliability of Concurrent Component-Based Software", University of Nebraska-Lincoln, Lincoln, NE, USA, Mar. 2005

"Publish/Subscribe Networking: Open Problems and Recent Results", Intel Research Cambridge, Cambridge, UK, Mar. 2005

"Predicting the Reliability of Concurrent Component-Based Software", Università dell'Aquila, L'Aquila, Italy, Mar. 2005

"Two Decades of Software Systems Research", University College London, UK, Oct. 2003

"Publish/Subscribe Networking: Concepts, Results, Open Problems", Università della Svizzera Italiana, Lugano, Switzerland, Jun. 2003

"Publish/Subscribe Networking: Concepts, Results, Open Problems", University College London, London, UK, May 2003

"Publish/Subscribe Networking: Concepts, Results, Open Problems", University of Southern California, Los Angeles, CA, May 2003

"Reconciling Software Architecture Models and Software Component Standards", Politecnico di Milano, Milan, Italy, Jan. 1999

"Reconciling Software Architecture Models and Software Component Standards", Technische Universität Wien, Vienna, Austria, Jan. 1999

"Bringing Software Architecture into the Software Development Process", Southern California Process Improvement Network (SPIN), Irvine, CA, Jun. 1998

"Internet-Scale Event Observation and Notification", Bay Area Round Table (BART), Palo Alto, CA, Feb. 1998

"Validation of Distributed Component-Based Software Systems", Bay Area Round Table (BART), Palo Alto, CA, Aug. 1997

"Predicting the Cost-Effectiveness of Regression Testing Methods", University of California, Irvine, Irvine, CA, May 1996

"A Specification-Based Approach to Automated Fault Detection", University of Colorado at Boulder, Boulder, CO, Apr. 1996

"A Specification-Based Approach to Automated Fault Detection", University of Maryland, College Park, MD, Mar. 1996

"A Specification-Based Approach to Automated Fault Detection", Columbia University, New York, NY, Feb. 1996

"Experience with a Selective Regression Testing System", Clemson U., Clemson, SC, Oct. 1995

"Software Specification", Columbia University, New York, NY, Oct. 1995

"Software Specification", Rutgers University, New Brunswick, NJ, Feb. 1995

"Experience with a System for Selective Regression Testing", Polytechnic University, Brooklyn, NY, Oct. 1994

"Software Research at Bell Labs", Stanford University, Stanford, CA, Aug. 1994

"Testtube: A System for Selective Regression Testing", Siemens Corporate Research, Princeton, NJ, Mar. 1994

"Specifying and Developing Programs with Anna", New York University, New York, NY, Mar. 1991

"Validation of Tasking Supervisors with TSL Specifications", Polytechnic University, Brooklyn, NY, Nov. 1989

"Automated Testing of Distributed Ada Tasking Supervisors", AT&T Bell Laboratories, Holmdel, NJ, Sep. 1987

"Automated Testing of Distributed Ada Tasking Supervisors", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1987

"An Efficient Distributed Implementation of Ada", Los Angeles Chapter of ACM SIGAda, Redondo Beach, CA, May 1987

- “Automated Testing of Distributed Ada Tasking Supervisors”, Bell Communications Research, Piscataway, NJ, Apr. 1987
- “Automated Testing of Distributed Ada Tasking Supervisors”, IBM T.J. Watson Research Center, Hawthorne, NY, Apr. 1987
- “A Distributed Implementation of Ada”, Supercomputing Research Center, Institute for Defense Analyses, Lanham, MD, Apr. 1986

Conference Tutorials

- Scalability of Software Systems: Concepts and Techniques*, 5th IFIP TC2 Summer School & 2nd ACM/IEEE ICSE 2010 Warm-Up Workshop (WUP 2009), Strand, South Africa, Apr. 2009
- Formal Software Engineering* (with Alexander L. Wolf), ACM SIGSOFT '95 Third Symp. Foundations of Software Engineering, Washington, DC, USA, Oct. 1995
- Formal Software Engineering* (with Alexander L. Wolf), ACM SIGSOFT '94 Second Symp. Foundations of Software Engineering, New Orleans, LA, USA, Dec. 1994

Other Conference Presentations

- Int'l Conf. Software Engineering (ICSE): 1992, 1994, 2007
- ACM SIGSOFT '96 Fourth Symp. Foundations of Software Engineering (FSE): 1996
- SIGSOFT '89: 3rd Symp. Software Testing, Analysis and Verification (TAV): 1989
- IEEE Computer Society 1984 Conf. Ada Applications and Environments (AAE): 1984

Conference and Workshop Committees

Steering Committees

- Int'l Conf. Software Engineering (ICSE), 2001–2007, 2009–2021 (Chair 2004–2007)
- Int'l Symp. Software Testing and Analysis (ISSTA), 2005–2014 (Chair 2007–2008)
- European Joint Conf. Theory and Practice of Software (ETAPS), 2008–2011
- Int'l Workshop and Conf. Distributed Event-Based Systems (DEBS), 2006–2009
- Int'l Conf. Model Driven Engineering Languages and Systems (MoDELS/UML), 2005–2009
- European Software Engineering Conf. (ESEC), 2005–2009
- Workshop on Program Analysis for Software Tools and Engineering (PASTE), 2005–2009
- Int'l Workshop on Software and Performance (WOSP), 2005–2009
- Int'l Symp. Automated and Analysis-Driven Debugging (AADEBUG), 2005–2006

Conference General Chair

- 2007 Int'l Symp. Software Testing and Analysis (ISSTA 2007)

Program or Track Chair/Co-Chair of International Events

- Journal-First Presentations, 44th Int'l Conf. Software Engineering (ICSE 2022)
- 50th Anniversary of Software Engineering, 40th Int'l Conf. Software Engineering (ICSE 2018)
- Technical Briefings, 39th Int'l Conf. Software Engineering (ICSE 2017)
- Visions of 2025 and Beyond Track, 38th Int'l Conf. Software Engineering (ICSE 2016)
- ICSE 2004 Most Influential Paper, 36th Int'l Conf. Software Engineering (ICSE 2014)
- ICSE 2013 Int'l Workshop on the Engineering of Mobile-Enabled Systems (MOBS 2013)
- ETAPS 2010 13th Int'l Conf. Fundamental Approaches to Software Engineering (FASE 2010)
- State-of-the-Art Presentations, Joint 11th European Software Engineering Conf. and 15th ACM SIGSOFT Int'l Symp. Foundations of Software Engineering (ESEC/FSE 2007)
- Doctoral Symposium, 29th Int'l Conf. Software Engineering (ICSE 2007)
- Conference Coordinator, Social Programs, 28th Int'l Conf. Software Engineering (ICSE 2006)
- State-of-the-Practice Track, 27th Int'l Conf. Software Engineering (ICSE 2005)
- 26th Int'l Conf. Software Engineering (ICSE 2004)
- ACM SIGSOFT Eighth Int'l Symp. Foundations of Software Engineering (FSE 2000)
- Workshop on Evaluating Software Architectural Solutions (WESAS 2000)
- Research Demos and Posters, 22nd Int'l Conf. Software Engineering (ICSE 2000)
- Publicity, ACM SIGSOFT Sixth Int'l Symp. Foundations of Software Engineering (FSE 1998)
- Publicity, 17th Int'l Conf. Software Engineering (ICSE 1995)

Program Committee Member (Series)

- Int'l Conf. Software Engineering (ICSE): 1997, 2000, 2001, 2002, 2003, 2005, 2011, 2012, 2013

SIGSOFT Int'l Symp. Foundations of Software Engineering (FSE): 1998, 2004, 2008, 2010, 2018
 European Software Engineering Conf. (ESEC/FSE): 1999, 2001, 2005
 SIGSOFT Int'l Symp. Software Testing & Analysis (ISSTA): 1994, 1996, 2004, 2008, 2010
 Int'l Workshop/Conf. Distributed Event-Based Systems (DEBS): 2002, 2003, 2004, 2006, 2007
 Int'l Conf. Automated Software Engineering (ASE): 1998, 1999
 SIGSOFT Symp. Component-Based Software Engineering (CBSE): 2011, 2012

Program Committee Member (Other)

Doctoral Symposium, 43rd Int'l Conf. Software Engineering (ICSE 2021)
 1st Workshop on AI Engineering – Software Engineering for AI, 43rd Int'l Conf. Software Engineering (WAIN 2021)
 New and Emerging Ideas Track, 2018 Int'l Conf. Software Architecture (ICSA 2018)
 First International Workshop on Establishing a Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE 2017)
 12th Int'l Symp. Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2017)
 Doctoral Symposium, 37th Int'l Conf. Software Engineering (ICSE 2015)
 Advisory Committee, 2nd Int'l Conf. Big Data Analytics in Healthcare (BDAH 2014)
 First Int'l Conf. Mobile Software Engineering and Systems (MOBILESoft 2014)
 Tutorial and Technical Briefings Track, 36th Int'l Conf. Software Engineering (ICSE 2014)
 FSE 2012 New Ideas and Emergent Results Track
 4th Int'l IEEE Conferences on Software – Science, Technology and Engineering (SwSTE 2010)
 2009 Workshop on Architecting Dependable Systems (WADS 2009)
 2nd ACM/IEEE ICSE 2010 Warm-Up Workshop (WUP 2009)
 FSE 2008 New Software Engineering Faculty Symposium
 7th ACM Workshop on Software and Performance (WOSP 2008)
 Second Int'l Conf. Tests and Proofs (TAP 2008)
 Doctoral Symposium, ACM 2008 Int'l Symp. Software Testing and Analysis (ISSTA 2008)
 Doctoral Symposium, 30th Int'l Conf. Software Engineering (ICSE 2008)
 ETAPS 2008 11th Int'l Conf. Fundamental Approaches to Software Engineering (FASE 2008)
 3rd Int'l IEEE Conferences on Software – Science, Technology and Engineering (SwSTE 2007)
 Most Promising New Work Track, 2007 Haifa Verification Conference (HVC 2007)
 6th ACM Workshop on Software and Performance (WOSP 2007)
 2006 Int'l Symp. Trustworthy Global Computing (TGC 2006)
 4th Int'l Workshop on Distributed and Mobile Collaboration (DMC 2006)
 Workshop on Web Services—Modeling and Testing (WS-MaTe 2006)
 Doctoral Symposium, ACM/IFIP/USENIX 6th Int'l Middleware Conf. (Middleware 2005)
 2nd Int'l Workshop on Software Quality (SOQUA 2005)
 5th Int'l Workshop on Software Engineering and Middleware (SEM 2005)
 Fifth Working IFIP/IEEE Conf. Software Architecture (WICSA 2005)
 UML 2004 Workshop on Software Architecture Description and UML
 3rd Int'l Workshop on Software Engineering and Middleware (SEM 2002)
 2nd Int'l Workshop on Engineering Distributed Objects (EDO 2000)
 Doctoral Symposium, 22nd Int'l Conf. Software Engineering (ICSE 2000)
 Second USENIX Conf. Domain-Specific Languages (DSL '99)
 ICSE '99 Workshop on Engineering Distributed Objects (EDO '99)
 First Working IFIP Conf. Software Architecture (WICSA1)
 Eighth Annual California Software Symposium (CSS '98)
 DEXA '98 Int'l Workshop on Large-Scale Software Composition
 ISSTA '98 Second Workshop on Formal Methods in Software Practice
 Seventh Annual California Software Symposium (CSS '97)
 Second Int'l Workshop on Empirical Studies of Software Maintenance (WESS '97)
 ISSTA '96 Workshop on Formal Methods in Software Practice
 8th Int'l Workshop on Software Specification and Design (IWSSD-8)
 1993 Conf. Computer Software and Applications (COMPSAC '93)
 Tutorials, 14th Int'l Conf. Software Engineering (ICSE-14)

Selected Reviewer Service

IEEE Computer Society Fellows Committee
 IEEE Senior Member Review Panel (UK and Republic of Ireland Section)
 The Royal Society
 International Grant Assessments Panel

International Fellowships Panel
 Engineering and Physical Sciences Research Council (EPSRC)
 Review Panel, Technology Programme, Computer Science, Nov. 2005
 Prioritisation Panel, Information and Communications Technology Programme, Jun. 2008
 Peer Review College, 2006–2009
 National Science Foundation (NSF) Review Panels
 Computer and Information Science and Engineering (CISE) Directorate, May 2000
 Computer and Information Science and Engineering (CISE) Directorate, Feb. 1997
 Vienna Science and Technology Fund (WWTF)
 Jury Member, Information and Communication Technology (ICT) Call, 2010

Hosting of Extended Visits by Academics

National University of Singapore

Anthony I. Wasserman, Carnegie Mellon University West, 2014
 Carlo Ghezzi, sabbatical leave from Politecnico di Milano, 2014
 Mehdi Jazayeri, sabbatical leave from Università della Svizzera Italiana, 2015

University College London

Richard N. Taylor, sabbatical leave from University of California, Irvine, 2005
 Sebastian Elbaum, sabbatical leave from University of Nebraska–Lincoln, 2006–2007
 Leon J. Osterweil, sabbatical leave from University of Massachusetts–Amherst, 2007
 Lori A. Clarke, sabbatical leave from University of Massachusetts–Amherst, 2007

Doctoral and Post-Doctoral Supervision and Examination

In addition to doctoral and post-doctoral supervision, I have supervised individual and group projects, theses and internships for over one hundred Bachelors and Masters Students

Post-Doctoral *University of California, Irvine*

Supervisor Elisabetta Di Nitto, 1998; now Full Professor at Politecnico di Milano

University College London

Arun Mukhija, 2006–2009; now Vendor Governance Lead at Roche, Switzerland
 Andrew Dingwall-Smith, 2006–2007; now Senior Developer at Message Automation
 Genáina Rodrigues, 2008; now Professora at Universidade de Brasília
 Franco Raimondi, 2008–2009; now Professor of Computer Science at Middlesex U.
 Sonia Ben Mokhtar, 2009; now Senior Researcher at CNRS

National University of Singapore

Darren Carlson, 2013–2016; now Asst. Professor at U. of Hawai’I at Mānoa
 Guoxin Su, 2013–2017; now Lecturer at University of Wollongong
 Li Liu, 2013–2016; now Associate Professor at Chongqing University
 Omar al-Bataineh, 2018; now Research Fellow at Nanyang Technological U.
 Ye Liu, 2018–2019; now Research Engineer at Twitter

Ph.D. *University College London*

Supervisor Jorge Luis Ortega Arjona, 2007; thesis *Architectural Patterns for Parallel Programming – Models for Performance Estimation*; now Professor at Universidad Nacional Autónoma de México, Mexico
 Genáina Nunes Rodrigues, 2008; thesis *A Model Driven Approach for Software Reliability Prediction*; now Professora at Universidade de Brasília, Brazil
 Ana Leticia De Cerqueira Leite Duboc, 2010; thesis *A Framework for the Characterization and Analysis of Software Systems Scalability*; now Researcher at La Salle – Ramon Llull University, Spain
 Costin Raiciu, 2011; thesis *ROAR: Increasing the Flexibility and Performance of Distributed Search*; now Associate Professor at Universitatea Politehnica din Bucuresti, Romania
 Michele Sama, 2011; thesis *Context-Driven Methodologies for Context-Aware and Adaptive Systems*; now Vice President of Engineering and Chief Technology Officer at Gluru, UK
 Fokion Zervoudakis, 2014; thesis *Cascading Verification: An Integrated Method for Domain-Specific Model Checking*; now Senior Software Engineer at Broadridge Financial Solutions, UK

Panagiotis Papakos, 2014; thesis *Volare Mobile Context-aware Adaptation for the Cloud*;
now Software Engineer at RateSetter, UK

National University of Singapore

Ratul Saha, 2018 (co-supervised with P.S. Thiagarajan); thesis *Quantitative Model Checking of Distributed Probabilistic Systems*; now Konductor at GetonChain, Singapore

Yongpo Jia, 2018; thesis *Multi-Source Learning from Social Network Data*; now Data Analytics Research Manager at HeBei Iron & Steel Group (HBIS Group), China

Marc Sven Brünink, 2018; thesis *Analyzing the Behavior of Deployed Software*; now Site Reliability Engineer at Google Zürich, Switzerland

Ye Liu, 2018; thesis *User Attribute Learning from Social Media and Ubiquitous Sensors*; now Senior Research Engineer at Twitter, USA

Yamilet Rosario Serrano Llerena, 2018; thesis *Applications of Perturbation Analysis in Probabilistic Model Checking*; now Assistant Professor at Universidad de Ingeniería y Tecnología, Peru

Nirandika Wanigasekara, 2019; thesis *Decision Models for Context-Aware IoT Applications*; now Data Analyst at IBM Sydney, Australia

Kun Ouyang, 2020; thesis *Human Mobility Analytics with Deep Representations*; now Software Engineer at Shopee, Singapore

Thilina Thanthriwatta, 2021; thesis *Modeling Dynamic Aspects of Context-Aware Recommender Systems*

Ph.D. Robert Chatley, Imperial College London, 2005
External Octavian Patrascoiu, University of Kent, 2005
Examiner Ashok Argent-Katwala, Imperial College London, 2006
Alexander Buckley, Imperial College London, 2006
William Heaven, Imperial College London, 2007
Lucio Duarte, Imperial College London, 2007
Gürkan Güleşir, De Universiteit Twente, 2008
Matteo Migliavacca, Politecnico di Milano, 2008
Konstantinos Adamopoulos, M.Phil., King's College London, 2008
Eskinder Asmare, Imperial College London, 2009
Paolo Di Benedetto Università dell'Aquila, 2010
Xiaoyuan Xie, Swinburne University of Technology, 2012

Ph.D. *University of Colorado at Boulder*
Committee Jonathan E. Cook, 1996
Member

University of California, Irvine

Gregory A. Bolcer, 1998	E. James Whitehead, Jr. , 2000
Nenad Medvidovic, 1998	Roy T. Fielding, 2000
Juei Chang, 1999	Michael Kantor, 2001
David M. Hilbert, 1999	Chang Liu, 2002
Jason E. Robbins, 1999	Peter Kammer, 2004
Peyman Oreizy, 2000	

Politecnico di Milano

Antonio Carzaniga, 1998

National University of Singapore

Yongzheng Wu, 2012	Zhu Shenggao, 2017
Jian Zhu, 2012	Shin Hwei Tan, 2018
Marcel Böhme, 2014	Pin Sym Foong, 2018
Chen Penghe, 2015	Quang Trung Ta, 2018
Narcisa Andreea Milea, 2015	Shruti Shrikant Tople, 2018
Limiao Bai, 2016	Mathugamavithanage Dilruk Dasanatha Perera, 2020

George Mason University

Yuxing Chen, 2021

Visiting *University College London*
Ph.D. Javier Cubo, Universidad de Málaga, 2008
Student

Teaching

North Texas State University

IBM System 370 Assembly Language: Fall 1982 (2 sections), Spring 1983 (2 sections)

Rutgers University

Software Engineering: Spring 1996

University of California, Irvine

Software Architectures, Distributed Systems and Interoperability: Spring 1998, Spring 1999, Fall 1999, Fall 2000

Software Engineering: Fall 1998, Winter 2001

Software Testing and Analysis: Spring 1999

Formal Methods in Software Engineering: Spring 1999, Winter 2001

Software Processes: Fall 1997, Winter 2000

Project in System Design: Spring 1998, Fall 1998

Introduction to Software Engineering: Spring 1997

Systematic Software Construction: Winter 1997

Internet-Scale Software Engineering: Fall 1999

Topics in Distributed Object Computing: Spring 1997

Seminar in Software: Fall 1997

University College London

Database and Information Management Systems: 2004–05

Technology Management and Professional Issues: 2004–05, 2005–06, 2006–07, 2007–08, 2008–09, 2009–10, 2010–11

Advanced Analysis and Design: 2005–06, 2006–07, 2007–08, 2008–09, 2009–10, 2010–11

Software Engineering: 2005–06, 2006–07

Architecture and Design of Large-Scale Software Systems: 2006–07, 2007–08, 2008–09

Validation and Verification: 2007–08, 2008–09, 2009–10

National University of Singapore

Automated Software Validation: 2011–12, 2012–13, 2014–15, 2016–17, 2017–18, 2018–19, 2019–20

Software Engineering Project I: 2012–13

Other University Service

University of California, Irvine

Bay Area Round Table (BART), Irvine Research Unit in Software (IRUS) 1997–1998

Representative Assembly, Academic Senate, 1996–1998

Faculty Advisor for Undeclared / Undecided Undergraduate Students, 1997–1999

University Research Park Advisory Committee, 1998–2001

Council on Educational Policy, 1998–1999

Student Discipline Grievance Review Board, Sep. 10, 1997

ICS Department Software Faculty Search Committee, 1998–1999, 2000–2001

ICS Department Graduate Policy Committee, 2000–2001

ICS Department Graduate Policy and Ph.D. Admissions Committee, 1998–1999

ICS Department Graduate Admissions Committee, 1997–1998

ICS Department Undergraduate Policy Committee, 1997–1998

ICS Department Graduate Policy and Admissions Committee, 1996–1997

University College London

UCLse Centre for Systems Engineering, 2005–2011