

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC, SAMSUNG ELECTRONICS CO. LTD.,
AND SAMSUNG ELECTRONICS AMERICA, INC.,
Petitioner

v.

MULTIMODAL MEDIA LLC,
Patent Owner.

Case No. IPR2024-00063
U.S. Patent No. 9,185,227

**DECLARATION OF DR. BENJAMIN B. BEDERSON IN SUPPORT
OF PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 9,185,227**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND AND QUALIFICATIONS	2
III. INFORMATION CONSIDERED	9
IV. RELEVANT LEGAL STANDARDS	9
A. Claim Interpretation	9
B. Perspective of One of Ordinary Skill in the Art.....	10
C. Obviousness.....	10
V. LEVEL OF ORDINARY SKILL IN THE ART	13
VI. SUMMARY OF MY OPINIONS	14
VII. TECHNOLOGICAL BACKGROUND	15
A. Call completion applications were well known.	15
B. Call completion actions, other than voicemail, were well known.	23
VIII. THE '227 PATENT AND ITS FILE HISTORY	29
IX. CLAIM CONSTRUCTION	32
A. “call completion application” (Claims 1, 5, 6, 9, 10, 11, and 15).....	32
1. The specification supports the plain and ordinary meaning	33
2. The dependent claims contradict the <i>Oppo</i> construction.....	33
3. The agreed-upon construction makes no sense	34
B. “one of text data, audio data, video data, audiovisual data, image data, multimedia data, message data, and any combination thereof” (Claims 8, 17).....	35
X. OVERVIEW OF THE APPLIED PRIOR ART REFERENCES	36
A. Prikowitsch (Ex-1011)	36
B. Roujinsky (Ex-1012).....	38
C. Sar-Shalom (Ex-1013).....	39

TABLE OF CONTENTS
(continued)

	Page
XI. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS	41
A. Ground 1: Claims 1, 2, 5, 8-12, 15, and 17-19 are Obvious Over Prikowitsch	41
1. Independent Claims 1 and 11	41
a. Elements 1[pre]/11[pre]: A computer implemented method/[A computer system] for completing an incomplete call made by a calling party to a called party, comprising	41
b. Element 1[a]: providing a call completion application executable by at least one processor on a calling party device;	42
Element 11[a]: a non-transitory computer readable storage medium configured to store a call completion application on a calling party device;	42
Element 11[b]: at least one processor communicatively coupled to said non-transitory computer readable storage medium, said at least one processor configured to execute said call completion application on said calling party device; and	42
c. Element 1[b]: detecting said incomplete call made by said calling party to said called party, by said call completion application on said calling party device, ...	48
Element 1[c]: wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events;	48
Element 11[c]: said call completion application on said calling party device, comprising:	48
Element 11[d]: a call detection module configured to detect said incomplete call made by said calling party to said called party, ...	48

TABLE OF CONTENTS
(continued)

	Page
d. Element 1[d(i)]: receiving one or more of a plurality of call completion actions to be performed by said call completion application on said calling party device to complete said detected incomplete call,.....	50
Element 11[e(i)]: an action determination module configured to receive one or more of a plurality of call completion actions to be performed to complete said detected incomplete call, ...	50
e. Elements 1[d(ii)]/11[e(ii)]: wherein said plurality of call completion actions are selected according to one or more responses provided by said calling party through a graphical user interface of said calling party device,.....	52
f. Elements 1[d(iii)]/11[e(iii)]: wherein said call completion actions comprise [1] setting a reminder to call back said called party at a configurable time, [2] recording media data on said calling party device, [3] transmitting said media data to a called party device, [4] transmitting a missed call alert to said called party device, [5] transmitting a notification of said detected incomplete call to said called party device, [6] transmitting a notification on availability of said called party, [7] transmitting said media data to a social networking platform, [8] transmitting an automated message requesting said called party to call back said calling party when available, and [9] any combination thereof; and.....	53
g. Element 1[e]: triggering execution of said received one or more of said call completion actions by said call completion application on said calling party device based on one or more of action execution criteria for said completion of detected incomplete call.....	56

TABLE OF CONTENTS

(continued)

Page

	Element 11[f]: an action execution module configured to trigger execution of said received one or more of said call completion actions based on one or more of action execution criteria one or more responses selected by said calling party for said completion of said detected incomplete call.....	56
2.	Dependent Claims 2, 5, 8-10, 12, 15, and 17-19	58
a.	Element 12[a]: The computer system of claim 11, wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events,.....	58
b.	Claim 2/Element 12[b]: [The computer implemented method of claim 1, / The computer system of claim 11,] wherein said events comprise said called party being busy, said called party device being in an out of coverage area, said called party device being unreachable, said called party device being switched off, network congestion, and said call not being answered by said called party.	58
c.	Claims 5 and 15	60
d.	Claims 8/17: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said media data comprises one of text data, audio data, video data, audiovisual data, image data, multimedia data, message data, and any combination thereof.	66

TABLE OF CONTENTS
(continued)

	Page
e. Claims 9/18: [The computer implemented method of claim 1, further comprising determining / The computer system of claim 11, wherein said action execution module is further configured to determine] a mode of transmission of media data from said calling party device to a called party device via a network, by said call completion application on said calling party device based on one or more of said action execution criteria.	67
f. Claims 10/19: The computer implemented method of claim 1, further comprising monitoring/[The computer system of claim 11, wherein said call detection module is further configured to monitor] said incomplete call by said call completion application on said calling party device as said incomplete call progresses from said incomplete call being initiated, said incomplete call being one of answered and rejected, and said incomplete call being terminated.	69
B. Ground 2: Claims 3, 4, 9, 13, 14, and 18 are Rendered Obvious by Prikowitsch in View of Roujinsky	71
1. A POSITA would have been motivated to combine Prikowitsch and Roujinsky and had a reasonable expectation of success.....	71
2. Dependent Claims 3, 4, 9, 13, 14, and 18	74
a. Claims 3/13: [The computer implemented method of claim 1/The computer system of claim 11], wherein said incomplete call is a call of a short duration that does not enable an intended communication to occur between said calling party and said called party.	74

TABLE OF CONTENTS

(continued)

	Page
b. Claim 4: The computer implemented method of claim 1, wherein said one or more of said call completion actions are received by said call completion application based on a duration of said detected incomplete call 75 - Claim 14: The computer system of claim 11, wherein said action determination module is further configured to receive said one or more of said call completion actions based on a duration of said detected incomplete call.....75 c. Claims 9/18: [The computer implemented method of claim 1, further comprising determining / The computer system of claim 11, wherein said action execution module is further configured to determine] a mode of transmission of media data from said calling party device to a called party device via a network, by said call completion application on said calling party device based on one or more of said action execution criteria. 77	
C. Ground 3: Claims 1-2, 6, 8-12, and 17-19 are Rendered Obvious by Sar-Shalom 78	
1. Independent Claims 1 and 11.....78 a. Elements 1[pre]/11[pre]: A computer implemented method/[A computer system] for completing an incomplete call made by a calling party to a called party, comprising 78 b. Element 1[a]: providing a call completion application executable by at least one processor on a calling party device;.....79 - Element 11[a]: a non-transitory computer readable storage medium configured to store a call completion application on a calling party device;.....79	

TABLE OF CONTENTS

(continued)

Page

	Element 11[b]: at least one processor communicatively coupled to said non-transitory computer readable storage medium, said at least one processor configured to execute said call completion application on said calling party device; and	79
c.	Element 1[b]: detecting said incomplete call made by said calling party to said called party, by said call completion application on said calling party device, ...	82
	Element 1[c]: wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events;	82
	Element 11[c]: said call completion application on said calling party device, comprising:	82
	Element 11[d]: a call detection module configured to detect said incomplete call made by said calling party to said called party	82
d.	Element 1[d(i)]: receiving one or more of a plurality of call completion actions to be performed by said call completion application on said calling party device to complete said detected incomplete call, ...	84
	Element 11[e(i)]: an action determination module configured to receive one or more of a plurality of call completion actions to be performed to complete said detected incomplete call, ...	84
e.	Elements 1[d(ii)]/11[e(ii)]: wherein said plurality of call completion actions are selected according to one or more responses provided by said calling party through a graphical user interface of said calling party device, ...	86

TABLE OF CONTENTS
(continued)

	Page
f. Elements 1[d(iii)]/11[e(iii)]: wherein said call completion actions comprise [1] setting a reminder to call back said called party at a configurable time, [2] recording media data on said calling party device, [3] transmitting said media data to a called party device, [4] transmitting a missed call alert to said called party device, [5] transmitting a notification of said detected incomplete call to said called party device, [6] transmitting a notification on availability of said called party, [7] transmitting said media data to a social networking platform, [8] transmitting an automated message requesting said called party to call back said calling party when available, and [9] any combination thereof; and.....	89
g. Element 1[e]: triggering execution of said received one or more of said call completion actions by said call completion application on said calling party device based on one or more of action execution criteria for said completion of detected incomplete call.....	91
Element 11[f]: an action execution module configured to trigger execution of said received one or more of said call completion actions based on one or more of action execution criteria one or more responses selected by said calling party for said completion of said detected incomplete call.....	91
2. Dependent Claims 2, 6, 8-10, 12, and 17-19	93
a. Element 12[a]: The computer system of claim 11, wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events,.....	93

TABLE OF CONTENTS
(continued)

Page

b.	Claim 2/Element 12[b]: [The computer implemented method of claim 1,] wherein said events comprise said called party being busy, said called party device being in an out of coverage area, said called party device being unreachable, said called party device being switched off, network congestion, and said call not being answered by said called party.	94
c.	Claim 6: The computer implemented method of claim 1, wherein said one or more of said call completion actions to be performed by said call completion application on said calling party device to complete said detected incomplete call is based on one or more of configurable criteria configured by said calling party.	95
d.	Claims 8/17: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said media data comprises one of text data, audio data, video data, audiovisual data, image data, multimedia data, message data, and any combination thereof.	95
e.	Claims 9/18: [The computer implemented method of claim 1, further comprising determining / The computer system of claim 11, wherein said action execution module is further configured to determine] a mode of transmission of media data from said calling party device to a called party device via a network, by said call completion application on said calling party device based on one or more of said action execution criteria.	96

TABLE OF CONTENTS
(continued)

	Page
f. Claims 10/19: [The computer implemented method of claim 1, further comprising monitoring / The computer system of claim 11, wherein said call detection module is further configured to monitor] said incomplete call by said call completion application on said calling party device as said incomplete call progresses from said incomplete call being initiated, said incomplete call being one of answered and rejected, and said incomplete call being terminated.	97
D. Ground 4: Claims 3-5 and 13-15 are Obvious Over Sar-Shalom in View of Roujinsky	98
1. A POSITA would have been motivated to combine Sar-Shalom and Roujinsky and had a reasonable expectation of success	98
2. Dependent Claims 3-5 and 13-15	101
a. Claims 3/13: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said incomplete call is a call of a short duration that does not enable an intended communication to occur between said calling party and said called party.	101
b. Claim 4: The computer implemented method of claim 1, wherein said one or more of said call completion actions are received by said call completion application based on a duration of said detected incomplete call	103
Claim 14: The computer system of claim 11, wherein said action determination module is further configured to receive said one or more of said call completion actions based on a duration of said detected incomplete call.	103
c. Claims 5 and 15	104

TABLE OF CONTENTS
(continued)

	Page
XII. CONCLUSION.....	112

LIST OF EXHIBITS

Ex-1001	U.S. Patent No. 9,185,227
Ex-1002	Declaration of Dr. Benjamin B. Bederson
Ex-1003	Curriculum Vitae of Dr. Benjamin B. Bederson
Ex-1004	File History of U.S. Patent No. 9,185,227
Ex-1005	INTENTIONALLY LEFT BLANK
Ex-1006	INTENTIONALLY LEFT BLANK
Ex-1007	INTENTIONALLY LEFT BLANK
Ex-1008	INTENTIONALLY LEFT BLANK
Ex-1009	INTENTIONALLY LEFT BLANK
Ex-1010	INTENTIONALLY LEFT BLANK
Ex-1011	PCT Publication WO 2010139478 (“Prikowitsch”)
Ex-1012	PCT Publication WO2008050325 (“Roujinsky”)
Ex-1013	U.S. Patent Publication No. 20040267884 (“Sar-Shalom”)
Ex-1014	INTENTIONALLY LEFT BLANK
Ex-1015	INTENTIONALLY LEFT BLANK
Ex-1016	INTENTIONALLY LEFT BLANK
Ex-1017	INTENTIONALLY LEFT BLANK
Ex-1018	INTENTIONALLY LEFT BLANK
Ex-1019	INTENTIONALLY LEFT BLANK
Ex-1020	Claim Mapping Table
Ex-1021	<i>Multimodal Media LLC v. Guangdong Oppo Mobile Telecommunications Corp., Ltd.</i> , 2:21-cv-00436, D.I. 65 (E.D. Tex. Apr. 24, 2023) (Claim Construction Order) (“ <i>Oppo</i> Litigation”)

Ex-1022	<i>Multimodal Media LLC v. TCL Technology Group Corporation et al.</i> , 2:22-cv-00463, D.I. 23 (E.D. Tex. Feb. 22, 2023) (Stay Order) (“TCL litigation”)
Ex-1023	<i>Multimodal Media LLC v. Samsung Electronics Co., Ltd. et al.</i> , 2:22-cv-00462, Infringement Contentions (E.D. Tex. May 30, 2023) (“Samsung Infringement Contentions”)
Ex-1024	INTENTIONALLY LEFT BLANK
Ex-1025	INTENTIONALLY LEFT BLANK
Ex-1026	Benjamin B. Bederson & James D. Hollan, Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, UIST ‘94 Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology 17 (1994), DOI: http://dx.doi.org/10.1145/192426.192435
Ex-1027	INTENTIONALLY LEFT BLANK
Ex-1028	INTENTIONALLY LEFT BLANK
Ex-1029	INTENTIONALLY LEFT BLANK
Ex-1030	INTENTIONALLY LEFT BLANK
Ex-1031	David Dearman, Amy Karlson, Brian Meyers, and Ben Bederson, Multi-modal text entry and selection on a mobile device (2010), In Proceedings of Graphics Interface 2010 (GI ‘10), Canadian Information Processing Society, CAN, 19–26
Ex-1032	INTENTIONALLY LEFT BLANK
Ex-1033	INTENTIONALLY LEFT BLANK
Ex-1034	INTENTIONALLY LEFT BLANK
Ex-1035	INTENTIONALLY LEFT BLANK
Ex-1036	INTENTIONALLY LEFT BLANK
Ex-1037	INTENTIONALLY LEFT BLANK
Ex-1038	INTENTIONALLY LEFT BLANK

Ex-1039	INTENTIONALLY LEFT BLANK
Ex-1040	INTENTIONALLY LEFT BLANK
Ex-1041	INTENTIONALLY LEFT BLANK
Ex-1042	INTENTIONALLY LEFT BLANK
Ex-1043	INTENTIONALLY LEFT BLANK
Ex-1044	INTENTIONALLY LEFT BLANK
Ex-1045	INTENTIONALLY LEFT BLANK
Ex-1046	<i>Multimodal Media LLC v. Guangdong Oppo Mobile Telecommunications Corp., Ltd.</i> , 2:21-cv-00436, D.I. 74 (E.D. Tex. Sept. 6, 2023) (Order of Dismissal)
Ex-1047	Rebecca L.J. Chen, et al., Business Models and Drivers for Next-Generation IMS Services (2007)
Ex-1048	U.S. Patent No. 6,639,973 (“Wheeler”)
Ex-1049	U.S. Patent Publication No. 20030169865 (“Oren”)
Ex-1050	U.S. Patent Publication No. 20090142085 (“Anderl”)
Ex-1051	U.S. Patent No. 8,340,702 (“Bychkov”)
Ex-1052	U.S. Patent No. 8,229,409 (“Bluvand”)
Ex-1053	Harvey G. Cragon, Computer Architecture and Implementation (2000)
Ex-1054	Nokia 7650 User’s Guide

1. I, Dr. Benjamin B. Bederson, declare as follows.

I. INTRODUCTION

2. I have been retained by Google LLC, Samsung Electronics Co. Ltd., and Samsung Electronics America, Inc. (collectively, “Petitioner”) as an independent expert consultant in this inter partes review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”).

3. I have been asked by Petitioner Counsel (“Counsel”) to consider whether certain references disclose, teach and/or suggest the features recited in Claims 1-6, 8-15, and 17-19 of U.S. Patent No. 9,185,227 (“the ’227 Patent”) (Ex-1001), which I understand is currently assigned to Patent Owner Multimodal Media LLC (“PO”).¹ My opinions and the bases for my opinions are set forth below.

4. I am being compensated at my ordinary and customary consulting rate for my work, which is \$600 per hour. My compensation is in no way contingent on the nature of my findings, the presentation of my findings in testimony, or the outcome of this or any other proceeding. I have no other direct financial interest in this proceeding.

¹ Where appropriate, I refer to exhibits that I understand are attached to the petition for IPR of the ’227 Patent.

II. BACKGROUND AND QUALIFICATIONS

5. All of my opinions stated in my Declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in designing, developing, researching, and teaching the technology referenced in my Declaration.

6. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth in my Declaration. I understand that a copy of my current curriculum vitae, which details my education and professional and academic experience, is being submitted as Ex-1003. The following provides a brief overview of some of my experience that is relevant to the matters set forth in my Declaration.

7. I am currently Professor Emeritus of Computer Science at the University of Maryland (“UMD”). From 2014 to 2018, I was the Associate Provost of Learning Initiatives and Executive Director of the Teaching and Learning Transformation Center at the UMD. I am a member and previous director of the Human-Computer Interaction Lab (“HCIL”), the oldest and one of the best known Human-Computer Interaction (“HCI”) research groups in the country. I am co-founder of the J.S. Bryant School, a therapeutic high school planning to open in 2025. I was also co-founder and Chief Scientist of Zumobi, Inc. from 2006 to 2014, a Seattle-based startup that is a publisher of content applications and advertising

platforms for smartphones. I am also co-founder and co-director of the International Children's Digital Library ("ICDL"), a web site launched in 2002 that provides the world's largest collection of freely available online children's books from around the world at www.childrenslibrary.org with an interface aimed to make it easy for children and adults to search and read children's books online. I was also co-founder and prior Chief Technology Officer of Hazel Analytics, a data analytics company to improve food safety and better public health whose product sends alerts in warranted circumstances. In addition, I have for more than 15 years consulted for numerous companies in the area of user interfaces, including Microsoft, Logitech, the Palo Alto Research Center, Sony, Lockheed Martin, Hillcrest Labs, and NASA Goddard Space Flight Center.

8. The claims of the '227 Patent generally relate to voice short message service messaging system. For more than 30 years, I have studied, designed, and worked in the field of computer science and HCI. My experience includes 30 years of teaching and research, with research interests in HCI and the software and technology underlying today's interactive computing systems. This includes the design and implementation of hardware and software systems including the use of interactive voice interfaces as well as many other kinds of sensors and interfaces for electronic devices on a range of devices, including embedded systems, controllers, smart phones and PDAs.

9. At UMD, I have been focused primarily on the area of HCI, a field that relates to the development and understanding of computing systems to serve users' needs. Researchers and practitioners in this field are focused on making universally usable, useful, efficient, and appealing systems to support people in their wide range of activities. My approach is to balance the development of innovative technology that serves people's practical needs. Example systems following this approach that I have built include Cortex-I (1992 embedded computer vision system that sensed licensed plates with custom motor, camera and controller), Audio Augmented Reality (1995 embedded system for sensing a user's location and playing audio suited to that location), Fisheye Menus (2000 software for sensing movement within and selection of linear list of items in a menu), PhotoMesa (2001 software for end users to browse personal photos), DateLens (2002 software for end users to use their mobile devices to efficiently access their calendar information), SlideBar (2005 linear sensor to control scrolling), LaunchTile (2005 "home screen" software for mobile devices to allow users to navigate apps in a zoomable environment), SpaceTree (2001 software for end users to efficiently browse very large hierarchies), ICDL (as I describe above), and StoryKit (a 2009 iPhone app for children to create stories).

10. I built a voice interface system in 2009, described in a 2010 paper², to support user's rich text entry with multi-modal input on mobile devices to support communications including text messages. As illustrated in this figure from the paper, we enabled users to enter formatted text (bold, colored, etc.) using a touch screen, voice recognition, tilt, and a foot pedal and we measured which approach was most effective.

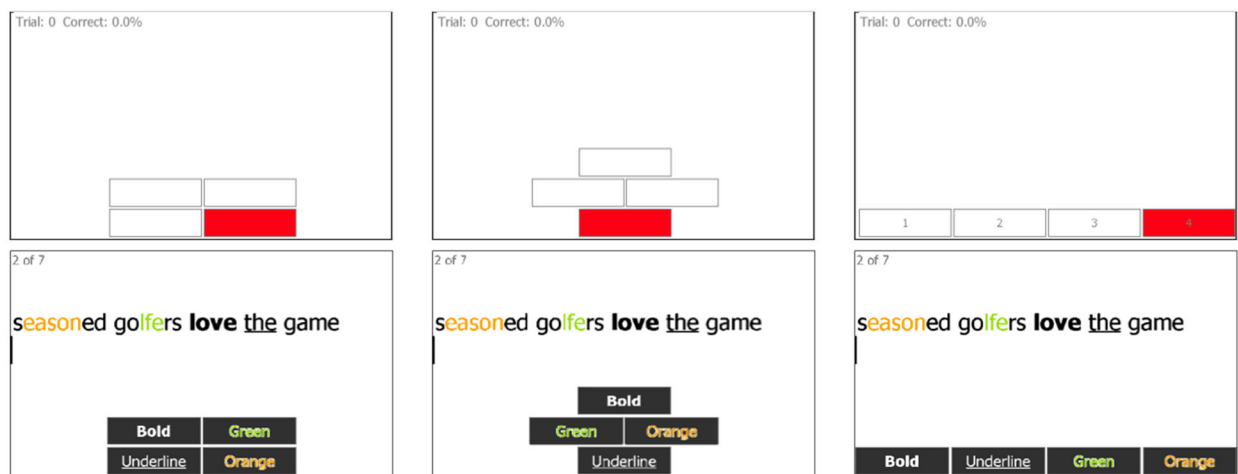


Figure 2. The software interface for the Target Selection (top) and Text Formatting (bottom) tasks. Presented are the Foot (left), Tilt (middle) and Touch/Voice (right) interfaces. The target to select for the Target Selection task is red. The phrase used in the Text Formatting task includes all four formats: 'eason' is orange; 'lfe' is green; 'love' is bold; and 'the' is underlined.

11. Throughout the 1990s and 2000s, I worked on a range of “zoomable user interfaces,” which are systems that support the multi-scale and spatial organization of and magnification-based navigation among multiple documents or visual objects. I built several different “ZUI” systems over the years, including

² David Dearman, Amy Karlson, Brian Meyers, and Ben Bederson, Multi-modal text entry and selection on a mobile device (2010), In Proceedings of Graphics Interface 2010 (GI '10), Canadian Information Processing Society, CAN, 19–26 (Ex-1031).

Pad++, Jazz and Piccolo. In those systems, I used a range of solutions to allow users to control zooming through the information space.³

12. My work with ZUIs led to my creation of Zumobi in 2006, where I was responsible for investigating new software platforms and developing new user interface designs that provide efficient and engaging interfaces to permit end users to access a wide range of content on mobile platforms (including the iPhone and Android-based devices). For example, I designed and implemented software called “Ziibii,” a “river” of news for iPhone that used a capacitive sensor for controlling linear movement through news, software called “ZoomCanvas,” a zoomable user interface for several iPhone apps, and iPhone apps including “Inside Xbox” for Microsoft and Snow Report for REI.

13. At the ICDL, I have since 2002 been the technical director responsible for the design and implementation of the web site, www.childrenslibrary.org (originally at www.icdlbooks.org). In particular, I have been closely involved in designing the user interface as well as the software architecture for the web site since its inception in 2002. Beginning in the mid-1990s, I have been responsible for the design and implementation of numerous other web sites in addition to the ICDL. For

³ Benjamin B. Bederson & James D. Hollan, Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, UIST '94 Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology 17 (1994), DOI: <http://dx.doi.org/10.1145/192426.192435> (Ex-1026).

example, I designed and built my own professional web site when I was an Assistant Professor of Computer Science at the University of New Mexico in 1995 and have continued to design, write the code for, and update both that site (which I moved to the UMD in 1998, currently at <http://www.cs.umd.edu/~bederson/>) as well as numerous project web sites, such as Pad++, <http://www.cs.umd.edu/hcil/pad++/>.

14. I received the Janet Fabri Memorial Award for Outstanding Doctoral Dissertation for my Ph.D. work in robotics and computer vision. My work has been published extensively in more than 160 technical publications, and I have given about 100 invited talks, including 9 keynote lectures. I have won a number of awards including the Brian Shackel Award for “outstanding contribution with international impact in the field of HCI” in 2007, and the Social Impact Award in 2010 from the Association for Computing Machinery’s (“ACM”) Special Interest Group on Computer Human Interaction (“SIGCHI”). ACM is the primary international professional community of computer scientists, and SIGCHI is the primary international professional HCI community. I have been honored by both professional organizations. I am an “ACM Distinguished Scientist,” which “recognizes those ACM members with at least 15 years of professional experience and 5 years of continuous Professional Membership who have achieved significant accomplishments or have made a significant impact on the computing field.” I am a member of the “CHI Academy,” which is described as follows: “The CHI Academy

is an honorary group of individuals who have made substantial contributions to the field of HCI. These are the principal leaders of the field, whose efforts have shaped the disciplines and/or industry, and led the research and/or innovation in human-computer interaction.” The criteria for election to the CHI Academy are: (1) cumulative contributions to the field; (2) impact on the field through development of new research directions and/or innovations; and (3) influence on the work of others. I have received two “Test of Time” awards from IEEE InfoVis for my 2001 and 2002 work on visualizing hierarchies, including those containing digital photos.⁴

15. I have appeared on radio shows numerous times to discuss issues relating to user interface design and people’s use and frustration with common technologies, web sites, and mobile devices. My work has been discussed and I have been quoted by mainstream media around the world over 120 times, including by the New York Times, the Wall Street Journal, the Washington Post, Newsweek, the Seattle Post Intelligencer, the Independent, Le Monde, NPR’s All Things Considered, New Scientist Magazine, and MIT’s Technology Review.

16. I have designed, programmed, and publicly deployed dozens of user-facing software products that have cumulatively had millions of users. My work is cited by several major companies, including Amazon, Apple, Facebook, Google, and

⁴ <https://ieevis.org/year/2021/info/awards/test-of-time-awards;>
[https://ieevis.org/year/2022/info/awards/test-of-time-awards.](https://ieevis.org/year/2022/info/awards/test-of-time-awards)

Microsoft. I am a named inventor on 14 U.S. patents and 20 U.S. patent applications.

The patents are generally directed to user interfaces/experience.

17. I received a B.S. degree in Computer Science with a minor in Electrical Engineering in 1986 from the Rensselaer Polytechnic Institute. I received M.S. and Ph.D. degrees in Computer Science in 1989 and 1992, both from New York University.

III. INFORMATION CONSIDERED

18. In preparation for my Declaration, I have considered the materials I discuss in my Declaration, including, for example, the '227 Patent, the references cited by the '227 Patent, the prosecution history of the '227 Patent (including the references cited therein), various background articles and materials referenced in my Declaration, and the prior art references I identify in my Declaration. In addition, my opinions are further based on my education, training, experience, and knowledge in the relevant field.

IV. RELEVANT LEGAL STANDARDS

19. I am not an attorney and offer no legal opinions. For the purposes of my Declaration, I have been informed about certain aspects of the law that are relevant to my analysis, as summarized below.

A. Claim Interpretation

20. I have been informed and understand that in an IPR proceeding, claims are to be interpreted according to the Phillips claim construction standard. I

understand that under this standard, the meaning of claim terms is considered from the viewpoint of a hypothetical “person of ordinary skill in the art” (“POSITA”) at the time of the alleged invention. I have been informed and understand that claim construction is a matter of law and that the final claim constructions for this proceeding will be determined by the Patent Trial and Appeal Board (“PTAB”).

21. I provide my opinions regarding terms that may require explicit construction to resolve the grounds I present in my Declaration in Section XI, ¶¶75-162.

B. Perspective of One of Ordinary Skill in the Art

22. I have been informed and understand that a patent is to be understood from the perspective of a POSITA. Such an individual is considered to possess normal skills and knowledge in a particular technical field (as opposed to being a genius). I understand that in considering what the claims of a patent require, what was known prior to that patent, what a prior art reference discloses, and whether an invention is obvious or not, one must use the perspective of such a POSITA.

C. Obviousness

23. I have been informed and understand that a patent claim is obvious under 35 U.S.C. §103, and therefore invalid, if the claimed subject matter, as a whole, would have been obvious to a POSITA as of the priority date of the patent based on one or more prior art references and/or the knowledge of a POSITA.

24. I understand that an obviousness analysis must consider (1) the scope and content of the prior art, (2) the differences between the claims and the prior art, (3) the level of ordinary skill in the pertinent art, and (4) secondary considerations, if any, of non-obviousness (such as unexpected results, commercial success, long-felt but unmet need, failure of others, copying by others, and skepticism of experts).

25. I understand that a prior art reference may be combined with other references to disclose each element of the invention under 35 U.S.C. § 103. I understand that a reference may also be combined with the knowledge of a POSITA, and that this knowledge may be used to combine multiple references. I further understand that a POSITA is presumed to know the relevant prior art. I understand that the obviousness analysis may take into account the inferences and creative steps that a POSITA would employ.

26. In determining whether a prior art reference would have been combined with other prior art or other information known to a POSITA, I understand that the following principles may be considered:

- a. whether the references to be combined involve non-analogous art;
- b. whether the references to be combined are in different fields of endeavor than the alleged invention in the Patent;
- c. whether the references to be combined are reasonably pertinent to the problems to which the inventions of the Patent are directed;

- d. whether the combination is of familiar elements according to known methods that yields predictable results;
- e. whether a combination involves the substitution of one known element for another that yields predictable results;
- f. whether the combination involves the use of a known technique to improve similar items or methods in the same way that yields predictable results;
- g. whether the combination involves the application of a known technique to a prior art reference that is ready for improvement, to yield predictable results;
- h. whether the combination is “obvious to try”;
- i. whether the combination involves the known work in one field of endeavor prompting variations of it for use in either the same field or a different one based on design incentives or other market forces, where the variations are predictable to a POSITA;
- j. whether there is some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill in the art to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- k. whether the combination requires modifications that render the prior

art unsatisfactory for its intended use;

- l. whether the combination requires modifications that change the principle of operation of the reference;
- m. whether the combination is reasonably expected to be a success; and
- n. whether the combination possesses the requisite degree of predictability at the time the invention was made.

27. I understand that in determining whether a combination of prior art references renders a claim obvious, it is helpful to consider whether there is some teaching, suggestion, or motivation to combine the references and a reasonable expectation of success in doing so. I understand, however, that a teaching, suggestion, or motivation to combine is not required.

V. LEVEL OF ORDINARY SKILL IN THE ART

28. I am familiar with the level of ordinary skill in the art with respect to the '227 Patent around its filing date. Based on my experience working, teaching, and conducting research in the relevant field, and based on my review of the '227 Patent specification, claims, file history, and prior art, I believe one of ordinary skill in the art around the time of the alleged invention of the '227 Patent would have been someone with at least a bachelor's degree in electrical engineering, computer science, computer engineering, or a related field, and two-three years of experience

with multimedia and Internet-related communications technologies, or the equivalent, with additional education substituting for experience and vice versa.

29. In determining the level of ordinary skill in the art, I considered, for example, the type of problems encountered in the art, prior art solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field.

30. I met or exceeded the knowledge and experience that a POSITA would have had in 2012. In my work as a professor and entrepreneur, I worked with graduate students and others who had the knowledge and experience of a POSITA in 2012, and I am able to render opinions from the perspective of a POSITA based on that knowledge and experience. My opinions concerning the '227 Patent claims and the prior art are from the perspective of a POSITA, as set forth above.

VI. SUMMARY OF MY OPINIONS

31. As I explain below in detail in my Declaration, it is my opinion that Claims 1-6, 8-15, and 17-19 of the '227 Patent would have been obvious under 35 U.S.C. §103 on the following grounds:

- Ground 1: Claims 1-2, 5, 8-12, 15, and 17-19 are obvious over Prikowitsch (Ex-1011)
- Ground 2: Claims 3-4, 9, 13-14, and 18 are obvious over Prikowitsch (Ex-1011) in view of Roujinsky (Ex-1012)

- Ground 3: Claims 1-2, 6, 8-12, and 17-19 are obvious over Sar-Shalom (Ex-1013)
- Ground 4: Claims 3-5 and 13-15 are obvious over Sar-Shalom (Ex-1013) in view of Roujinsky (Ex-1012)

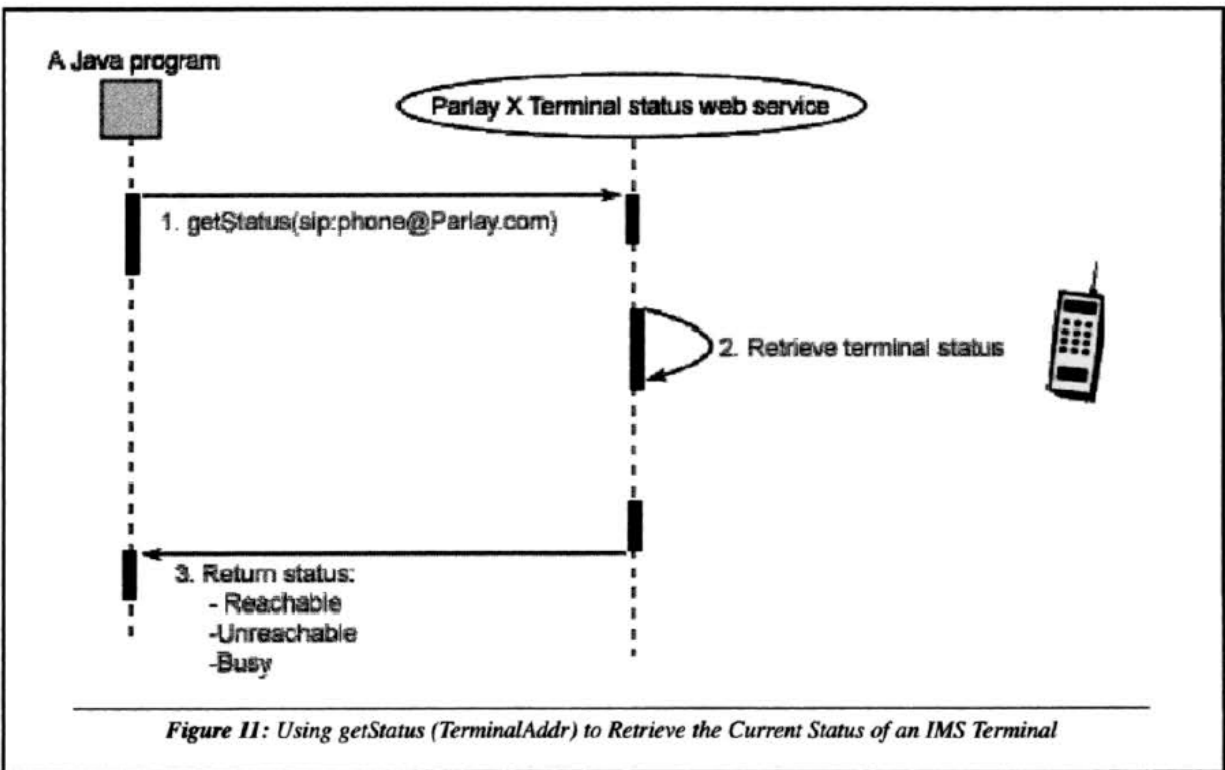
VII. TECHNOLOGICAL BACKGROUND

A. Call completion applications were well known.

32. Internet protocol multimedia subsystems (IMS) were popular alternatives to leaving a voicemail. For example, in 2007, Rebecca Chen, Victor Chen, Elisa Su, and Yo-Hong Wang explained that “IMS” systems were the “next ‘big thing’ in the telecom industry because the systems integrated services provided by long-existing IP network with the mobility of digital devices.” Ex-1047, 9 (“the Chen Paper”). In the Chen paper, IMS is explained to “fill the gap between the existing traditional telecommunications technology and internet technology” in order to provide the environment of cell phone devices with IP based services available on the internet. *Id.* Through IMS, IBM created Parlay X Web Services. Parlay X Web Services (“Parlay X”) allowed users to initiate a call connection between two phones through high-level service interface. *Id.*, 14. After establishing the call, a call identifier is returned representing the specific call session such as “call status, the duration of the session, the call start time, and termination cause, if the session has been terminated.” *Id.* Parlay X allowed the user to cancel a call and stop

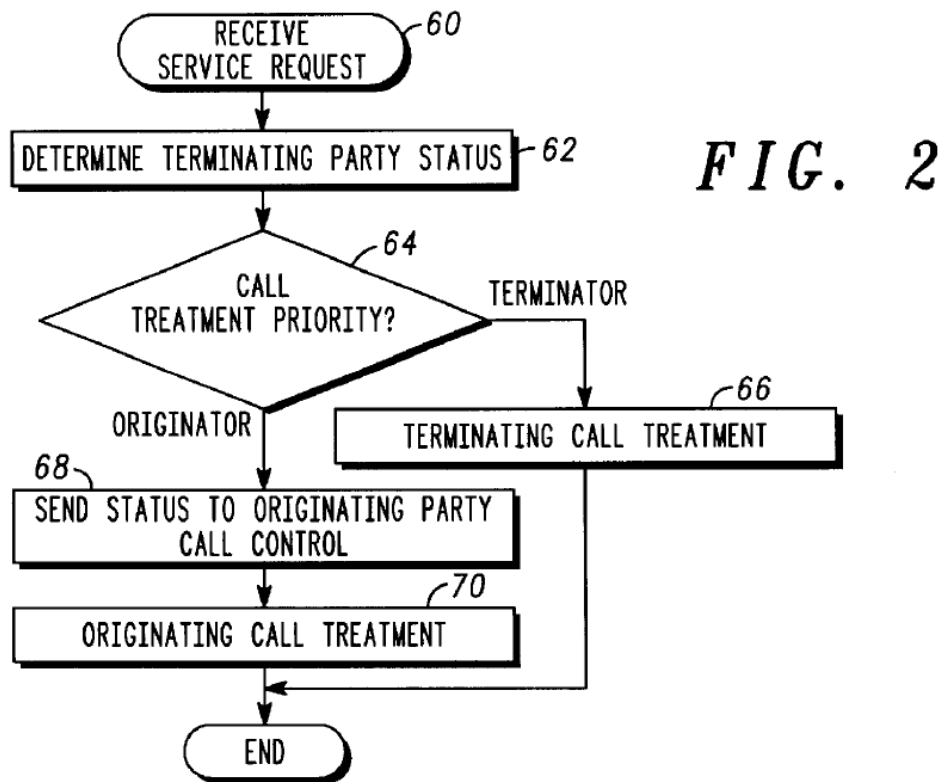
a call. “Cancel call” was used during the time period when the session had not been fully established while the “stop call” was when the session had already been established. *Id.*, 16.

33. Parlay X also allowed a user’s presence information to be determined. For example, presence information about the device’s included querying the current presence information and receiving a notification when the device changed its status. Ex-1047, 16. As shown in Figure 11 below, device’s status would indicate whether the device was “reachable, unreachable, and busy.” *Id.*, 18. This was applicable to a group of phones where phones 1 and 2 could indicate that they were busy while phone 3 was reachable. *Id.*



34. Parlay X extended to MMS and SMS messaging. The Chen Paper explained that “text and multimedia messaging revenue in the telecom market [was expected to exceed] \$4.3 billion in 2005, so an ambitious service provider should not omit the innovations spawned from SMS or MMS.” Ex-1047, 19. To meet this demand, Parlay X incorporated the ability to send and receive SMS or MMS messages, which included sending “ring tones and logos” like a jpeg, gif or png image format to another user. *Id.*, 19-20. The Chen Paper further describes sending an MMS message including “multimedia contents such as video, image or audio, which formed the basic services provided by most 3G operates.” *Id.*, 20.

35. Similarly, U.S. Patent No. 6,639,973 to Wheeler et al. (“Wheeler”) described a network-based system that would determine the called party’s status and take responsive action. Ex-1048. For example, Wheeler described that a call is initiated through “originating party call control” into the network, which then “determine[d] the terminating party status” by determining if the called party “has its line busy; is blocked or is attempting to place the incoming call on hold.” Ex-1048, 3:1-7. As shown in Figure 2, the called party’s device status is sent to the caller and the caller’s call control performs “call treatment logic” to perform the call treatment. *Id.*, 3:19-35.



36. U.S. Patent Publication No. 20030169865 to Oren (“Oren”, Ex-1049) also teaches a computer program embodied on a computer readable medium wherein each function is executed using a code segment to “complete a call when a called party has not answered the call.” Ex-1049 ¶¶1, 6. Specifically, Oren explained that significant system resources are devoted to complete a call when the called party does not answer the call because it causes the mobile service center or switch office of the network provider to “forward” or “divert” unanswered call to a voicemail or unified messaging intelligent peripheral. *Id.* ¶2. This “increases circuit switching within the network, reduces circuit quality, and prolongs the duration or ‘air-time’ of the connection” (*id.* ¶3), and results in increased operation and maintenance costs

for the network operator and additional expenses for the calling party and called party related to the “prolonged ‘air-time’ and the messaging service.” *Id.* ¶4.

37. Oren provides “flexible architecture” that allows the calling party’s terminal “to complete a call that has not been answered” “by completing an interaction between the parties, rather than just completing a voice call.” Ex-1049 ¶5. Oren’s functionality may be implemented on the calling party’s terminal. For example, Oren provides a method to detect a signal indicating that the called party has not answered the call and terminating the call. *Id.* ¶6. As shown in Figure 4, the user’s terminal detects an “‘announcement’ signal” (such as a message that the called party has selected to be sent to the calling party, *id.* ¶23), a “busy signal, a “no answer” signal or any other similar signal. *Id.* ¶¶25, 27. The terminal 400 (the calling party’s device) detects the signal indicating that the called party has not answered the call and prompts the calling party to compose a multimedia message to be delivered. *Id.* ¶27.

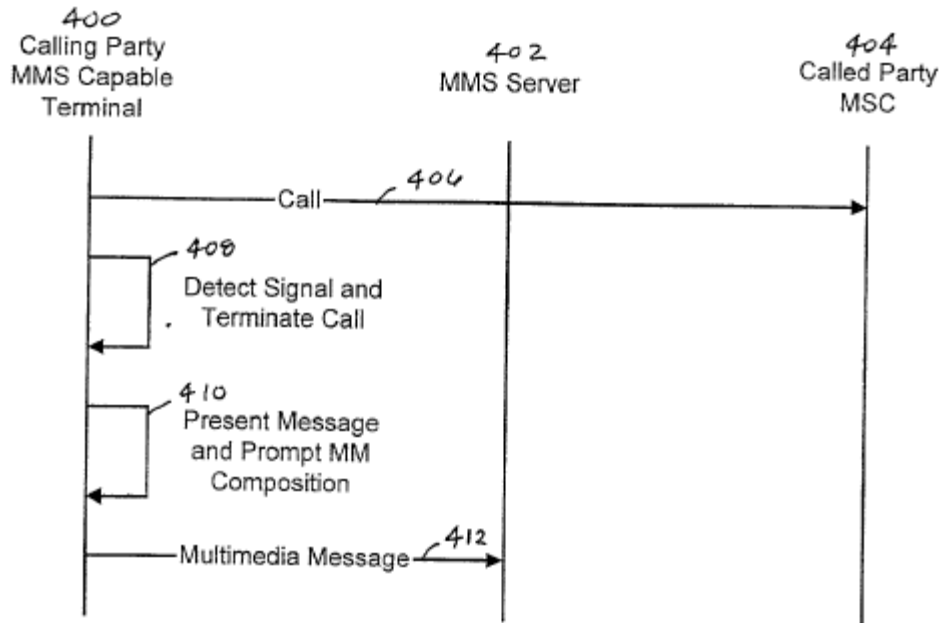


FIG. 4

38. Similarly, U.S. Patent Publication No. 20090142085 to Anderl et al. (“Anderl,” Ex-1050), explains a method (and computer program product comprising computer executable instructions, Ex-1050 ¶35) of completing a call made by a first party on a first network to second party on a second network. Ex-1050 ¶19; *see also id.* 37 (“The present invention can be implemented using standard wireless telephone devices capable of sending and receiving text SMS messages, voice SMS messages, MMS messages, multimodal messages and may be extended to other device types, and network technologies”). Specifically, the call is attempted but the first network detects “termination due to occurrence of a busy signal, an out of coverage signal, an unreachable signal, congestion of one or more of the first network 203 and the second network 204, an unanswered signal from the second party 205, and manual

call termination by the first party 201.” Ex-1050 ¶19. As shown in Figure 1, once the call attempt is terminated, the first network 203 communicates 102 availability of a messaging service (such as an SMS, voice SMS, multimedia messaging service, and a multimodal messaging service) to the first party 201. Ex-1050 ¶20. When the first party 201 accepts the messaging service, the first party 201 creates 104 a message to be sent to the second party 205 on the second network 204. Ex-1050 ¶25.

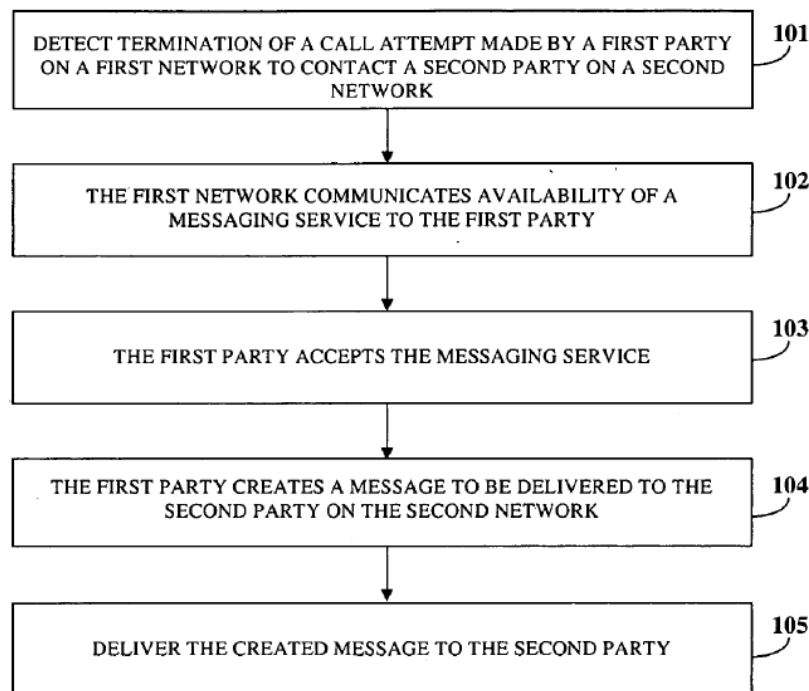


FIG. 1

39. Another method for a mobile phone to provide an unsuccessful call alert was disclosed in U.S. Patent No. 8,340,702 to Bychkov et al. (“Bychkov,” Ex-1051). Bychkov alerts a callee that a caller called them but was unable to reach them by

first monitoring the outgoing call from the caller and then determining whether the callee party is available. Ex-1051, 3:41-47. Bychkov describes that a callee “may not be available for a variety of reasons,” such as the callee’s phone being turned off or busy. Ex-1051, 3:44-49. Moreover, Bychkov identifies that the callee simply may not wish to answer the call or the callee’s phone may forward to a voice mail box, but the caller did not want to leave a voice message. Ex-1051, 3:49-52. Additionally, Bychkov determines that the callee is available based on call duration; e.g., a duration shorter than 10 seconds may be used as an indicator that the callee is not available. Ex-1051, 3:52-55.

40. U.S. Patent No. 8,229,409 to Bluvand et al. (“Bluvand”) further explains the need for call completion actions. Bluvand notes that more than 40% of the voice calls are not completed, thus presenting a need for alternatives for when a call that was not established. Ex-1052, 1:18-21. Bluvand provides a computer program product on the calling party’s device that “enable[s] calling party to better control the situation when a conversation session failed to be established (for example when the calling party is responded with waiting call signal, a no-answer signal, a redirection of the call to answering machine, a busy tone, a ‘wrong number’ announcement, a ‘network difficulty’ announcement, etc.”). *Id.*, 2:35-41. The calling party receives signal(s) indicating that the voice call could not be established and a list of alternative options for the calling party to establish other communication

channels with the called party, such as sending a SMS message, MMS message, or calling an alternative phone number. *Id.*, 3:10-31.

41. Thus, it was well known to have call completion applications that monitor the call as it progresses and determine that the called party was unavailable because they were busy, out of the network, unreachable, switched off, experiencing network issues, or simply that the call was unanswered.

B. Call completion actions, other than voicemail, were well known.

42. While voicemail is a commonly known method to notify the called party that the caller wanted to get in touch with them, POSITAs built off of that idea to make additional methods to notify the called party. For example, Parlay X would retry the failed number or use another way to contact the customer when determining that the call could not be completed. Ex-1047, 22.

43. Typical call control exercises by the network were call forwarding, call hold, and voicemail. Ex-1048, 1:61-63. However another example of completing the call when the call cannot be completed beyond the typical network call control was described in Wheeler. Ex-1048. Wheeler describes “call handling features” that are “triggered by an originating party for various supplementary services such as under conditions in which the terminating party is not reachable.” Ex-1048, 1:22-30. For example, Wheeler describes that when a connection from a caller to the “terminating subscriber” is unable to be completed, the system may use “pre-configure[d] options

for particular individuals” in the event that they are not reachable. Ex-1048, Figure 5, 120. Wheeler provides control logic set by the caller to control what happens to the attempted connection. *Id.*, 1:65-2:2. The control logic is essentially rules on what action should be taken when the called party is unavailable before “the terminating party’s subscriber services such as call forwarding or voicemail.” Ex-1048, 2:3-10, 2:34-39. Wheeler provides that when the termination party is not available the caller may establish a set of preferences as listed in Table 1, below. *Id.*; *see also id.*, 4:15-20 (explaining that the caller can set up specific actions such as sending a page to a particular number, sending an SMS message, or trying an alternate number when the called party is determined to be busy).

TABLE 1

1)	initiate autodialing from a prestored list of phone numbers or IP addresses;
2)	send a prestored message to a prestored address (phone number, IP address, alpha-numeric pager, etc.);
3)	send a voicemail message or instant message to one or more of the terminating parties 20 contact addresses; and
4)	periodically redial the phone number or reinitiate the IP address from a prestored list of numbers or addresses.

44. Wheeler shows an example execution of this system in Figure 5. Figure 5 depicts the call treatment rule set for a “Call forwarding example” where the caller has already configured options in tables for the rule set to follow when the called party is not reachable. For example, the caller calls Joe Smith but Joe Smith’s line is

busy so Wheeler's "call treatment rule set 33 selects Joe's mobile and appropriate corresponding address, 122" and sends the caller's default message of "urgent, please call." Ex-1048, 4:36-51. The message is then sent to Joe Smith's mobile phone. *Id.*, 4:51-59.

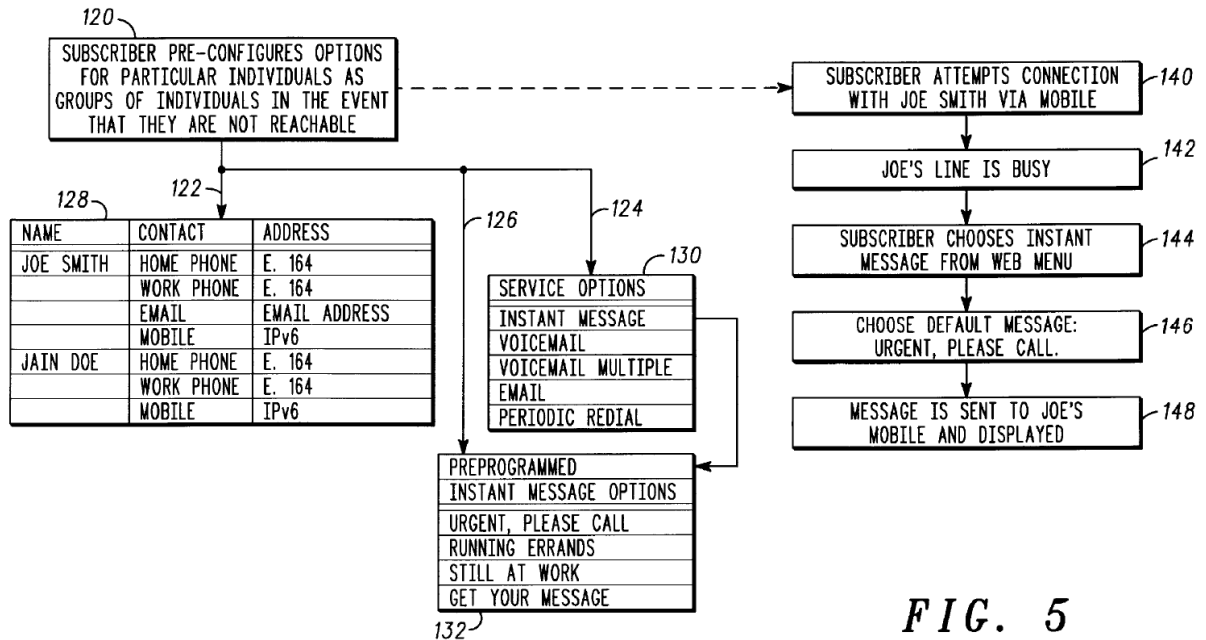


FIG. 5

45. Wheeler further explains that caller may invoke sending an instant message to a pager or another mobile phone, selecting voicemail or multiple voicemails, sending an email which may be prestored or selectively input, periodically redialing the called party until communication is achieved. Ex-1048, 4:60-5:4.

46. Wheeler describes the benefits of allowing the caller to set the rules as placing "control of the call processing is not in the hands of the originating party instead of typically leaving it in the hands of the terminating party" so that the caller

may be “able to take non-standard actions or alternatives based upon [their] circumstances.” Ex-1048, 2:56-64; *see also id.*, 5:4-10.

47. As I described earlier, Oren (Ex-1049) also provides a method of “completing a call when a called party has not answered the call from a calling party.” Abstract. Oren’s terminal 400 (the calling party’s device) detects the signal indicating that the called party has not answered the call and prompts the calling party to compose a multimedia message to be delivered. Ex-1049 ¶27. The message can be a default or a customized message, as determined by the calling party/terminal 400 preference. *Id.* The message may also include text, pictures, and audio files. *Id.* ¶28. If the calling party completes the multimedia message, the message is sent to an MMS server for delivery to the called party. *Id.* ¶28. The multimedia message may be set for deferred delivery on a later schedule, which may be selected by the network operator (e.g., low cost, low priority, quality of service parameters, service plan, etc.), by the called party (e.g., “I’ll be back from my honeymoon on . . .”, etc.) or by the calling party (e.g., a “birthday card”, etc.). Ex-1049 ¶28. Likewise, the multimedia message may be set for expedited or “rush” delivery on an earlier schedule. *Id.* The MMS server may send confirmation to the calling party, terminal 400, indicating successful receipt of the multimedia message. *Id.*

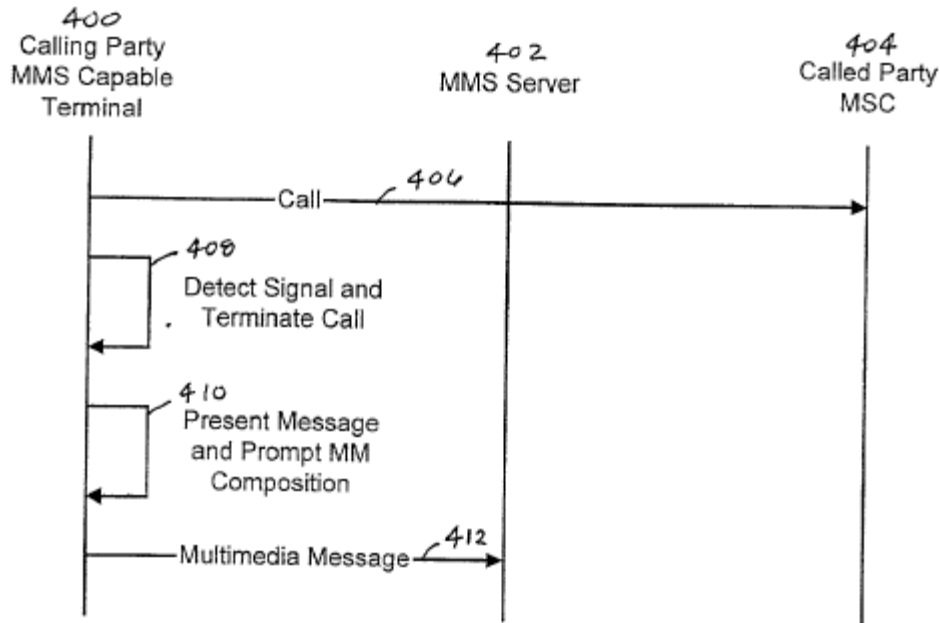


FIG. 4

48. Oren's method eliminates or minimizes the amount of resources devoted to completing a call when the called party does not answer. As I described earlier, Oren explains that significant resources are used for a mobile service center or switch office of the network provider to "forward" or "divert" unanswered call to a voicemail or unified messaging intelligent peripheral. Ex-1049 ¶2. This "increases circuit switching within the network, reduces circuit quality, and prolongs the duration or 'air-time' of the connection" (*id.* ¶3), and results in increased operation and maintenance costs for the network operator and additional expenses for the calling party and called party related to the "prolonged 'air-time' and the messaging service." *Id.* ¶4. Oren provides "flexible architecture" that allows the calling party's

terminal to complete a call that has not been answered by completing an interaction between the parties, rather than just completing a voice call.” Ex-1049 ¶5.

49. Another example of completing a call beyond call forwarding or diverting the call, is disclosed in Anderl. Anderl recognized that there was “a need for a messaging service to enable the caller to create and deliver messages including short messaging service (SMS) messages, voice SMS messages, multimedia messaging service (MMS) messages, and multimodal messages without making another call, or re-initiating the call to the recipient” when a call is “terminated due to a busy signal, an unanswered signal, an unreachable signal, [and, or] manual termination by the caller.” Ex-1050 ¶6. Once it is determined that a call attempt has been terminated, the caller is notified of the called party’s availability and is the caller “creates a message to be delivered to the second party on the second network” such as by “recording one or more of a text SMS message, a voice SMS message, a multimedia messaging service message, and a multimodal message on a communication device of the first party.” *Id.* ¶12; *see also id.* ¶25. “The text SMS message, the voice SMS message, the multimedia service message, and the multimodal message are delivered to the second party in a text short messaging service message format, a voice SMS message format, a multimedia messaging service message format, and a multimodal message format respectively.” *Id.* ¶13.

50. Bychkov also allows the caller to send a SMS or MMS to the callee after it is determined that callee is unavailable and that the call is unanswered. Specifically, Bychkov activates a “message editor on the caller's mobile phone” which “displays a pre-defined message” that the caller can edit. Ex-1051, 3:58-63, 4:9-14. For example, the caller may substitute a different phone number for the return call or provide their email address. *Id.*, 4:9-14.

51. Thus, it was well known in the art to complete an unanswered call by notifying the called party by SMS, MMS, video, audio, multimedia messages and multimodal messages, which POSITAs understood to be more advantageous than using call forwarding or diverting the call.

VIII. THE '227 PATENT AND ITS FILE HISTORY

52. The '227 Patent purports to describe a method and system for completing an incomplete call made by a calling party to a called party by executing a call completion action. Ex-1001, 2:5-16. The claimed method and system allegedly resolves “a long felt ... need” over conventional call completion solutions such as voicemail which are “dependent on the called party having enabled a voice message service for receipt of voice messages.” *Id.*, 1:44-50. Figure 1A illustrates the method of the invention:

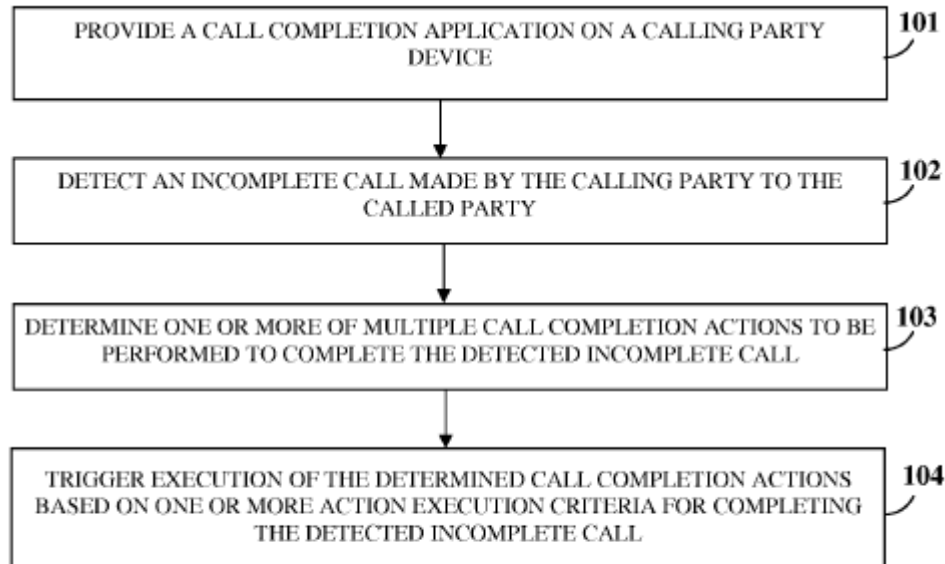


FIG. 1A

53. As the figure illustrates, a call completion application is provided on a calling party's device, Ex-1001, 6:14-20, which detects an "incomplete call" made to a called party (*id.*, 6:44-46). An "incomplete call" is one that is "not received or completed by the called party, or not connected to the called party device," because the called party's device is busy, out of coverage, unreachable, switched off, subject to network congestion or the call is of short duration or unanswered. Ex-1001, 6:46-59. In response to a detected incomplete call, the call completion application determines and executes a call completion action to "complete the detected incomplete call." *Id.*, 7:15-18. Call completion actions include setting a reminder to call again later or transmitting a recorded message, missed call alert or notification,

availability notification, or automated message to the called party's device. *Id.*, 7:18-

28. Simply put, the '227 Patent allows a user to send a message or reminder to the called party in response to an unanswered call.

54. The '227 Patent was filed on December 12, 2013 and claims priority to a provisional application filed on December 14, 2012. The examiner rejected independent Claims 1 and 17 as anticipated. Ex-1004, 116-131. In response, applicant amended Claims 1 and 17 to add limitations reciting specific call completion actions that can be received and performed by the call completion application. Applicant argued that the prior art did not "teach or suggest 'receiving one or more of a plurality of call completion actions to be performed by said call completion application ... wherein said plurality of call completion actions are selected according to one or more responses provided by said calling party.'" *Id.*, 139-151. Based on this argument, the examiner allowed all claims.

55. For the reasons I discuss below, the prior art teaches the claimed method for receiving one or more call completion actions by the user to be performed by a call completion application and renders the challenged claims unpatentable. Accordingly, it is my opinion that Claims 1-6, 8-15, and 17-19 of the '227 Patent are invalid as obvious over the prior art.

56. I understand from Counsel that all of the prior art references I discuss in my Declaration are prior art.

IX. CLAIM CONSTRUCTION

57. For the purposes of my Declaration, I do not believe that any term requires explicit construction to resolve the issues I present in my Declaration, except for the terms below. Terms not expressly construed should be given their plain and ordinary meaning. I understand from Counsel that various terms were construed or agreed upon by the parties (Multimodal and Guangdong OPPO Mobile Telecommunications Corp.) in the *Oppo* district court litigation where Patent Owner asserted the '227 Patent against an unrelated party.⁵ I apply those constructions in my analysis, except where I note below.

A. “call completion application” (Claims 1, 5, 6, 9, 10, 11, and 15)

58. I understand from Counsel that this claim term was construed by the district court in the *Oppo* litigation—pursuant to an agreement by the parties—to mean “a software module that completes an incomplete call made by a calling party without any dependence on the underlying network or called party’s device.” Ex-1021, 47. I disagree with this construction. Instead, in my opinion, the claim should be given its plain and ordinary meaning in view of the specification, which is: “a software module that completes an incomplete call made by a calling party that is independent of preconfigured call completion service, such as voicemail, that is on the called device.”

⁵ I understand from Counsel that the *Oppo* litigation was settled. Ex-1046.

1. The specification supports the plain and ordinary meaning

59. The '227 Patent states that the call completion application is “independent *of a preconfigured call completion service on the called party’s network or a called party device.*” Ex-1001, 1:53-62. A “call completion service” is, for example, a voicemail service on the called device. *See, e.g., id.*, 1:24-43. Therefore, the specification makes clear that the call completion application is “independent of a preconfigured call completion service” like a voicemail on the called party’s device or network.

60. The *Oppo* construction is wrong because the specification makes clear that the claimed call completion application is a “software module that completes an incomplete call made by a calling party” by “transmit[ting] a message of the calling party to the called party device via the network based on one or more action execution criteria.” Ex-1001, 3:56-59. Moreover, the '227 Patent establishes a “communications link between the calling party” on a “first network component” and a “second network component” of the called party. *Id.*, 6:18-30. In other words, the call completion application clearly depends, at least in part, on the called party’s network connection to work.

2. The dependent claims contradict the *Oppo* construction

61. In my opinion, the dependent claims themselves do not support the narrow construction that was agreed to by the parties in the *Oppo* litigation. Claims

1 and 11 require a call completion application that detects an “incomplete call” based on the “occurrence of one or more of a plurality of events.” Claims 2 and 12 narrow the events to details of the called party device being busy, out of a coverage area, unreachable, switched off, subject to network congestion, or the call unanswered. Claims 5 and 15 recite action execution criteria based on the “network options” provided to the called party device.

62. Dependent Claims 2, 5, 12, and 15 thus explicitly depend at some level “on the underlying network or called party’s device.” Independent Claims 1 and 11 thus should not exclude “the underlying network or called party’s device” because that would cause Claims 2, 5, 12, and 15 to contradict, rather than narrow the independent claims from which they depend.

3. The agreed-upon construction makes no sense

63. In addition to not being tethered to the specification or reconcilable with the dependent claims, the *Oppo* agreed-upon construction simply makes no sense. Making the call completion service operate “without any dependence on the underlying network or called party’s device” misunderstands how call-related applications work. Calls are made over a network and to called devices and, therefore, contrary to the *Oppo* construction, applications for making calls must at least have some dependency on network and messaging protocols. Accordingly, the

prior agreement reached by the *Oppo* litigation parties does not and cannot reflect the proper understanding of this term.

64. On the other hand, consistent with the intrinsic evidence, the plain meaning of the claimed “call completion application” should be understood as “a software module that completes an incomplete call made by a calling party that is independent of a preconfigured call completion service, such as voicemail, that is on the called device.” I apply this construction for purposes of this Declaration and explain herein that the prior art in the Declaration invalidates the challenged claims under either the *Oppo* construction or my construction.

B. “one of text data, audio data, video data, audiovisual data, image data, multimedia data, message data, and any combination thereof” (Claims 8, 17)

65. I understand from Counsel that this claim term was construed by the district court in the *Oppo* litigation to mean “text data, audio data, video data, audiovisual data, image data, multimedia data, message data, or any combination thereof.” Ex-1021, 27. I apply this construction for purposes of this Declaration and explain herein that the prior art in the Declaration invalidates the challenged claims under this construction, which is the plain and ordinary meaning of the claim.

X. OVERVIEW OF THE APPLIED PRIOR ART REFERENCES

A. Prikowitsch (Ex-1011)

66. Prikowitsch, titled “Method For Call Control When Setting Up A Communication Link,” was published as WO2010/139478 on December 9, 2010. Ex-1011, Cover.⁶ I have been informed that Prikowitsch is §102(b) prior art.

67. Prikowitsch discloses a method for call control that allows a calling party to customize and define actions to react to reported call statuses, Ex-1011, 1:22-2:2, such as when the called party is busy or not logged in to the network. *Id.*, 4:13-15. Specifically, Prikowitsch discloses a call application on the calling party device (*id.*, 4:30-32), which monitors the call from a calling party (10, blue) to a called party (40, orange) and processes events based on the detected and “reported status information.” *Id.*, 6:33-7:2, Fig. 4 (annotated). The reported status (46, red) may include “the call status,” “login status of the called terminal device,” “availability of the called terminal device and/or the network status.” *Id.*, 3:5-15, Cl. 2 (“The method according to claim 1, characterized in that the status reported back to the first terminal device from the network is the call status and/or login status of the called terminal device and/or the availability of the called terminal device and/or the network status, particularly the availability of the network or resource

⁶ The Prikowitsch WIPO publication does not include line numbers. For ease of reference, Petitioner has added line numbers to the reference and cites to the page and appropriate line number.

availability within the communication network(s).”). The call application evaluates the reported status from the network (*id.*, 2:13-18) and allows the user to implement and control the desired responsive action. *Id.*, 7:4-7. The application is therefore a user interface and contains “logic for how to react to respective events.” *Id.*, 7:16-19.

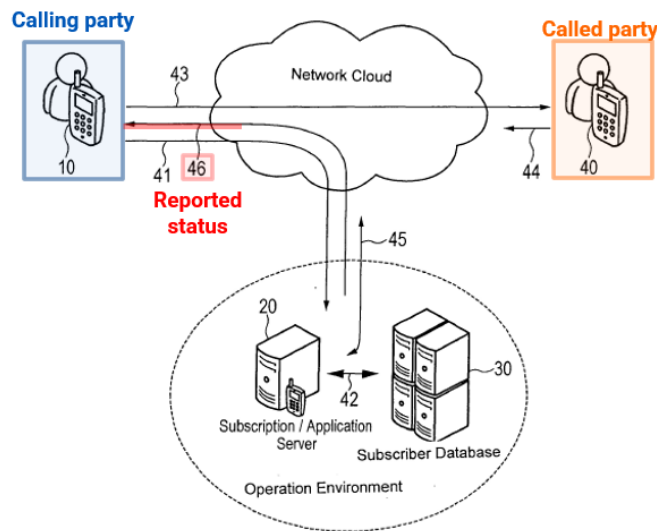


Fig. 4

68. The user’s responsive actions include individual announcements, implementing diversion or call forwarding options, and sending ready-made text and voice messages in response to the called party device being unavailable or busy. Ex-1011, 5:26-33 (“By way of example, these actions prescribed by the A subscriber may be individual announcements, tones or music, or may be the implementation of diversions or call forwarding operations. In addition, it is possible to send ready-made text messages and/or multimedia messages and/or voice messages, particularly

including individual recorded voice messages and the like. It is also possible to transmit appropriate announcements, and, by way of example, also to make calendar entries for a requested return call if the B subscriber is unavailable.”), 7:4-15 (same).

B. Roujinsky (Ex-1012)

69. Roujinsky, titled “Method and System for Automatic Call Completion of Unanswered Calls Made to or From a Telecommunication Device Method and System for Obtaining Feedback from at Least One Recipient Via a Telecommunication Network,” was published as WO2008/050325 on May 2, 2008. Ex-1012, Cover. I have been informed that Roujinsky is §102(b) prior art.

70. Roujinsky discloses “call completion” software stored on the calling party device that compiles a list of unanswered calls, which trigger “a user-defined action so as to enable communication” between a calling party and called party. Ex-1012, 10:28-11:1. Specifically, Roujinsky’s software reads a new record added to the calling party device’s call log when the device attempts to make a call, analyzes the record parameters, and registers an “uncompleted call” (*id.*, 16:9-14) if, for example, the called party’s device is busy, out of coverage, turned off, experiencing network problems, or the call is of short duration. Ex-1012, 1:6-16; *see also id.*, Fig 3 (annotated in green).

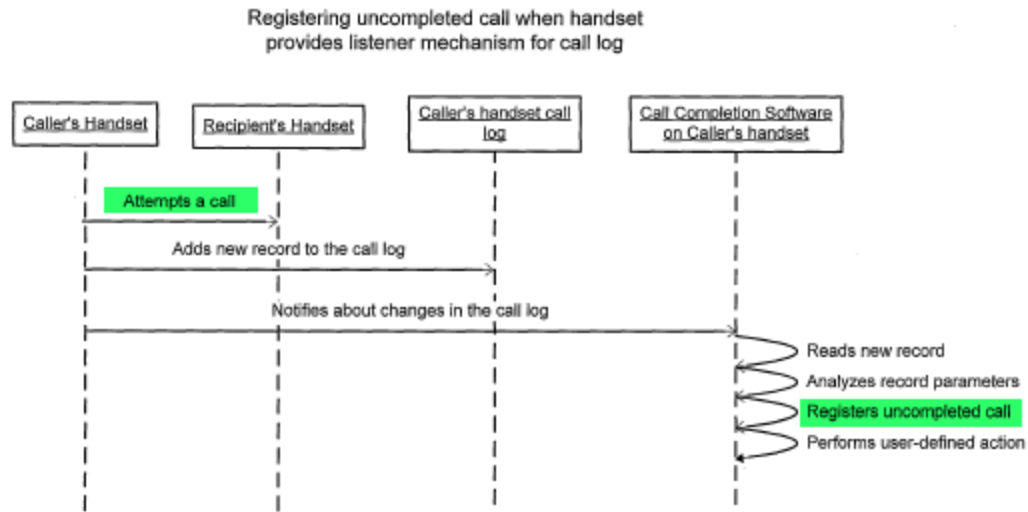


FIG. 3

71. Roujinsky's software performs a user-defined action based on the registration of "unanswered calls." Ex-1012, 12:13-14. User-defined actions include automatically issuing a predefined message such as an SMS, WAP Push, MMS, instant message, or voice message to the called party about the call attempt with an invitation to return the call, automatically prompting the caller to record and send a customized message, or setting an automatic reminder to call the recipient again. Ex-1012, 12:13-30.

C. Sar-Shalom (Ex-1013)

72. Sar-Shalom, titled "Automatic Messaging Client Launcher for a Communication Device," was published as US2004/0267884 on December 30, 2004. Ex-1013, Cover. I have been informed that Sar-Shalom is §102(b) prior art.

73. Sar-Shalom discloses a messaging client that is automatically launched on a calling party's device to perform "selected tasks or processes" when the called

party's device is unavailable. Ex-1013, Abstract, ¶25. For example, Sar-Shalom discloses with Figure 3 that a determination is made whether a call is answered by the called party. *Id.* ¶53, Fig. 3 (S310, yellow). The called party is considered “unavailable” such that the call cannot be completed if “a busy signal is detected,” “the call is not answered after a predetermined number of rings,” “the call connects to a voicemail box,” or the called party device is offline. *Id.* ¶38.

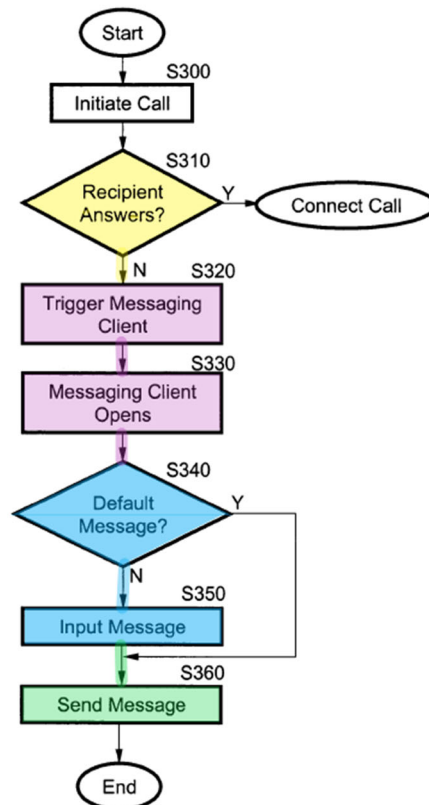


FIG. 3

74. If the call is not answered, the messaging client is triggered to launch (S320, purple) and open on the calling party's device (S330, purple) (Ex-1013 ¶53), which allows the user to choose a response action. *Id.* ¶45. The application

determines “whether a default message is available” (S340, blue) or allows the calling party to input a message (S350, blue), which is then sent to the called party (S360, green). *Id.* When the calling party inputs the message, the calling party may define the message that is sent (*id.* ¶42) as an “SMS, EMS, or MMS message” comprising “text, audio clips, video clips, still images, multimedia presentation and voice.” *Id.* ¶40.

XI. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

75. Claims 1-6, 8-15, and 17-19 of the '227 Patent are substantially similar. Like claim elements are discussed together in the below element-by-element analysis. *See* Ex-1020 (summary table). To the extent that there are meaningful differences, I identify and discuss such differences individually. As I explain below, the challenged claims are disclosed and/or taught by the prior art.

A. Ground 1: Claims 1, 2, 5, 8-12, 15, and 17-19 are Obvious Over Prikowitsch

1. Independent Claims 1 and 11

- a. Elements 1[pre]/11[pre]: A computer implemented method/[A computer system] for completing an incomplete call made by a calling party to a called party, comprising**

76. Prikowitsch teaches the preambles of Claims 1 and 11. Prikowitsch teaches a method for call control that allows a calling party to customize and define actions to react to reported call statuses, Ex-1011, 1:22-2:2, such as when the called

party is busy, not logged in to the network (offline), or the line is busy. *Id.*, 4:13-15.

Specifically, a call application on the calling party device (*id.*, 4:30-32) allows the calling party to complete an incomplete call by, for example, sending a ready-made text or voice message to the called party device. *Id.*, 7:4-15. Prikowitsch further discloses “a computer program product” that executes the call application on the calling party’s device. Ex-1011, 2:20-23, Cl. 11 (“A computer program product which performs the method according to one of the preceding claims when it is executed on a terminal device, particularly a mobile radio terminal device, and/or on a server associated with the communication network, particularly a mobile radio network, or when it is partly executed on a terminal device, in particular a mobile radio terminal device, and partly on a server associated with the communication network, particularly a mobile radio network.”).

b. Element 1[a]: providing a call completion application executable by at least one processor on a calling party device;

Element 11[a]: a non-transitory computer readable storage medium configured to store a call completion application on a calling party device;

Element 11[b]: at least one processor communicatively coupled to said non-transitory computer readable storage medium, said at least one processor configured to execute said call completion application on said calling party device; and

77. Prikowitsch teaches Elements 1[a] and 11[a]-11[b]. Prikowitsch teaches a call competition application executable by at least one processor on a

calling party device. Specifically, Prikowitsch's method for call control employs an application (i.e. software) executed on the calling party device. Ex-1011, 4:30-32. The application monitors the call from a calling party (10, blue) to a called party (40, orange) and processes events based on reported "status information." *Id.*, 6:33-7:2, 7:26-27, Fig. 4 (annotated). The status (46, red) may include "the call status," "login status of the called terminal device," "availability of the called terminal device[,] and/or the network status." *Id.*, 3:5-15, Cl. 2 ("The method according to claim 1, characterized in that the status reported back to the first terminal device from the network is the call status and/or login status of the called terminal device and/or the availability of the called terminal device and/or the network status, particularly the availability of the network or resource availability within the communication network(s)."). The call application evaluates the reported status (Ex-1011, 2:13-18) and allows the user to implement a desired responsive action such as implementing diversion or call forwarding options, sending individual announcements, or ready-made text or voice messages to the called party. *Id.*, 7:4-15.

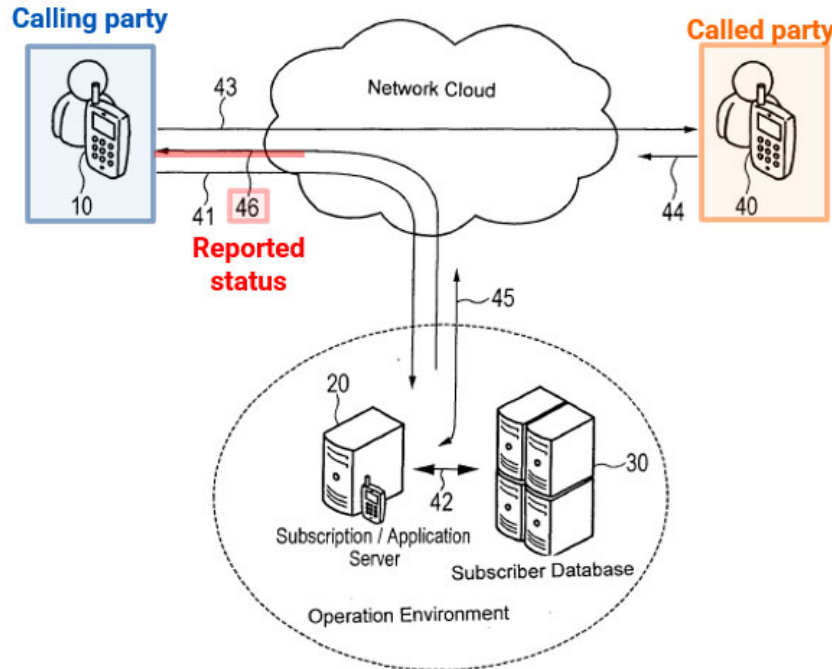


Fig. 4

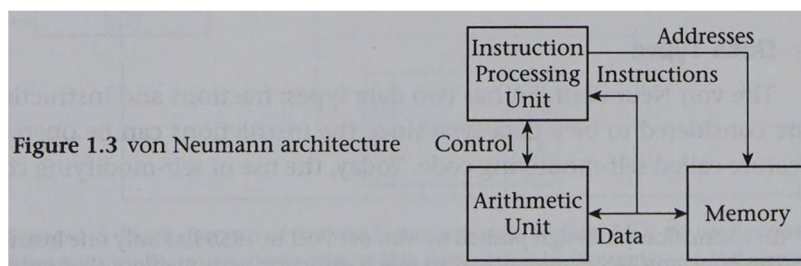
78. Prikowitsch's application software uses a processor to process these actions implemented by the calling party that "complete" an unanswered call, independent of a preconfigured call completion service, such as voicemail. For example, the application software implements actions such as sending a ready-made text message to the called party, independent of a call completion service, such as voicemail. Ex-1011, 7:4-15. This is so because a text message is not sent over a voice channel, it is instead sent with a distinct service, SMS, that does not depend on voice calls or a call completion service such as voicemail to operate. Moreover, the

application is executable by a processor because the application contains “logic for how to react to respective events.” *Id.*, 7:16-19.

79. Prikowitsch further teaches a non-transitory computer readable storage medium communicatively coupled to the processor and configured to store the call completion application on the calling party device. Specifically, Prikowitsch discloses a “computer program product” that executes the call completion application method on the calling party’s mobile device. Ex-1011, 2:20-23 (“Preferably, the first calling terminal device is therefore a mobile radio terminal device within a mobile radio network.”), 4:30-33 (“[T]he application can be executed on the first calling terminal device of the A subscriber and/or on a server associated with the first communication network.”), Cl. 11 (“A computer program product which performs the method according to one of the preceding claims when it is executed on a terminal device, particularly a mobile radio terminal device, and/or on a server associated with the communication network, particularly a mobile radio network, or when it is partly executed on a terminal device, in particular a mobile radio terminal device, and partly on a server associated with the communication network, particularly a mobile radio network.”); *see also id.* 5:19-24 (“An A subscriber installs an appropriate piece of application software on his terminal device. The settings implemented on the terminal device provide the A subscriber with the option of implementing and prescribing certain actions on the

basis of the status reports and events reported back in the course of the call control.”).

In order for the computer program product to execute the application, a memory would store the application, such as Prikowitsch’s “memory unit associated with said terminal device.” *Id.*, 3:34-4:2. The memory unit would include non-transitory, computer readable program memory accessible to the processor because a POSITA would have understood that software application on the calling party’s device is stored in non-volatile memory such as a Flash memory that is communicatively coupled to the processor so that it can be accessed for execution. This is well known as disclosed by the “classical and enduring” von Neumann architecture that goes back to 1946 and “has been the unifying concept in all computer designs since 1950.” Ex-1053, 0014-0019 (textbook pages 2-6). The von Neumann architecture describes an electronic device design with an instruction processing unit (such as a CPU), an arithmetic logic unit, and memory that stores data and instructions. As illustrated in the following figure, these three elements all worked closely together. Therefore it is well known that the basic computer architecture would have processors with memory to run a computer stored-program. *Id.*



80. For example, as I explained above in paragraph 38 in Section VI.A, Anderl discloses a computer program product comprising computer executable instructions (Ex-1050 ¶35) of completing a call made by a first party on a first network to second party on a second network. Ex-1050 ¶19. Anderl notes that it would be “readily apparent that the various methods and algorithms” would be implemented in a computer readable medium computer readable medium appropriately programmed for general purpose computers and computing devices. *Id.* ¶35. Specifically, “a processor, for e.g., one or more microprocessors will receive instructions from a memory or like device, and execute those instructions, thereby performing one or more processes defined by those instructions.” *Id.* A POSITA would have understood that Prikowitsch’s computer program product would be stored in a memory so that it can be performed by the processor on the calling party’ device.

- c. **Element 1[b]: detecting said incomplete call made by said calling party to said called party, by said call completion application on said calling party device,**

Element 1[c]: wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events;

Element 11[c]: said call completion application on said calling party device, comprising:

Element 11[d]: a call detection module configured to detect said incomplete call made by said calling party to said called party,

81. Prikowitsch teaches Elements 1[b], 1[c], 11[c], and 11[d]. Prikowitsch teaches detecting an incomplete call made by the calling party to the called party by a call completion application. Specifically, Prikowitsch teaches a call completion application which monitors a call from a calling party to a called party based on “reported status information.” Ex-1011, 6:33-7:2. The reported status includes “the call status,” “login status of the called terminal device,” “availability of the called terminal device and/or the network status.” *Id.*, 3:5-15, Cl. 2. A POSITA would have understood that “call status” includes whether the call is incomplete due to the occurrence of one or more events, such as whether the call was answered and if the called party is busy (Ex-1011, 4:15 (“the line is busy”), Cl. 2 (listing an example of an “event” as the “called party being busy”)); “login status of the called terminal device,” (Ex-1011, 3:11-15, 4:14-15) which a POSITA would have understood to include whether the device is reachable; “availability of the called terminal device,”

which a POSITA would have understood to include whether the device is unreachable (Ex-1011, 5:33 (“subscriber is unavailable”)); “and/or the network status,” which a POSITA would have understood to include whether the device is out of coverage. Ex-1011, 2:13-18, 3:5-8, 13:5-9. A POSITA would have understood that the call to a device could not be completed when that device is unavailable to the telecommunications network because the device is unreachable and cannot connect to be answered, as I explained above in Section VII.A (paragraphs 32-41). Specifically, it was well known to detect the reason for which the called party is unavailable beyond just that the call did not connect. Thus Prikowitsch detects an incomplete call made by said calling party to said called party when the line is busy, the device is unreachable, unavailable, or out of coverage.

82. Prikowitsch’s call application on the calling party device performs the “detecting” step. The application on the calling party device (Ex-1011, 4:30-32), detects that the incomplete call does not connect to a called party device due to the occurrence of one or more of a plurality of events (e.g., the called party device being busy or unavailable), based on the reported status information. Ex-1011, 5:19-33, 7:4-15; *see also id.*, 13:5-9 (“event 44 ... is signalled 46 to the application on the terminal device 10 for further evaluation”), 2:29-3:3 (“This is preferably realized by a software”). The ’227 Patent’s “call detection module” is software since it is part of the “call completion application”. Ex-1001, 14:21-24. A POSITA would have

understood that Prikowitsch's call application includes "a call detection module," i.e. software to detect the incomplete call because it is a computer program product, made of software instructions executable by the processor on the user's terminal device. A POSITA would have understood that the module is software to carry out the function because, like Oren, there are code segments to execute each function of the call completion method. *See, e.g.*, Ex-1049 ¶6, Claims 15, 22-24.

- d. Element 1[d(i)]: receiving one or more of a plurality of call completion actions to be performed by said call completion application on said calling party device to complete said detected incomplete call,**

Element 11[e(i)]: an action determination module configured to receive one or more of a plurality of call completion actions to be performed to complete said detected incomplete call,

83. Prikowitsch teaches Elements 1[d(i)] and 11[e(i)]. Prikowitsch teaches receiving one or more of a plurality of call completion actions to be performed to complete the detected incomplete call. Specifically, Prikowitsch's calling party defines individualized outgoing actions that allow the call completion application "to react to any reported call statuses" as the user desires (Ex-1011, 1:22-2:2), if the called party is unavailable. *Id.*, 5:26-34. For example, the calling party may prescribe the following actions: "individual announcements, tones or music," "diversions and call forwarding operations," "send ready-made" text, multimedia, and/or voice messages (including individual recorded voice messages), "generate and transmit

appropriate announcements,” and “make calendar entries” to return calls. *Id.*, 5:26-34; *see also id.*, 4:26-27 (application “configure[s] the actions to be performed on the basis of the status report”). Prikowitsch thus provides for multiple call completion actions to be performed to complete the call when an incomplete call is detected. These user-defined actions are implemented when the called party device is, for example, unavailable or busy, such that the call cannot be completed. Ex-1011, 5:21-33, 7:4-15. A POSITA would have understood that the calling party could predefine call completion actions on their device such that the “receiving” step in Claim 1 would be performed before the “detecting” step as an obvious matter of design choice. Indeed, dependent Claims 6-7 of the ’227 Patent recite an embodiment wherein the actions are based on criteria previously configured by the calling party (such as call settings on the calling party device). Ex-1001, Cls. 6-7; *see also id.*, 7:61-67 (“[A] calling party can configure sending a missed call alert to a called party device, each time a call to a specific number of the called party results in being unanswered.”). Moreover, as I explained above in Section VII.A (paragraphs 43-46), Wheeler provides that when the termination party is not available the caller may establish a set of preferences. Ex-1048, 2:3-10, 2:34-39; *see also id.*, 4:15-20 (explaining that the caller can set up specific actions such as sending a page to a particular number, sending an SMS message or try an alternate number when the called party is determined to be busy), 4:36-59 (providing an example of

using user-defined rule set to send a message to an unavailable called party (Joe Smith)).

84. The call completion application on the calling party's terminal device "contains the logic for how to react to respective events." Ex-1011, 6:1-3. A POSITA would have understood that Prikowitsch's call completion application thus includes an "action determination module," i.e., software to allow the user to define actions (logic for how to react) to complete the call because it is a computer program product, made up of software instructions executable by the processor on the user's terminal device. A POSITA would have understood that that the module is software to carry out the function because, like Oren, there are code segments to execute each function of the call completion method. *See, e.g.*, Ex-1049 ¶¶6, Claims 15, 22-24. As I described above in Section VII.A (paragraphs 47-48), Oren's terminal 400 (the calling party's device) detects the signal indicating that the called party has not answered the call and prompts the calling party to compose a multimedia message to be delivered using code segments. Ex-1049 ¶¶6, 27.

- e. **Elements 1[d(ii)]/11[e(ii)]: wherein said plurality of call completion actions are selected according to one or more responses provided by said calling party through a graphical user interface of said calling party device,**

85. Prikowitsch teaches Elements 1[d(ii)]/11[e(ii)]. Prikowitsch further explains that the calling party provides the call completion actions through responses via a graphical interface on their device. Specifically, the user selects the appropriate

responsive action via a “graphical user interface with appropriate menu control.” Ex-1011, 4:17-25 *see also id.* Claim 5. For example, the caller may “select[] a profile of announcement” for the scenario where ““subscriber [is] busy,”” and a responsive action to “return call reminder if busy.” Ex-1011, 12:24-28. The call completion actions are thus selected according to the user’s response on the calling party device’s graphical user interface.

- f. **Elements 1[d(iii)]/11[e(iii)]: wherein said call completion actions comprise [1] setting a reminder to call back said called party at a configurable time, [2] recording media data on said calling party device, [3] transmitting said media data to a called party device, [4] transmitting a missed call alert to said called party device, [5] transmitting a notification of said detected incomplete call to said called party device, [6] transmitting a notification on availability of said called party, [7] transmitting said media data to a social networking platform, [8] transmitting an automated message requesting said called party to call back said calling party when available, and [9] any combination thereof; and**

86. Prikowitsch teaches Elements 1[d(iii)]/11[e(iii)],⁷ specifically at least actions labeled [1]-[3]. These claim elements list the possible call completion actions but, according to Element 1[d(i)]/11[e(i)], the call completion application receives “*one or more of* a plurality of call completion actions” or “any combination thereof.” Thus, if the call completion action receives at least one of the call completion actions

⁷ I have numbered the call completion actions for ease of reference.

in Elements 1[d(iii)]/11[e(iii)], then the claim language is met. In other words, it is only necessary to receive one of these call completion actions to satisfy this claim.

87. Prikowitsch teaches that the call completion actions comprises at least “[1] setting a reminder to call back said called party at a configurable time.” Prikowitsch teaches that the user may set a reminder to call the called party at a configurable time, by making “calendar entries relating to return calls,” Ex-1011, 7:9-15, or selecting a “return call reminder if busy” action on their user interface. *Id.*, 12:24-28. Specifically, an “appointment” may be created with a “subject line” stating ““return call to No xxx required.”” *Id.*, 13:14-15. Prikowitsch’s calendar appointment is the claimed call completion action of setting a call reminder to the called party. That is, it is a reminder to complete the call that was unanswered because the called party was busy.

88. Prikowitsch also teaches call completion actions comprising “[2] recording media data on said calling party device” and “[3] transmitting said media data to a called party device.” Specifically, the user may configure the call completion application to transmit “ready-made text messages and/or multimedia messages and/or voice messages,” including “individual recorded voice messages” to the called party. Ex-1011, 5:28-34, 7:9-14. Prikowitsch’s multimedia or voice messages are the claimed recording and transmitting media data call completion

actions because they are individual recorded text or voice messages sent to the called party to complete the unanswered call.

89. As I described above in Section VII.B (paragraphs 42-51), it was common and well known to implement call completion actions beyond the use of voicemail. Specifically, I explained how POSITAs acknowledged the benefits of allowing the calling party to control what happens to the attempted call so as to allow the calling party to take non-standard actions or alternatives based upon [their] circumstances.” Ex-1048, 2:56-64; *see also id.*, 5:4-10; Ex-1049 ¶5 (providing “flexible architecture” that allows the calling party’s terminal to complete a call that has not been answered “by completing an interaction between the parties, rather than just completing a voice call”), ¶¶27-28 (prompting the calling party to compose a multimedia message including text, pictures, and audio files); Ex-1050 ¶6, 12 (once it is determined that a call attempt has been terminated, the caller is notified of the called party’s availability and is the caller “creates a message to be delivered to the second party on the second network” such as by “recording one or more of a text SMS message, a voice SMS message, a multimedia messaging service message, and a multimodal message on a communication device of the first party.”); Ex-1051, 3:58-63, 4:9-14 (activates a “message editor on the caller's mobile phone” which “displays a pre-defined message” that the caller can edit).

- g. Element 1[e]: triggering execution of said received one or more of said call completion actions by said call completion application on said calling party device based on one or more of action execution criteria for said completion of detected incomplete call.**

Element 11[f]: an action execution module configured to trigger execution of said received one or more of said call completion actions based on one or more of action execution criteria one or more responses selected by said calling party for said completion of said detected incomplete call.

90. Prikowitsch teaches Elements 1[e] and 11[f]. Prikowitsch triggers execution of the call completion actions by the application on the calling party device based on action execution criteria for completion of the detected incomplete call. Specifically, Prikowitsch's call completion application "contains the logic for how to react to respective events," Ex-1011, 6:1-3, that triggers the caller's responsive actions (i.e., "call completion actions") to complete an incomplete call to be executed, by sending "individual announcements," implementing "diversions and/or forwarding operations," sending "ready-made SMS/MMS/voice messages," generating "appropriate announcements," and/or making "calendar entries relating to return calls." *Id.*, 7:9-15.

91. The actions are executed based on action execution criteria because the user prescribes the actions to be executed depending on the reported status information. Ex-1011, Cl. 5. For example, when the reported status indicates that the called party is busy, the calling party's defined actions or profile (i.e., configurable

criteria) linked to the “busy” event (e.g., call forwarding operations) are executed to complete the call. Ex-1011, 13:5-15. As in the ’227 Patent, the prescribed action such as implementing call forwarding when the reported status indicates that the called party is “busy,” is an example of action execution criteria because it sets out what the application will do to execute the selected the call completion action. *See* Ex-1001, 8:49-53 (“The action execution criteria are a set of factors that determine an appropriate method for executing a call completion action, for example, transmitting a message to the called party device.”). In Prikowitsch, the application thus triggers execution of the call completion actions based on a prescribed action (i.e., action execution criteria) similar to the way this feature is performed in the challenged patent.

92. A POSITA would have understood that Prikowitsch’s call completion application includes an “action execution module,” i.e., software that triggers execution of the call completion actions on the caller’s device because it is a computer program product, made of software instructions executable by the processor on the user’s terminal device in order to react to the reported call status. A POSITA would have understood that that the module is software to carry out the function because, like Oren, there are code segments to execute each function of the call completion method. *See, e.g.,* Ex-1049 ¶6, Claims 15, 22-24. As I described above in Section VII.A (paragraphs 32-41), Oren’s terminal 400 (the calling party’s

device) detects the signal indicating that the called party has not answered the call and prompts the calling party to compose a multimedia message to be delivered using code segments. Ex-1049 ¶¶6, 27.

2. Dependent Claims 2, 5, 8-10, 12, 15, and 17-19

- a. Element 12[a]: The computer system of claim 11, wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events,**

93. Prikowitsch teaches Element 12[a]. Prikowitsch's incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events for the same reasons explained in Section XI.A.1.c (Element 1[c]).

- b. Claim 2/Element 12[b]: [The computer implemented method of claim 1, / The computer system of claim 11,] wherein said events comprise said called party being busy, said called party device being in an out of coverage area, said called party device being unreachable, said called party device being switched off, network congestion, and said call not being answered by said called party.**

94. Prikowitsch teaches Claim 2 and Element 12[b], which list the “one or more plurality of events” in Elements 1[c]/12[a] that cause a call to not be connected to a called party device. Thus, if at least one of the events in Claim 2/Element 12[b] occurs, these limitations are met.

95. Prikowitsch teaches that the reported status may include “call status,” which a POSITA would have understood to include whether the call was answered

and if the called party is busy (Ex-1011, 4:15 (“the line is busy”)); “login status of the called terminal device,” (Ex-1011, 3:11-15, 4:14-15), which a POSITA would have understood to include whether the device is switched off; “availability of the called terminal device,” which a POSITA would have understood to include whether the device is unreachable (Ex-1011, 5:33 (“subscriber is unavailable”)); “and/or the network status,” which a POSITA would have understood to include whether the device is out of coverage; or simply not answered. Ex-1011, 2:13-18, 3:5-8, 13:5-9. As I explained above in Section VII.A (paragraphs 32-41), a called party may be unavailable for a variety of reasons, such as the called party’s phone being turned off or busy (Ex-1051, 3:44-49), that the called party may not wish to answer the call or the called party’s phone may forward to a voice mail box, but the caller did not want to leave a voice message. Ex-1051, 3:49-52. Additionally, it was well known to determine that the called party was unavailable based on call duration; e.g., a duration shorter than 10 seconds may be used as an indicator that the callee is not available. Ex-1051, 3:52-55. Thus, a POSITA would have been motivated to include these events in order to give the calling party additional control on determining the best way to communicate with the called party, i.e., complete the call.

c. Claims 5 and 15

- (1) Elements 5[a]/15[a]: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said action execution criteria comprise configuration of said call completion actions on said calling party device,**

96. Prikowitsch teaches Elements 5[a]/15[a]. Prikowitsch teaches that the action execution criteria comprises configuration of the call completion actions on the calling party device. Specifically, the call completion application software installed on the calling party's device, allows the caller to implement and prescribe certain responsive actions (i.e., action execution criteria) for the application to execute based on reported call status and events (e.g., the called party device is busy) to complete the call. Ex-1011, 5:19-25. The prescribed actions are a factor used in determining the appropriate method to complete the call because they dictate the specific action to be executed.

- (2) Elements 5[b]/15[b]: one of a presence and an absence of one or more of a message box and a messaging application for receiving media data on one of a network account of said called party and a called party device,**

97. Prikowitsch teaches Elements 5[b]/15[b]. Prikowitsch teaches that the action execution criteria comprises the presence of a message box for receiving media data on the called party's device. Prikowitsch's "memory unit" "associated with the called party device" includes a "recording unit, a telephone answering

machine or an electronic memory” and “may also be a buffer integrated in a communication network” so that messages may be transmitted at a later time if the called party is not “logged in or available.” Ex-1011, 4:1-8. Prikowitsch’s telephone answering machine associated with the called party device is a message box for receiving media data. A POSITA would have understood that the presence of a telephone answering machine for receiving media data on the called party device is a factor used by the application to determine the appropriate method for executing a call completion action such as whether media data like a text or video message can be sent to the called party to complete the call. For example, Oren considers the presence of a message box when completing the call action when determining to expedite or delay the multimedia message. Ex-1049 ¶28.

(3) Elements 5[c]/15[c]: one of a presence and an absence of a data connection from said calling party device to a network,

98. Prikowitsch teaches Elements 5[c]/15[c]. Prikowitsch teaches that the action execution criteria comprises the data connection from the calling party device to a network. For example, “network services required for the application to work are signaled to the” calling party’s device in real time using a server. Ex-1011, 6:6-9. Activation of the call completion application requires communication, or a data connection between the device and the “subscription/application server.” *Id.*, 11:22-25. The network that connects the terminal devices to each other and the server is

annotated in yellow in Figure 4 below. A POSITA would have understood that the presence of a data connection from the calling party device to a network is a factor used by the application to determine the appropriate method for executing a call completion action such as whether a ready-made text, video message, or an announcement can be sent to the called party to complete the call.

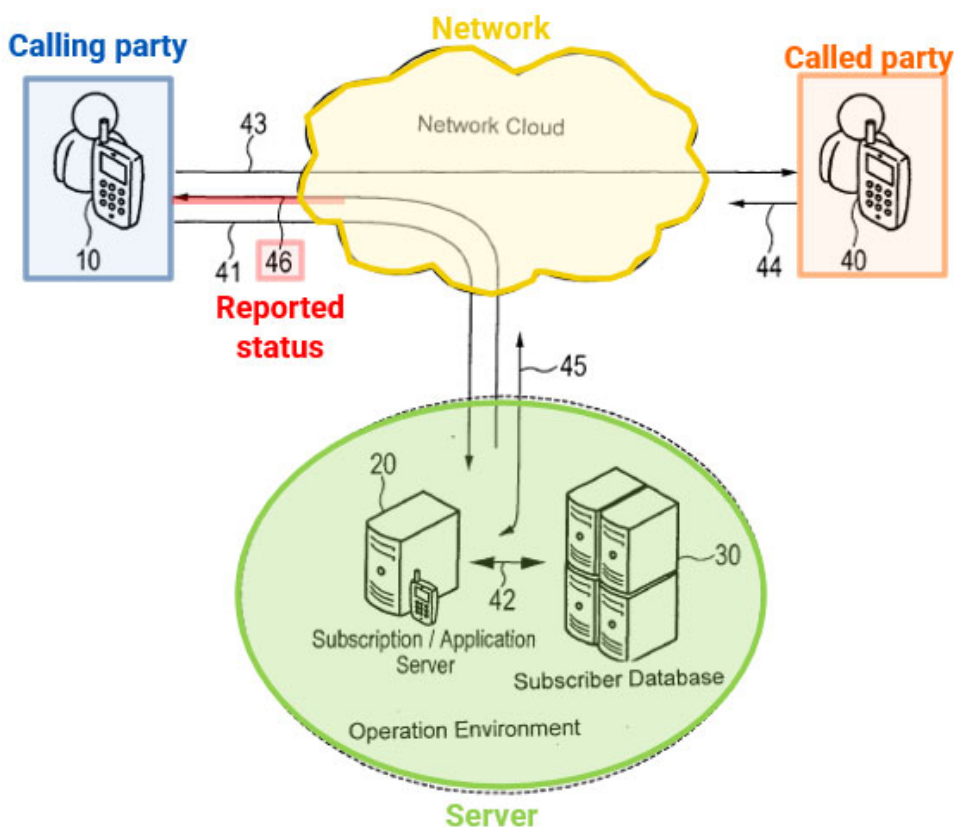


Fig. 4

99. A POSITA would have understood further in view of Anderl which detects a failed call attempt due to “occurrence of a busy signal, an out of coverage signal, an unreachable signal, congestion of one or more of the first network 203 and

the second network 204, an unanswered signal from the second party 205, and manual call termination by the first party 201.” Ex-1050 ¶19. Once the call attempt is terminated, the first network 203 communicates 102 availability of a messaging service (such as an SMS, voice SMS, multimedia messaging service, and a multimodal messaging service) to the first party 201. Ex-1050 ¶20. Thus, it was well known to consider whether there is a data connection when determining the appropriate method for executing a call completion action.

(4) Elements 5[d]/15[d]: one of a presence and an absence of a messaging service in a network component of said called party,

Elements 5[e]/15[e]: one of a presence and an absence of a messaging service in a network component of said calling party,

100. Prikowitsch teaches Elements 5[d]/15[d] and 5[e]/15[e]. Prikowitsch teaches that the action execution criteria comprises a presence of a messaging service in a network component of the called party. Specifically, Prikowitsch’s call completion application of the calling party allows the “sending of ready-made SMS/MMS/voice messages” to the called party. Ex-1011, 7:13. A POSITA would understand that in order to perform the stated “sending of” messages, a messaging service within the network that connects to the called party would be used. Moreover, Prikowitsch’s memory unit includes an answering machine and “buffer integrated in a communication network” that allows transmission of messages at a

later time if the called party is not “logged in or available.” Ex-1011, 4:1-8. A POSITA would have understood this capability to be possible only with the presence of a messaging service in the network component of both the called party and calling party, which would be a factor used by the application to determine the appropriate method for executing a call completion action such as whether a text or video message can be sent to the called party to complete the call. In particular, whichever network component transmits messages is a messaging service. This would have been known by a POSITA for the same reasons I explained in the previous section and Section VII (paragraphs 32-51).

(5) Elements 5[f]/15[f]: reach of said messaging service to deliver said media data to one of said network component of said called party and said called party device,

101. Prikowitsch teaches Elements 5[f]/15[f]. Prikowitsch teaches that the action execution criteria includes the reach of the messaging service to deliver the media data to the called party’s network component or device. A POSITA would have understood “reach of the messaging service” to mean the messaging service’s ability or availability to transmit a media data. As I explained in Section VII.A (paragraphs 32-34), Parlay X would determine if the device was “reachable, unreachable, and busy.” Ex-1047, 18. If a device was reachable, it would be available and able to receive a call. If it was unreachable, it was not available to receive a call. Thus, the “reach” of the messaging service would have been

understood as the availability of the “messaging service.” Prikowitsch’s action execution criteria includes reach because it determines whether it is possible to deliver a message to the called party based on the “quality or availability of a radio link.” Ex-1011, 3:20. A POSITA would have understood that the reach of the messaging service (e.g., the quality and availability of a radio link) is a factor used by the application to determine whether a ready-made text or video message can be sent to the called party to complete the call. *See, e.g.*, Ex-1049 ¶28 (a multimedia message may be set for deferred delivery on a later schedule, which may be selected by the network operator (e.g., low cost, low priority, quality of service parameters, service plan, etc.), by the called party (e.g., “I’ll be back from my honeymoon on . . .”, etc.) or by the calling party (e.g., a “birthday card”, etc.)).

(6) Elements 5[g]/15[g]: network options provided to said calling party device and said called party device by said network to which said calling party device and said called party device are connected, and

102. Prikowitsch teaches Elements 5[g]/15[g]. Prikowitsch teaches that the action execution criteria includes network options provided to the calling party device and the called party device by the network to which the devices are connected. Specifically, Prikowitsch teaches that network status, availability, and overload are factors used to determine the appropriate method for executing a call completion action. Ex-1011, 3:17-23. For example, Prikowitsch’s memory unit may be “a buffer

integrated in a communication network” that allows transmission of the calling party’s messages at a later time if the called party is not logged into the network or available. *Id.*, 4:4-8. Because they are factors used by the application to determine the appropriate method of completing the incomplete call by sending the message over the network immediately or delayed, the network options are action execution criteria.

(7) Elements 5[h]/15[h]: call completion preferences configured by said call completion application on said calling party device.

103. Prikowitsch teaches Elements 5[h]/15[h]. Prikowitsch teaches that the action execution criteria includes call completion preferences configured by the calling party device. Specifically, Prikowitsch’s calling party defines call completion preferences on their device for the application to execute the described action if a call is incomplete (Ex-1011, 7:4-15), which are saved in an appropriate user “profile on the server of the network itself.” *Id.*, 6:24-30.

d. Claims 8/17: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said media data comprises one of text data, audio data, video data, audiovisual data, image data, multimedia data, message data, and any combination thereof.

104. Prikowitsch discloses Claims 8 and 17. Prikowitsch discloses that the media data comprises at least text, audio, multimedia, and message data. For example, Prikowitsch transmits media data comprising text, audio, multimedia, and

message data such as “ready-made text messages and/or multimedia messages and/or voice messages.” Ex-1011, 5:28-33.

- e. **Claims 9/18: [The computer implemented method of claim 1, further comprising determining / The computer system of claim 11, wherein said action execution module is further configured to determine] a mode of transmission of media data from said calling party device to a called party device via a network, by said call completion application on said calling party device based on one or more of said action execution criteria.**

105. Prikowitsch teaches Claims 9 and 18. Prikowitsch determines a mode of transmission of media data from the calling party device to a called party device via a network, by the call completion application on the calling party device based on one or more of the action execution criteria defined by the user. *See* Section XI.A.1.g, paragraphs 90-92 (Element 1[e]/11[f]). Specifically, Prikowitsch’s call completion application executes the caller’s defined responsive actions to complete an incomplete call, by sending “individual announcements,” implementing “diversions and/or forwarding operations,” sending “ready-made SMS/MMS/voice messages,” generating “appropriate announcements,” and/or making “calendar entries relating to return calls.” Ex-1011, 7:9-15. The actions are executed by the application based on action execution criteria (i.e., user-defined responsive actions) depending on the reported status information. *Id.*, Cl. 5. For example, when the reported status indicates that the called party is busy, the calling party’s defined

actions or profile (i.e., configurable criteria) linked to the “busy” event (e.g., call forwarding operations) are used to complete the call. *Id.*, 13:5-15; *see also id.*, 4:5-8.

106. Prikowitsch teaches determining the “mode of transmission” of the media data based on the action execution criteria in the same manner as the ’227 Patent. The ’227 Patent states that the “mode of transmission” is the “form” of media data that would be transmitted to the called party, such as “a voice message, a text message, a missed call alert,” or “a notification a call attempt”; and the way it is transmitted, for example, directly or through a voice short message server. Ex-1001, 9:3-16. Prikowitsch determines the appropriate form of media data based on the caller’s defined actions, for example, sending a ready-made text or voice message, announcement, or return call calendar entry if the called party is busy or unavailable. Moreover, Prikowitsch determines the appropriate way the media data (e.g., ready-made text or voice message) is transmitted, for example, directly to the called party or through a buffer in the communication network to be transmitted at a later time. Ex-1011, 4:5-8.

- f. **Claims 10/19: The computer implemented method of claim 1, further comprising monitoring/[The computer system of claim 11, wherein said call detection module is further configured to monitor] said incomplete call by said call completion application on said calling party device as said incomplete call progresses from said incomplete call being initiated, said incomplete call being one of answered and rejected, and said incomplete call being terminated.**

107. Prikowitsch teaches Claims 10 and 19. Prikowitsch call completion application monitors the calling party device as the incomplete call progresses from being initiated, answered or rejected, and terminated. Specifically, Prikowitsch begins monitoring the call when it is outgoing (i.e., *initiated*). Ex-1011, 3:17. If “the network sends a ‘Busy’ acknowledgement,” this *rejection* “event” is detected from the network and signaled to the application on the calling party’s device “for further evaluation.” Ex-1011, 13:5-9. The user-defined “actions linked to the ‘busy’ event” are then performed (*Id.*, 13:11-12), when the application determines that the call is terminated because these actions would end the call process. Therefore, Prikowitsch monitors the call progression from when its initiated, rejected because the caller is busy, and *terminated* when the reactive actions linked to busy are executed.

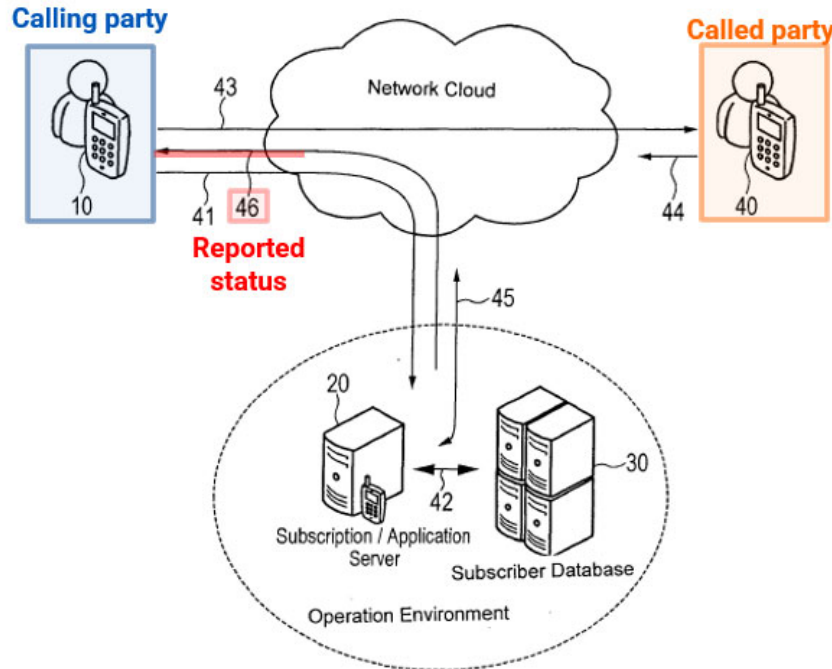


Fig. 4

108. As I explained above in Section VII.A, it was well known to monitor the call's progression. Specifically, in paragraph 39, I explained how Bychkov monitors the outgoing call from the caller and then determine whether the callee party is available. Ex-1051, 3:41-47. Bychkov describes that a callee “may not be available for a variety of reasons,” such as the callee’s phone being turned off or busy. Ex-1051, 3:44-49. Moreover, Bychkov identifies that the callee simply may not wish to answer the call or the callee’s phone may forward to a voice mail box, but the caller did not want to leave a voice message. Ex-1051, 3:49-52. Additionally, Bychkov determines that the callee is available based on call duration; e.g., a

duration shorter than 10 seconds may be used as an indicator that the callee is not available. Ex-1051, 3:52-55. Thus, it is well known to monitor the call's progression from when it was initiated to being unanswered and terminated.

B. Ground 2: Claims 3, 4, 9, 13, 14, and 18 are Rendered Obvious by Prikowitsch in View of Roujinsky

1. A POSITA would have been motivated to combine Prikowitsch and Roujinsky and had a reasonable expectation of success

109. In addition to the reasons set forth below, a POSITA would have been motivated to combine the teachings of Prikowitsch and Roujinsky (“Prikowitsch-Roujinsky”) and would have had a reasonable expectation of success in doing so because each relates to systems and methods for implementing pre-defined actions in response to an incomplete call. Ex-1012, 5:31-6:2. For example, Prikowitsch allows the user to define individualized actions for responding to an incomplete call, which are in their profile saved in an account database accessible by the server. Ex-1011, 1:22-2:2, 4:21-25, 5:21-24, 12:24-28. Prikowitsch defines an incomplete call as a call that was not answered due to the call settings, network condition, or the recipient's availability. Ex-1011, 3:5-8, 5:21-33, 7:4-15. The caller's defined actions include sending “individual announcements, tones or music, particularly for calls via a mobile radio network,” implementing “diversions and/or forwarding operations,” “sending of ready-made SMS/MMS/voice messages,” generating “appropriate

announcements, and “making of calendar entries relating to return calls.” Ex-1011, 7:4-17.

110. Similarly, Roujinsky provides call completion software that compiles a list of unanswered calls that trigger “a user-defined action so as to enable communication” between a caller and called party. Ex-1012, 10:28-11:1. As shown in Figure 3, Roujinsky specifically analyzes the record parameters and registers an “uncompleted call” when the recipient is out of coverage, the recipient’s device is turned off or busy, or the call had a short duration. Ex-1012, 1:6-16, 16:9-14. When an “unanswered call” is detected, Roujinsky determines the appropriate transmission mode for the user-defined action and automatically issues a predefined message such as SMS, MMS, or voice message to the recipient about the call attempt and an invitation to call the caller back. Ex-1012, 12:14-30.

111. A POSITA would have been motivated to combine Roujinsky’s teachings of detecting an incomplete call based on call duration and determining the appropriate mode of transmission with Prikowitsch’s system because they would have recognized a benefit of detecting an incomplete call across a variety of scenarios “regardless of whether they were missed or could not be received owing to communication failures, poor coverage and so on.” Ex-1012, 15:13-15. Thus, a POSITA seeking to maximize call completion in a communication system by automatically registering unanswered telephone calls made over a communication

channel and triggering user-defined actions, would have known and consulted both of these proposed solutions.

112. In the combined system, Prikowitsch's call application software would detect incomplete calls based on call duration and determine the appropriate mode of transmission for the user-defined action, as disclosed in Roujinsky, in order to ensure that the recipient receives notification of the caller's attempted call.

113. A POSITA would have had a reasonable expectation of success in combining these teachings of Prikowitsch and Roujinsky because each purports to solve these problems in similar and predictable ways, such as by monitoring incomplete calls and automatically triggering events that are individualized and defined by the caller in order to improve call completion of all types of unanswered calls without expensive equipment and central processing servers. Ex-1012, 15:5-15. A POSITA would have understood that they could have combined these teachings with little or no change in their respective functions because they teach compatible systems. A POSITA would have understood that these references disclose interrelated teachings using caller and recipient terminal devices that can execute software that is easily downloaded regardless of SIM version and size, with routine network and data connections and technologies that would have been amendable to various well-understood and predictable combinations, as explained in further detail below.

2. Dependent Claims 3, 4, 9, 13, 14, and 18

- a. Claims 3/13: [The computer implemented method of claim 1/The computer system of claim 11], wherein said incomplete call is a call of a short duration that does not enable an intended communication to occur between said calling party and said called party.**

114. Prikowitsch-Roujinsky teaches Claims 3 and 13. Prikowitsch determines that a call is incomplete based on reported status information such as “the call status and/or the login status of the called terminal device, and/or the availability of the called terminal device and/or the network status.” Ex-1011, 3:11-15, Cl. 2. The reported status specifies that an incomplete call was a call that did not connect to a called party device due to the called party’s login status and availability. A POSITA would have understood that when a call is disrupted and cut short due to network status, the call application would thus consider the duration of the call to determine that the call was incomplete. *See, e.g.*, Ex-1051, 3:52-55 (determining that the callee is unavailable based on call duration; e.g., a duration shorter than 10 seconds may be used as an indicator that the callee is not available). This would have further been obvious because it was well known, as disclosed in Roujinsky.

115. Roujinsky discloses that the call completion software analyzes the call’s record parameters and registers an “uncompleted call” based on whether there was “insufficient time elapsed after initiation of the call prior to hang-up to have allowed for the call to be answered.” Ex-1012, 16:9-23. This is just like the ’227

Patent's short duration call, which is described as an unanswered call that "does not enable an intended communication to occur between the calling party and called party." Ex-1001, 2:60-65, 11:44-46; *see also id.*, 4:66-5:2 (a call "less than about 2 seconds"), 7:11-14 (same), 16:23-26 (explaining that a duration of 2 seconds or more indicates that the call was answered).

116. A POSITA would have been motivated to use Roujinsky's teachings to detect an incomplete call based on its short duration with Prikowitsch's system to implement user-defined actions for a scenario when a call is incomplete because insufficient time has elapsed for it to have been answered. Moreover, a POSITA would have been motivated to consider "call completion of all types of unanswered calls regardless of whether they were missed or could not be received owing to communication failures, poor coverage and so on." Ex-1012, 15:13-15.

b. Claim 4: The computer implemented method of claim 1, wherein said one or more of said call completion actions are received by said call completion application based on a duration of said detected incomplete call

Claim 14: The computer system of claim 11, wherein said action determination module is further configured to receive said one or more of said call completion actions based on a duration of said detected incomplete call.

117. Prikowitsch-Roujinsky teaches Claims 4 and 14. Prikowitsch-Roujinsky teaches receiving call completions actions based on the duration of the detected incomplete call. As discussed in Ground 1 (*see* Section XI.A.1.d,

paragraphs 83-84), Prikowitsch executes the caller's defined responsive actions when an incomplete call is detected. To the extent that Prikowitsch does not explicitly or inherently disclose receiving a defined responsive action based on the duration of an incomplete call, it would have been obvious in view of Roujinsky.

118. Roujinsky's software registers an incomplete call by "determining that insufficient time elapsed after initiation of the call." Ex-1012, 16:9-23. Upon "determining that a call was uncompleted, the call completion software performs some user-defined action." *Id.* Thus, Roujinsky receives the call completion action based on the time elapsed after initiating the call by the caller. This is similar to how the '227 Patent's call completion application completes the short duration call—the application displays a message such as "Looks like the called party could not talk to you at this time. Do you want to send a message to the called party?" Ex-1001, 12:7-10. A POSITA would have been motivated to implement Roujinsky's call completion actions based on a duration of the incomplete call into Prikowitsch for the same reasons explained in Section XI.B.1 (paragraphs 109-113)—to ensure that ensure completion of "all types of unanswered calls." Ex-1012, 15:13-15.

- c. **Claims 9/18: [The computer implemented method of claim 1, further comprising determining / The computer system of claim 11, wherein said action execution module is further configured to determine] a mode of transmission of media data from said calling party device to a called party device via a network, by said call completion application on said calling party device based on one or more of said action execution criteria.**

119. Prikowitsch-Roujinsky teaches Claims 9 and 18. Prikowitsch's call completion application determines the mode of transmission of media data from the calling party device to the called party device. *See* Section XI.A.2.e. This would have been further obvious in view of Roujinsky's software that compiles an "SMS notification" and sends it to the server to convey to the called party's device. Roujinsky applies the same principle when compiling other "types of message[s] sent over different networks such as text, email[,] and so on sent over the IP network and to WAP push messages." Ex-1012, 17:24-31. Roujinsky conveys the caller's message to the server which then "composes a message in a different format for conveying to the called party's [device] via a different network." *Id.*, 18:23-32. Roujinsky also forwards the message "to the called party by any suitable additional network" in accordance "with a list of preference pre-established by the called party." Ex-1012, 17:14-16. Therefore, Roujinsky's software determines the appropriate type and form of the media data (i.e., mode of transmission) to convey the caller's media data to the called party, in the same way as the '227 Patent. Ex-

1001, 9:5-16 (the “mode of transmission” is the “type” and “form” of media data that would be transmitted to the called party, such as determining whether to transmit the message directly or to transmit the message through a voice short message server). Roujinsky determines the appropriate type and form of the media data that the caller had configured to send to the called party if the call was incomplete.

120. A POSITA would have been motivated to implement Roujinsky’s teachings regarding mode of transmission with Prikowitsch’s call completion application in order to ensure that the called party, for example, receives a message that the caller “tried unsuccessfully to make contact.” Ex-1012, 11:18-20. A POSITA would have had a reasonable expectation of success in implementing Roujinsky’s mode of transmission into Prikowitsch because it is merely computer software code using conventional computer hardware.

C. Ground 3: Claims 1-2, 6, 8-12, and 17-19 are Rendered Obvious by Sar-Shalom

1. Independent Claims 1 and 11

- a. Elements 1[pre]/11[pre]: A computer implemented method/[A computer system] for completing an incomplete call made by a calling party to a called party, comprising**

121. Sar-Shalom teaches the preambles of Claims 1 and 11. Sar-Shalom teaches an “automatic messaging client launcher,” which initiates the communication’s device messaging client to “perform[] or complete[] selected tasks or processes” when the called party is unavailable. Ex-1013, Abstract, ¶25.

Specifically, Sar-Shalom's messaging client allows the calling party to complete an incomplete call, by for example, easily sending messages to an unavailable call recipient. *Id.* ¶33. Sar-Shalom's automatic messaging client launcher is a method and/or system "implemented by hardware or by software on any operating system of any hardware or firmware," such as a computer. *Id.* ¶25;

- b. Element 1[a]: providing a call completion application executable by at least one processor on a calling party device;**

Element 11[a]: a non-transitory computer readable storage medium configured to store a call completion application on a calling party device;

Element 11[b]: at least one processor communicatively coupled to said non-transitory computer readable storage medium, said at least one processor configured to execute said call completion application on said calling party device; and

122. Sar-Shalom teaches Elements 1[a] and 11[a]-11[b]. Sar-Shalom teaches a call competition application executable by a processor on a calling party device. Specifically, Sar-Shalom's "automatic caller launcher" is a "method and/or system" that launches a messaging client in response to a call being answered or not answered. Ex-1013 ¶¶23, 53. The messaging client is launched "in accordance with a set of user-definable rules" and allows the user to define and select the responsive action to be executed in response to an incomplete call. *Id.* ¶¶45, 46. For example, the messaging client is launched and may display "an input screen on the

communication device” for the calling party to input the message content and/or other data to be sent. *Id.* ¶¶41, 58.

123. Sar-Shalom’s “automatic caller launcher” uses a processor to process these actions implemented by the calling party that “complete” an unanswered call, independent of a preconfigured call completion service, such as voicemail. For example, Sar-Shalom’s caller launcher initiates the messaging client to send a SMS, EMS, or MMS message to the called party based on a trigger signaling that the call was incomplete but without the need of a voicemail. Ex-1013 ¶40. Sar-Shalom teaches that SMS messaging is an attractive communication option for many callers, and that MMS was expected to increase that trend. *Id.* ¶61. Sar-Shalom thus completes the call independent of a voicemail or preconfigured call completion service because it messages an unavailable destination device for the caller using commonly-adopted communication options, e.g., SMS or MMS, operating independently from a voicemail service.

124. Sar-Shalom’s messaging client with the integrated messaging launcher (i.e., software) is a call completion application on the calling party device Sar-Shalom’s “automatic caller launcher” (and thus the messaging client and integrated messaging launcher) is “implemented by hardware or by software on any operating system of any hardware or firmware or a combination thereof.” Ex-1013 ¶25. The method and system is “performed by a data processor, such as a computing platform

for executing a plurality of instructions” (*id.* ¶25) on the calling party’s “communication device.” *Id.* ¶¶19, 36. The calling party device contains non-volatile memory for storing addresses and messages. Ex-1013 ¶¶39, 41. A POSITA would thus have understood that the automatic caller launcher is executed using a processor on the caller’s party device that communicates with memory, which also stores the call completion application, for the same reasons I explained in paragraph 79 above. The von Neumann architecture is a well-known basic computer architecture to run a computer stored-program. *See, e.g.*, Ex-1053.

125. Moreover, as I explained above in Section VI.A (paragraph 38), this was well known as evidenced by Anderl. Anderl discloses a computer program product comprising computer executable instructions (Ex-1050 ¶35) of completing a call made by a first party on a first network to second party on a second network. Ex-1050 ¶19. Anderl notes that it would be “readily apparent that the various methods and algorithms” would be implemented in a computer readable medium computer readable medium appropriately programmed for general purpose computers and computing devices. *Id.* ¶35. Specifically, “a processor, for e.g., one or more microprocessors will receive instructions from a memory or like device, and execute those instructions, thereby performing one or more processes defined by those instructions.” *Id.*

- c. **Element 1[b]: detecting said incomplete call made by said calling party to said called party, by said call completion application on said calling party device,**

Element 1[c]: wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events;

Element 11[c]: said call completion application on said calling party device, comprising:

Element 11[d]: a call detection module configured to detect said incomplete call made by said calling party to said called party

126. Sar-Shalom teaches Elements 1[b], 1[c], 11[c], and 11[d] by detecting an incomplete call made by the calling party to the called party. Sar-Shalom's automatic messaging client launcher includes "availability detector 110 [that] monitors the call" (Ex-1013 ¶36), from the "time the ring tone is present until that call is answered or until a busy signal or any other indication of failure to complete the call is obtained," in order to determine whether the device being called is available. *Id.* ¶38; *see also id.* ¶19. As shown in Figure 3, step S310 (yellow), a determination is made whether the call recipient answers. *Id.* ¶53. The call may be unanswered because the called party device is unavailable and "does not connect" so that the "voice call between the two parties cannot be completed." Ex-1013 ¶¶36, 38. The called party device is also considered unavailable if there is a busy signal, if the call is not answered "after a predetermined number of rings," the call connected to a voicemailbox, or if the called party device is offline. *Id.* ¶38. When the device

is determined to be unavailable and the call is not answered, the call is detected as incomplete.

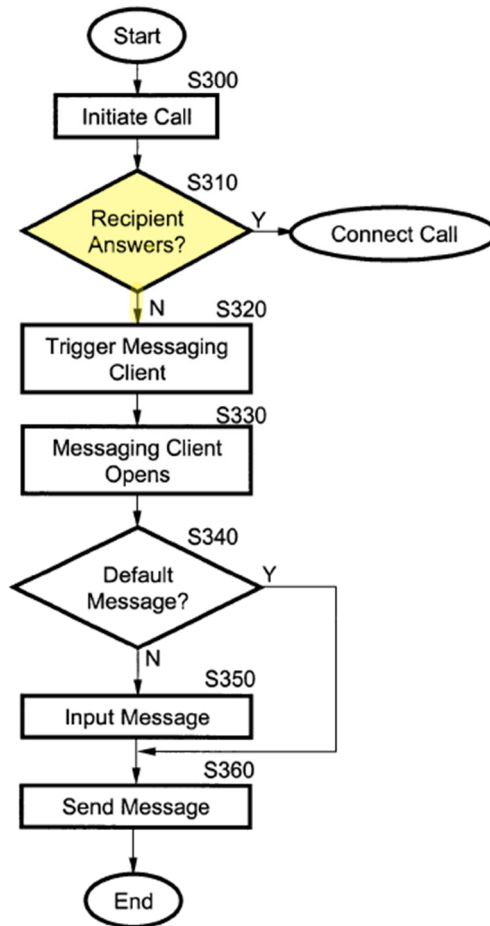


FIG. 3

- d. Element 1[d(i)]: receiving one or more of a plurality of call completion actions to be performed by said call completion application on said calling party device to complete said detected incomplete call,**

Element 11[e(i)]: an action determination module configured to receive one or more of a plurality of call completion actions to be performed to complete said detected incomplete call,

127. Sar-Shalom teaches Elements 1[d(i)] and 11[e(i)]. Sar-Shalom teaches receiving a plurality of call completion actions to be performed to complete the detected incomplete call. Sar-Shalom's automatic messaging client launcher provides a "trigger signal" derived from "answer/no-answer" data to open the messaging client (S330) to send a message (S350). Ex-1013 ¶¶23, 53. The messaging client 130 displays a choice of responses on the user device's input screen so that the user may input the "SMS, EMS, or MMS message" comprising "text, audio clips, video clips, still images, multimedia presentations and voice." *Id.* ¶¶40, 41, 45. The message may be standard or definable by the user such as to include the "caller name, and the date and time the call was made." *Id.* ¶42. These actions are received in order to complete the detected incomplete call. That is, the actions are triggered based on the "no-answer" signal status. Ex-1013 ¶¶23, 53 (based on "no-answer" signal status). Therefore, the call application receives "one or more of a plurality of call completion actions to be performed" to complete the incomplete call

when receiving the calling party's individualized and defined actions in response to a determined incomplete call.

128. These actions are performed by said call completion application on the calling party's device. As explained above, the communication device's messaging client 130 displays "an input screen" on the user's device, "for user input of message content." Ex-1013 ¶41. A POSITA would have understood that Sar-Shalom's messaging client thus includes an "an action determination module" because it is software that allows the user to configure the call completion actions to be performed to complete the detected incomplete call. A POSITA would have understood that that the module is software to carry out the function because, like Oren, there are code segments to execute each function of the call completion method. *See, e.g.*, Ex-1049 ¶6, Claims 15, 22-24.

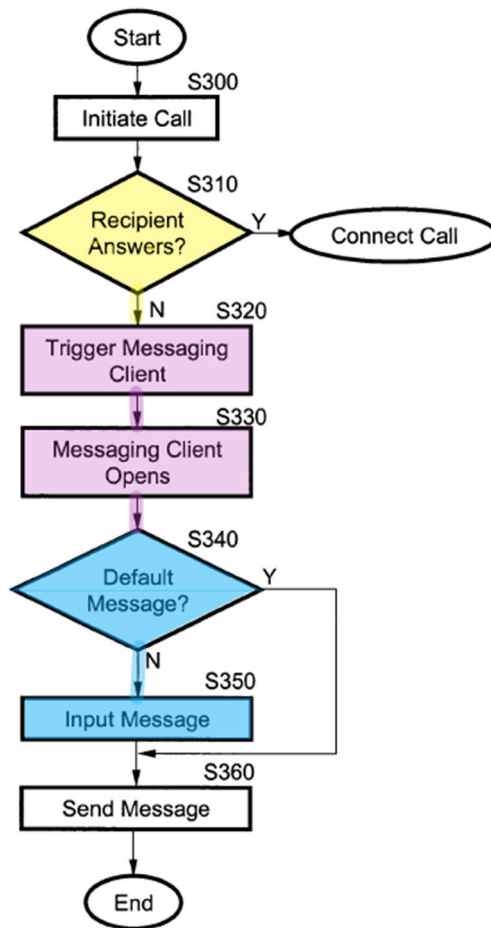
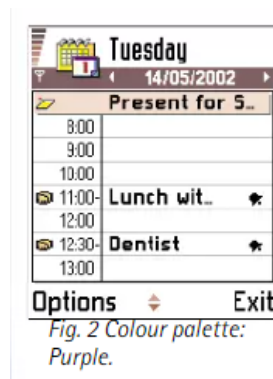


FIG. 3

- e. **Elements 1[d(ii)]/11[e(ii)]: wherein said plurality of call completion actions are selected according to one or more responses provided by said calling party through a graphical user interface of said calling party device,**

129. Sar-Shalom teaches Elements 1[d(ii)] and 11[e(ii)]. Sar-Shalom further explains that the calling party provides the call completion actions through responses via their device's graphical interface. The calling party "is given a choice of responses" on the screen of their device to select such as "leaving an SMS or MMS message, leaving a voice message, or disconnecting." Ex-1013 ¶45. For example,

the messaging client launches and displays “an input screen on the communication device” for the calling party to input the message content and/or other data to be sent to the called party upon the calling party’s user command. *Id.* ¶¶41, 58 (“input screen ... showing a blank or default message”). A POSITA would have understood that this “input screen” is part of a “graphical user interface” as claimed. For example, Sar-Shalom explains that its invention is implemented on a NOKIA 7650 cellular telephone, using the SYMBIAN operating system (Ex-1013 ¶56), which was known to have an input screen as part of a graphical user interface (as shown in Figure 2 below). *See* Ex-1054, 26.



Moreover, Sar-Shalom’s messaging initiator 120 launches messaging client 130 “in accordance with a set of user-definable rules” which is “within the communication device or a network address book.” *Id.* ¶46. A POSITA would have understood that the actions to execute to complete the call are selected according to one or more responses provided by the calling party by the “user-defin[ed]” rules set on the calling party’s device.

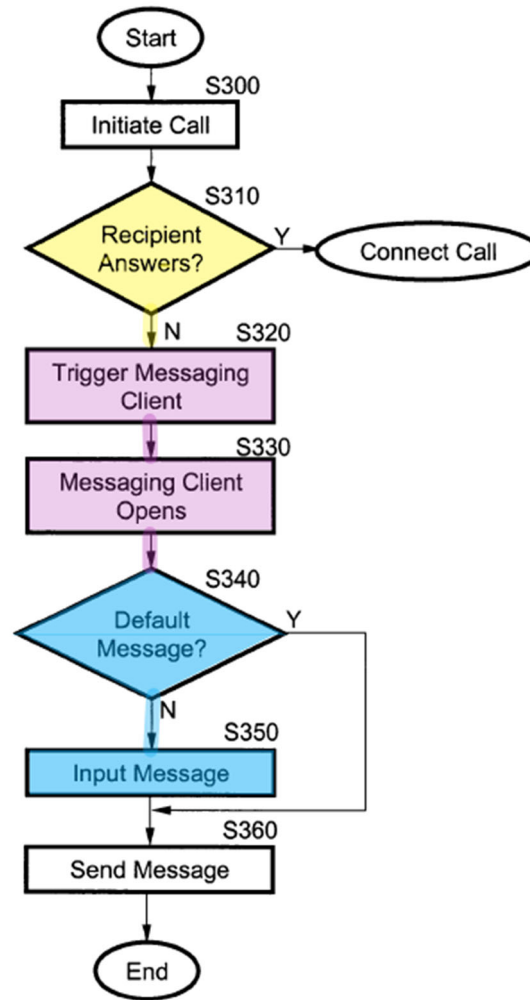


FIG. 3

- f. **Elements 1[d(iii)]/11[e(iii)]: wherein said call completion actions comprise [1] setting a reminder to call back said called party at a configurable time, [2] recording media data on said calling party device, [3] transmitting said media data to a called party device, [4] transmitting a missed call alert to said called party device, [5] transmitting a notification of said detected incomplete call to said called party device, [6] transmitting a notification on availability of said called party, [7] transmitting said media data to a social networking platform, [8] transmitting an automated message requesting said called party to call back said calling party when available, and [9] any combination thereof; and**

130. Sar-Shalom teaches Elements 1[d(iii)] and 11[e(iii)], specifically at least call completion actions labeled [2]-[3].⁸ Sar-Shalom ([3]) sends a “SMS, EMS, or MMS message” as a reply to the called party (Ex-1013 ¶50), comprising “text, audio clips, video clips, still images, multimedia presentations and voice.” *Id.* ¶¶40, 45. The message may be a standard or customized message that includes “the caller name, and the date and time the call was made.” *Id.* ¶42. Sar-Shalom’s audio, video, or multimedia message is recorded media ([2]) because the user can input a default or customized message using their voice. *Id.* ¶43 (“recording a voice message”). The media data is then sent to the calling party as a reply so that it is transmitted to the called party device. Thus, Sar-Shalom call completion actions include recording and transmitting media data to a called party device.

⁸ If the call completion application receives at least one of the call completion actions in Elements 1[d(iii)]/11[e(iii)], these elements are met. *See supra* Section XI.A.1.f.

131. As I described above in Section VII.B (paragraphs 42-51), it was common and well known to implement call completion actions beyond the use of voicemail. Specifically, I explained how POSITAs acknowledged the benefits of allowing the calling party to control what happens to the attempted call so as to allow the calling party to take non-standard actions or alternatives based upon [their] circumstances.” Ex-1048, 2:56-64; *see also id.*, 5:4-10; Ex-1049 ¶¶5 (providing “flexible architecture” that allows the calling party’s terminal to complete a call that has not been answered “by completing an interaction between the parties, rather than just completing a voice call”); ¶¶27-28 (prompting the calling party to compose a multimedia message including text, pictures, and audio files); Ex-1050 ¶¶6, 12 (once it is determined that a call attempt has been terminated, the caller is notified of the called party’s availability and is the caller “creates a message to be delivered to the second party on the second network” such as by “recording one or more of a text SMS message, a voice SMS message, a multimedia messaging service message, and a multimodal message on a communication device of the first party.”); Ex-1051, 3:58-63, 4:9-14 (activates a “message editor on the caller's mobile phone” which “displays a pre-defined message” that the caller can edit).

- g. Element 1[e]: triggering execution of said received one or more of said call completion actions by said call completion application on said calling party device based on one or more of action execution criteria for said completion of detected incomplete call.**

Element 11[f]: an action execution module configured to trigger execution of said received one or more of said call completion actions based on one or more of action execution criteria one or more responses selected by said calling party for said completion of said detected incomplete call.

132. Sar-Shalom teaches Elements 1[e] and 11[f]. Sar-Shalom’s application triggers execution of the call completion actions by the application on the calling party device based on action execution criteria for completion of the detected incomplete call. Sar-Shalom’s messaging initiator 120 detects that the called party communication device is unavailable and launches messaging client 130 “to apply the [user-definable] rules.” Ex-1013 ¶46. Once the user-defined rules are applied and the message is tailored by the user, the message is sent (Figure 3, S360 (green)). *Id.* ¶53; *see also id.* ¶47 (“the reply is sent to the unavailable destination device”). A POSITA would have understood that Sar-Shalom’s messaging initiator is an “action execution module,” i.e., software that triggers execution of the call completion actions on the caller’s device because it is a computer program product made of software instructions executable by the processor on the calling party’s communication device. Therefore, the software executes the caller’s responsive actions to complete the call by sending the message.

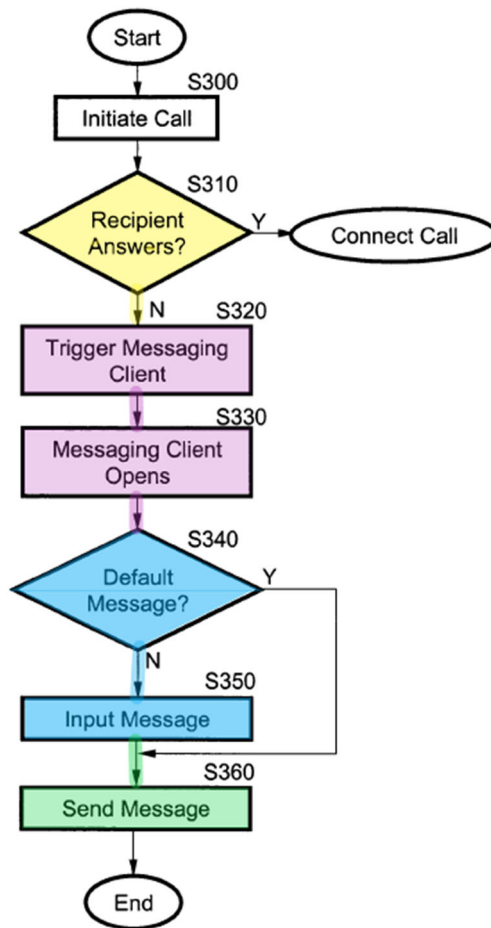


FIG. 3

133. The actions are executed based on action execution criteria. As explained above, once messaging initiator 120 detects that the called party's communication device is unavailable, messaging client 130 applies "the [user-definable] rules" to allow tailoring of the caller's message. Ex-1013 ¶¶46, 53. In such case, "unavailable" is an example of action execution criteria because it sets out what the application will do to execute the selected call completion action, such as implementing a tailored message to the unavailable device. A rule, such as implementing a tailored message when the called device is unavailable, is an

example of action execution criteria because it sets forth what the application will do to execute the selected call completion action. *See* Ex-1001, 8:49-53 (“The action execution criteria are a set of factors that determine an appropriate method for executing a call completion action.”).

2. Dependent Claims 2, 6, 8-10, 12, and 17-19

- a. Element 12[a]: The computer system of claim 11, wherein said incomplete call is a call that is not connected to a called party device due to occurrence of one or more of a plurality of events,**

134. Sar-Shalom teaches Element 12[a]. Sar-Shalom’s incomplete call is a call that is not connected to a called party device due to the occurrence of one or more of a plurality of events for the same reasons explained in Section XI.C.1.c (Element 1[c]). Sar-Shalom detects that the called party device is unavailable and thus that the call is incomplete when the “device does not connect” so that the “voice call between the two parties cannot be completed.”

- b. Claim 2/Element 12[b]: [The computer implemented method of claim 1,] wherein said events comprise said called party being busy, said called party device being in an out of coverage area, said called party device being unreachable, said called party device being switched off, network congestion, and said call not being answered by said called party.**

135. Sar-Shalom teaches Claim 2 and Element 12[b].⁹ Sar-Shalom determines that the call is not connected to a called party device due to the called party being busy, the called party device being out of coverage area, unreachable, or switched off, or the call not being answered. Specifically, the call is determined to be incomplete when there is a “busy signal detected” or if “an indication is received... that the destination device is offline.” Ex-1013 ¶38. A POSITA would have understood that when the called party device is determined to be busy or offline, then the call would be incomplete because the called party device is out of coverage, unreachable, switched off. Ex-1013 ¶¶36, 38.

136. Sar-Shalom further teaches that the call is incomplete due to the call not being answered after a predetermined number of rings or connecting to a voicemail box. Ex-1013 ¶38.

⁹ If at least one of the events in Claim 2/Element 12[b] occurs, these limitations are met. *See supra* Section XI.C.1.c (Element 1[c] reciting “one or more plurality of events”) and Section XI.C.2.a (Element 12[a] reciting “one or more plurality of events”).

- c. **Claim 6: The computer implemented method of claim 1, wherein said one or more of said call completion actions to be performed by said call completion application on said calling party device to complete said detected incomplete call is based on one or more of configurable criteria configured by said calling party.**

137. Sar-Shalom teaches Claim 6. Sar-Shalom bases the call completion actions to be performed on configurable criteria configured by the calling party. Messaging client 130 uses “user-definable rules” that include “various criteria” configured by the calling party on their communication device. Ex-1013 ¶46; *see also id.* ¶52 (“set and/or change preferences.”); Section XI.C.1.d (Elements 1[d(i)]/11[e(i)]). Specifically, messaging initiator 120 provides additional information “provisioned by the user” required by messaging client 130 to apply the rules so that the user can tailor or customize messages sent for different purposes. Ex-1013 ¶46. Therefore, the calling party configures the “call completion actions” by defining the actions to complete said detected incomplete call.

- d. **Claims 8/17: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said media data comprises one of text data, audio data, video data, audiovisual data, image data, multimedia data, message data, and any combination thereof.**

138. Sar-Shalom teaches the media data in Claims 8 and 17. Sar-Shalom teaches that the message includes media data like “text, audio clips, video clips, still

images, multimedia presentations and voice” via “SMS, EMS, or MMS message.”

Ex-1013 ¶¶40, 45.

- e. **Claims 9/18: [The computer implemented method of claim 1, further comprising determining / The computer system of claim 11, wherein said action execution module is further configured to determine] a mode of transmission of media data from said calling party device to a called party device via a network, by said call completion application on said calling party device based on one or more of said action execution criteria.**

139. Sar-Shalom teaches Claims 9 and 18. Sar-Shalom determines a mode to transmit the media data to the called party via a network based on one or more of the action execution criteria. *See* Section XI.C.1.g (Elements 1[e]/11[f]). Specifically, Sar-Shalom’s messaging initiator 120 detects that the called party communication device is unavailable and launches messaging client 130 “to apply the [user-definable] rules.” Ex-1013 ¶46. For example, when the called party device “is unavailable the user is given a choice of responses, such as leaving an SMS, or MMS message leaving voice message, or disconnecting without leaving any message.” *Id.* ¶45. Based on the calling party’s rules, the user may input the message content either manually or by selecting a defined message or the message is sent “automatically.” *Id.* ¶¶41-42, 44. Sar-Shalom thus determines the appropriate form to send the message—either automatically or after the user has composed the message—based on the user’s defined rules. This is the same “mode of

transmission” described in the ’227 Patent (Ex-1001, 9:5-16) because Sar-Shalom determines the appropriate “type” and “form” of the media data that the caller had configured to send to the called party once the call was determined incomplete.

- f. Claims 10/19: [The computer implemented method of claim 1, further comprising monitoring / The computer system of claim 11, wherein said call detection module is further configured to monitor] said incomplete call by said call completion application on said calling party device as said incomplete call progresses from said incomplete call being initiated, said incomplete call being one of answered and rejected, and said incomplete call being terminated.**

140. Sar-Shalom teaches Claims 10 and 19. Sar-Shalom monitors the incomplete call as it progresses from being initiated, answered or rejected, and terminated. Sar-Shalom’s availability detector 110 “monitors the calls placed” by the calling party and “detects whether a destination device is unavailable” (Ex-1013 ¶20), by “monitor[ing] the call from the time the ringtone is present until that call is answered or until a busy signal or any other indication of failure to complete the call is obtained.” *Id.* ¶38. For example, if the call is not “answered after a predetermined number of rings,” or it “connects to a voicemail box,” then the called party is determined to be unavailable. *Id.* The user-definable rules are then applied when it is determined that the call is terminated, which ends the call process.

D. Ground 4: Claims 3-5 and 13-15 are Obvious Over Sar-Shalom in View of Roujinsky

1. A POSITA would have been motivated to combine Sar-Shalom and Roujinsky and had a reasonable expectation of success

141. In addition to the reasons set forth below, a POSITA would have been motivated to combine the teachings of Sar-Shalom and Roujinsky (“Sar-Shalom-Roujinsky”) and would have had a reasonable expectation of success in doing so because each relates to systems and methods for implementing pre-defined actions in response to an incomplete call. Ex-1013 ¶¶46; Ex-1012, 5:31-6:2. For example, Sar-Shalom detects that the called party device is unavailable when the “device does not connect” or the “voice call between the two parties cannot be completed.” Ex-1013 ¶¶36, 38. Upon determining that the call is incomplete, Sar-Shalom’s automatic messaging client launcher provides a “trigger signal” to launch the messaging client “in accordance with a set of user-definable rules.” Ex-1013 ¶¶23, 46, 53. The calling party is given a choice of responses to send to the unavailable device, such as sending a “SMS, EMS, or MMS message” comprising of “text, audio clips, video clips, still images, multimedia presentations and voice.” *Id.* ¶¶40, 45. The message may be standard or definable by the user such as to include the “caller name, and the date and time the call was made.” *Id.* ¶52.

142. As explained above in Sections X.B (paragraphs 69-71) and XI.B.1 (paragraphs 109-113), Roujinsky also automatically issues “a predefined message”

to the recipient (Ex-1012, 12:14-30), in response to an incomplete call. *Id.*, 10:28-11:1. Roujinsky further determines the call completion action based on criteria about the incomplete call. For example, Roujinsky determines that the calling party device is “not within coverage or is engaged on the telephony channel, thus preventing the call from being completed.” Ex-1012, 16:18-23. When there is a network issue, Roujinsky sends a notification to the called party’s device as an SMS message “conveyed over the cellular network,” and allows “other types of message[s] sent over different networks such as text, email and so on sent over the IP network.” Ex-1012, 17:17-30.

143. A POSITA would have been motivated to combine Roujinsky’s teachings of determining the appropriate method for executing the user-defined actions based on network options and available messaging service with Sar-Shalom’s system because they would have recognized the benefit of considering network options and available messaging service when determining the appropriate method for executing the user-defined action in order to respond to the incomplete call “regardless of whether they were missed or could not be received owing to communication failures, poor coverage and so on.” Ex-1012, 15:13-15. A POSITA would have also been motivated to combine Roujinsky’s teachings of detecting an incomplete call based on call duration with Sar-Shalom’s system for the same reason. Thus, a POSITA seeking to maximize call completion in a communication

system by automatically registering unanswered telephone calls made over a communication channel and triggering user-defined actions in response to incomplete calls would have known of and consulted both of these proposed solutions.

144. In the combined system, Sar-Shalom's automatic client messaging launcher would determine the appropriate method for executing the user-defined actions based on network options, available messaging service, and call duration, as disclosed in Roujinsky to provide more actions that the calling party may predefine and increase the likelihood that incomplete calls will be detected.

145. A POSITA would have had a reasonable expectation of success in combining these teachings of Sar-Shalom and Roujinsky because each purports to solve these problems in similar and predictable ways such as by monitoring incomplete calls and automatically triggering events that are customized and defined by the caller in order to improve call completion of all types of unanswered calls without expensive equipment and central processing servers. Ex-1012, 15:5-15. A POSITA would have understood that they could have combined these teachings with little or no change in their respective functions because they teach compatible systems. A POSITA would have understood that these references disclose interrelated teachings using caller and recipient terminal devices that can execute software that is easily downloaded regardless of SIM version and size, with routine

network and data connections and technologies that would have been amendable to various well-understood and predictable combinations, as explained in further detail below.

2. Dependent Claims 3-5 and 13-15

- a. Claims 3/13: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said incomplete call is a call of a short duration that does not enable an intended communication to occur between said calling party and said called party.**

146. Sar-Shalom teaches Claims 3 and 13 wherein the incomplete call is a call of a short duration. Sar-Shalom's "availability detector 110 monitors the call from the time the ring tone is present until that call is answered or until a busy signal or any other indication of failure to complete the call is obtained." Ex-1013 ¶38.

147. An incomplete call is considered to be a voice call between two parties where "the call is not answered after a predetermined number of rings" or "the call connects to a voicemail box." Ex-1013 ¶¶36, 38. This is very similar to the '227 Patent, which explains that "the phrase 'incomplete call' [] refers to a call of a short duration, referred to as a 'short duration call', made by a calling party to a called party, that is considered incomplete as the duration of the call is too short for an intended communication to occur between the calling party and the called party." Ex-1001, 2:60-65. Sar-Shalom's incomplete call is a "call that is not answered after a predetermined number of rings" and, thus, the duration is "too short for an intended

communication to occur between the calling party and the called party.” Similarly, a call answered by voicemail would also be an incomplete call of short duration that does not enable intended communication between the parties just like the ’227 Patent’s “short duration answered call[]” which is a short “answered call attempt” such as a call that is answered but no intended communication has occurred between the calling party and the called party.” Ex-1001, 4:65-5:6; *see also* Ex-1051, 3:52-55 (determining that the callee is available based on call duration; e.g., a duration shorter than 10 seconds may be used as an indicator that the callee is not available). Sar-Shalom’s call that is answered by the voicemail would be an incomplete call where no intended communication has occurred between the calling party and the called party.

148. Claims 3/13 are also taught by Roujinsky, as described in Section XI.B.2.a. Specifically, Roujinsky’s call completion software registers an “uncompleted call” when there was “insufficient time elapsed after initiation of the call prior to hang-up to have allowed for the call to be answered.” Ex-1012, 16:9-23. This is just like the ’227 Patent’s short duration call, which is described as an unanswered call that “does not enable an intended communication to occur between the calling party and called party.” Ex-1001, 2:60-65, 11:44-46; *see also id.*, 4:66-5:2 (a call “less than about 2 seconds”), 7:11-14 (same), 16:23-26 (explaining that a duration of 2 seconds or more indicates that the call was answered). A POSITA

would have been motivated to combine Sar-Shalom-Roujinsky and had a reasonable expectation of success in doing so for the reasons described in Section XI.D.1.

- b. Claim 4: The computer implemented method of claim 1, wherein said one or more of said call completion actions are received by said call completion application based on a duration of said detected incomplete call**

Claim 14: The computer system of claim 11, wherein said action determination module is further configured to receive said one or more of said call completion actions based on a duration of said detected incomplete call.

149. Sar-Shalom-Roujinsky teaches Claims 4 and 14. Sar-Shalom teaches that the call completion actions are based on the duration of the incomplete call. For example, Sar-Shalom considers the duration of the call (i.e., predetermined number of rings or “if the call connects to a voicemail box”) when determining that the call is incomplete. Ex-1013 ¶¶36, 38. As I explained in the previous paragraph, in both scenarios, the call duration is “too short for an intended communication to occur between the calling party and the called party.” When the called party device does not connect and the call is determined incomplete, messaging initiator 120 launches messaging client 130 so that the calling party may send a message to the called party device, thus completing a “call completion action.” Ex-1013 ¶36. The user can be given a choice of responses, on the device’s input screen, such as to send a “SMS, EMS, or MMS message” comprising “text, audio clips, video clips, still images, multimedia presentations and voice.” *Id.* ¶¶40, 41, 45. Messaging client 130 (the

claimed “action determination module,” *see* Section XI.C.1.d (paragraph 128 discussing Element 11[e(i)]) thus receives a call completion action by receiving the message to send the called party device based on the call’s duration being too short.

150. Claims 4/14 are also taught by Roujinsky, as described in Section XI.B.2.b (paragraph 118). A POSITA would have been motivated to combine Sar-Shalom-Roujinsky and had a reasonable expectation of success in doing so for the reasons described in Section XI.D.1 (paragraphs 141-145).

c. Claims 5 and 15

(1) Elements 5[a]/15[a]: [The computer implemented method of claim 1 / The computer system of claim 11], wherein said action execution criteria comprise configuration of said call completion actions on said calling party device,

151. Sar-Shalom-Roujinsky teaches Elements 5[a]/15[a]. Sar-Shalom teaches action execution criteria comprising call completion actions configured on the caller’s device. Messaging client 130 uses “user-definable rules” that include “various criteria” set by the calling party on their communication device. Ex-1013 ¶46. Specifically, messaging initiator 120 provides additional information “provisioned by the user” required by messaging client 130 to apply the rules. *Id.* This allows the user to tailor the messages sent by messaging client 130 for different purposes. Thus, the calling party configures the “call completion actions” by

defining the factors that determine the appropriate method executing a call completion action in response to an incomplete call. *Id.*

- (2) **Elements 5[b]/15[b]: one of a presence and an absence of one or more of a message box and a messaging application for receiving media data on one of a network account of said called party and a called party device,**

152. Sar-Shalom-Roujinsky teaches Elements 5[b]/15[b]. Sar-Shalom's action execution criteria comprises the presence of a message box and messaging application on the called party's account and device. Specifically, Sar-Shalom's availability detector considers whether the call "connects to a voicemail box," when determining whether the destination device is unavailable. Ex-1013 ¶38. When the called party device does not connect, messaging initiator 120 launches the messaging client for the calling party to send a message to the unavailable device. *Id.* ¶¶38, 41. Therefore, messaging client 130 determines the action to be executed based on the presence of a messaging box on the called party device.

153. Similarly, Sar-Shalom's messaging initiator 120 detects whether a calling party device has "messaging capabilities," such as a messaging application for receiving media data, like a land-line telephone. *Id.* ¶39. If the calling party device does not have messaging capabilities, messaging client 130 is not launched thereby not sending a message to the called party from the calling party. *Id.*

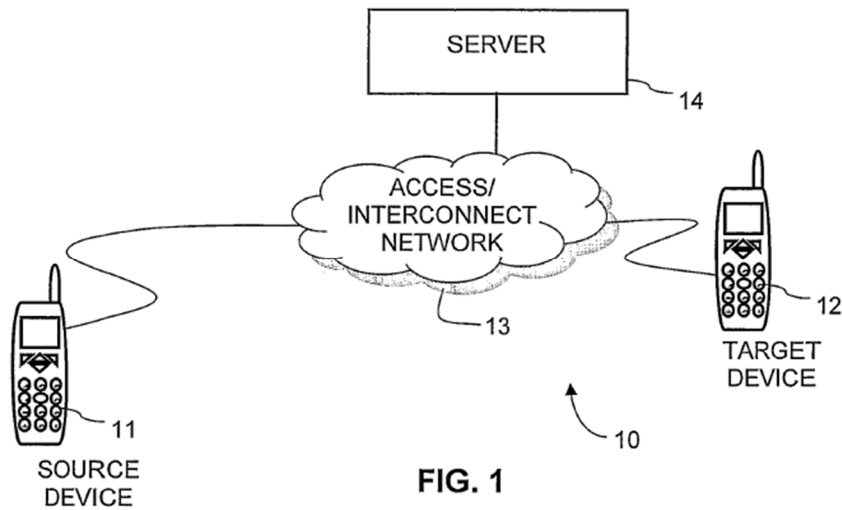
Therefore, messaging client 130 determines the action to be executed based on an application for receiving media data.

(3) Elements 5[c]/15[c]: one of a presence and an absence of a data connection from said calling party device to a network,

154. Sar-Shalom-Roujinsky teaches Elements 5[c]/15[c]. Sar-Shalom-Roujinsky teaches the action execution criteria comprises the data connection from the calling party device to a network. Specifically, Sar-Shalom explains that the communication device includes cellular telephones, IP telephones (portable and desktop), smart telephones and satellite telephones, “[s]ome fixed line telephone with SMS for wireline capabilities. Ex-1013 ¶37. Sar-Shalom explains that the availability detector “may have access to telephone network signaling information that can be used to determine if a called device is available.” Ex-1013 ¶49. A POSITA would thus have understood that Sar-Shalom considers whether there is a data connection for the communication device to execute the automatic client message launcher.

155. A POSITA would have further found it obvious, in view of Roujinsky, which teaches considering the data connection from the calling party device to a network. As shown in Roujinsky’s Figure 1, a call completion system is implemented between a “source telephone 11 and a target telephone 12 interconnected via a wireless network 13.” Ex-1012, 8:11-15. A POSITA thus would

have found it obvious that the action execution criteria considers the presence of a data connection from a calling party device to a network in order to execute the call completion actions. A POSITA would have been motivated to implement Sar-Shalom's action execution criteria comprising Roujinsky's explicit teaching of the presence of a data connection from the calling party device to a network in order to direct suitable messages to a callee device. Ex-1012, 8:21-23.



(4) Elements 5[d]/15[d]: one of a presence and an absence of a messaging service in a network component of said called party,

Elements 5[e]/15[e]: one of a presence and an absence of a messaging service in a network component of said calling party,

156. Sar-Shalom-Roujinsky teaches Elements 5[d]/15[d] and 5[e]/15[e]. Sar-Shalom teaches the action execution criteria comprises a messaging service in a network component of the called party and calling party. Sar-Shalom's messaging

initiator 120 detects whether the called party has “messaging capabilities” in order to determine the appropriate method of executing the call completion action. For example, messaging initiator 120 determines whether the called party is a land-line telephone and refrains from launching messaging client 130. Ex-1013 ¶39. Thus, the action to be executed comprises determining the presence or absence of the messaging service in the called party and calling party’s network component.

157. This would have been further understood as shown by Roujinsky. Roujinsky teaches that the “server” may compile a suitable notification and forward it to the called party by any “suitable additional network, possibly in accordance with a list of preferences pre-established by the called party.” Ex-1012, 17:14-16. Similarly, the “server stores a list of precompiled messages and is responsive to a request from the calling handset for selecting one of these messages.” *Id.*, 18:16-19. The message may be a pre-recorded voice message on a “voice server.” *Id.*, 18:19-21. A POSITA would have understood that the messaging service in a network component of the calling party is Roujinsky’s voice server that allows the calling party to access the messaging service. A POSITA would have been motivated to implement Sar-Shalom’s action execution criteria comprising Roujinsky’s explicit messaging service in a network in order to direct suitable messages to a called party device. Ex-1012, 8:21-23.

(5) Elements 5[f]/15[f]: reach of said messaging service to deliver said media data to one of said network component of said called party and said called party device,

158. Sar-Shalom-Roujinsky teaches Elements 5[f]/15[f]. Sar-Shalom-Roujinsky teaches that the action execution criteria comprises the reach of the messaging service to deliver the media data to the called party's network component or device. Specifically, Sar-Shalom's automatic message client launcher can be "switched on and off" by the party devices or the communications carrier. Ex-1013 ¶52. Sar-Shalom explains that the availability detector "may have access to telephone network signaling information that can be used to determine if a called device is available." Ex-1013 ¶49. When "an indication is received... that the destination device is offline" then the call is determined to be incomplete. *Id.* ¶38. Sar-Shalom's consideration of whether the called party is offline and unreachable shows that it considers the reach of the messaging service when determining the appropriate method to complete the call in order to ensure that the caller can send a message to the unavailable called party. *Id.* ¶61.

159. It would have been further obvious in view of Roujinsky. Roujinsky teaches that when the calling device calls over "a first network via the first communication channel," Roujinsky "maps the... address of the called party to a corresponding address on a second communication network that transports data over the second communication channel." Ex-1012, 17:2-7. The address may be to a

regular cellular telephone number allowing a SMS notification to be conveyed, or it may be an IP address allowing a notification to be sent over the packet network. Ex-1012, 17:2-7. Roujinsky's action execution criteria considers the "reach" of the calling device's messaging service by considering the availability the first and second communication channels to determine the appropriate method and network to deliver the message to the called party. A POSITA would have been motivated to include Roujinsky's reach into Sar-Shalom's action execution criteria for the same reasons stated in Section XI.D.1 (paragraphs 141-145).

(6) Elements 5[g]/15[g]: network options provided to said calling party device and said called party device by said network to which said calling party device and said called party device are connected, and

160. Sar-Shalom-Roujinsky teaches Elements 5[g]/15[g]. Sar-Shalom-Roujinsky teaches that the action execution criteria comprises the network options provided to said party devices. Sar-Shalom explains that the availability detector "may have access to telephone network signaling information that can be used to determine if a called device is available." Ex-1013 ¶49. A POSITA would have understood that accessing the telephone network signaling information factors in the network options available when determining what action to execute in response to an incomplete call (i.e., action execution criteria).

161. Moreover, a POSITA would have found this obvious, as disclosed in Roujinsky which teaches action execution criteria that factors in the available network options and the network connected to by devices. For example, Roujinsky determines that the calling party device is “not within coverage or is engaged on the telephony channel, thus preventing the call from being completed.” Ex-1012, 16:18-23. Roujinsky considers whether the calling party and the called party are “out of coverage,” “phone turned off,” experiencing “network problems” or whether the called party’s device is busy. Ex-1012, 1:6-16. When there is a network issue, Roujinsky sends a notification to the called party’s device as an SMS message “conveyed over the cellular network,” and allows “other types of message[s] sent over different networks such as text, email, and so on sent over the IP network.” Ex-1012, 17:17-30. Thus, a POSITA would have understood that the network connection options are factored into the action execution criteria when determining which user-defined action should be executed in response to the unanswered or incomplete call.

(7) Elements 5[h]/15[h]: call completion preferences configured by said call completion application on said calling party device.

162. Sar-Shalom-Roujinsky teaches Elements 5[h]/15[h]. Sar-Shalom teaches the action execution criteria comprises call completion preferences. The communication device has a “provision section” allowing the calling party to “set

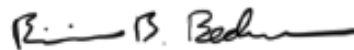
and/or change preferences.” Ex-1013 ¶52. Messaging client 130 further applies “user-definable rules” that include “various criteria” set by the calling party on their communication device. Ex-1013 ¶46. Specifically, messaging initiator 120 provides additional information “provisioned by the user” required by messaging client 130 to apply the rules. *Id.* This allows the automatic messaging client launcher to execute call completion actions based on the “call completion preferences” that the calling party defined.

XII. CONCLUSION

163. For the reasons above, it is my opinion that Claims 1-6, 8-15, and 17-19 of the '227 Patent are obvious based on each of the grounds specified in my Declaration.

164. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, 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.

Date: November 30, 2023



Dr. Benjamin B. Bederson