

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SLACK TECHNOLOGIES, LLC. and
SALESFORCE, INC.,
Petitioners,

v.

WRINKL, INC.,
Patent Owner

Case No.: IPR2024-01490
U.S. Patent No. 11,973,731

**DECLARATION OF DR. KEVIN C. ALMEROOTH
IN SUPPORT OF
PATENT OWNER RESPONSE**

TABLE OF CONTENTS

I.	INTRODUCTION	6
II.	SUMMARY OF OPINIONS	8
III.	QUALIFICATIONS	9
IV.	SUMMARY OF THE LAW OF ANTICIPATION AND OBVIOUSNESS	22
A.	Anticipation	22
B.	Obviousness.....	22
V.	LEVEL OF SKILL IN THE ART	23
VI.	CLAIM CONSTRUCTION	24
VII.	DESCRIPTION OF THE TECHNOLOGY.....	24
A.	Petitioner’s Statements Concerning the Accused Technology.....	25
B.	The ’731 Patent	29
C.	The Prior Art.....	35
1.	IRC	35
2.	Kakuta	37
D.	Matsumoto and Chang.....	47
VIII.	GROUND 1: KAKUTA DOES NOT ANTICIPATE THE CLAIMS	50
A.	Claim 1	50
1.	Kakuta Does Not Disclose Limitation 1[c]—A “first pane configured to display a plurality of channel identifiers”	50
2.	Kakuta Does Not Disclose Limitation 1[h]— “display a first user selectable control associated with said first messaging communication”	59
3.	Kakuta Does Not Disclose Limitation 1[i]—Kakuta Does Not Display a Third Pane “In Response To” a User Selection of “Said First User-Selectable Control”	71
4.	Kakuta Does Not Disclose Limitations 1[n] and 1[o]—Kakuta Does Not Have an Anchor Message and Has No “Second User-Selectable Control” and No “Indication” As Claimed	73

5. Kakuta Does Not Disclose Limitation 1[p]—The “Display” Button Does Not Cause a Portion of the Subsidiary Thread to be Displayed in the Third Pane	80
B. Claims 2-27	82
1. Claim 2	82
2. Claim 3	86
3. Claim 4	91
4. Claim 5	96
5. Claim 6	97
6. Claim 7	97
7. Claim 9	99
8. Claims 10, 11, & 12.....	100
9. Claims 15-19, 21-22, and 25-27.....	101
C. Claim 30 Is Not Anticipated By Kakuta	101
1. 30[c].....	101
2. 30[d]	101
3. 30[e].....	103
4. 30[f].....	103
IX. GROUND 2: THE PETITION FAILS TO ESTABLISH OBVIOUSNESS OVER KAKUTA	104
A. 1[c], 15[b], and 25[b]	105
B. 1[d], 15[c], and 25[c].....	110
C. 1[o], 15[n], and 25[n]	111
D. 1[i], 1[p], 3[a], 13[a], 15[h], 15[o], 17[a], 23[a], 25[h], 25[o], 27[a], and 28[a]113	
E. Claim 4	115
F. Claims 9, 12, and 21	117
G. Claim 13	119
H. Claim 14	121
I. Claims 23, 24, 28, and 29	122
J. Claim 30	123

X.	GROUND 3: KAKUTA/MATSUMOTO	125
A.	Matsumoto Fails to Fill the Gaps in Kakuta	125
B.	No Motivation to Combine	128
XI.	GROUND 4: CHANG DOES NOT ADDRESS THE DEFICIENCIES OF KAKUTA OR MATSUMOTO	132
A.	Claims 1, 15, and 25	132
1.	Chang Does Not Disclose a “Second User-Selectable Control”	132
2.	The Proposed Combination Would Not Result in The Claimed Invention 136	
B.	Claim 4 – The Combination Fails Because Petitioner Misstates the Teachings of Chang.....	139
C.	Claims 8 & 20 – The Combination Fails Because Petitioner Misstates the Teachings of Chang.....	140
D.	Claim 12 – The Combination Would Not Operate as Claimed.....	141
XII.	GROUND 5: CHANG DOES NOT FILL THE GAPS IN KAKUTA FOR CLAIM 30.....	143
XIII.	PETITIONER’S MOTIVATION SECTION (SECTION VII) APPLIES IMPROPER HINDSIGHT	145
XIV.	CONCLUSION	149

LIST OF EXHIBITS

Ex-2001	TechCrunch Article -- “Slack launches threaded messaging to take conversations off to the side” available at https://techcrunch.com/2017/01/18/slack-launches-threaded-messaging-to-take-conversations-off-to-the-side/
Ex-2002	Hubert Florin, <i>Threads in Slack A: Long Design Journey (Part 1 of 2)</i> , Slack Design, available at https://slack.design/articles/threads-in-slack-a-long-design-journey-part-1-of-2/
Ex-2003	----
Ex-2004	CV of Kevin C. Almeroth, Ph.D.
Ex-2005	<i>Wrinkl Wins MediaPost’s 2018 Appy Award</i> , May 16, 2018, available at https://www.globenewswire.com/news-release/2018/5/16/1507676/0/en/Wrinkl-Wins-MediaPost-s-2018-Appy-Award.html
Ex-2006	EXHIBIT A, <i>Initial Claim Chart Relating the Accused Product to the Asserted Claims of U.S. Patent No. 11,973,731</i>
Ex-2007	<i>Response to Non-Final Office Action dated March 9, 2020, filed June 8, 2020</i> in US Patent Application No. 16/399,349
Ex-2008	XChat: Channel List (nexlab.azzurra.org), available at https://commons.wikimedia.org
Ex-2009	Davey Alba, <i>Slack's New Threaded Messages Tame Your Meandering Chats</i> , WIRED, January 18, 2017
Ex-2010	U.S. Patent No. 11,539,649

I. INTRODUCTION

1. I have been retained on behalf of Wrinkl, Inc. in this proceeding, which concerns U.S. Patent No. 11,973,731, entitled “System and Methods for Subsidiary Channel-Based Thread Communications.” The patent issued on April 30, 2024, and has a priority chain leading back to various utility and provisional applications, the earliest of which is dated November 10, 2015. The art cited in this proceeding predates November 10, 2015.

2. For the Declaration, I will focus on figures 10A-I and 11 because they are better quality figures.

3. The Background section of the ’731 patent describes one of the impetuses leading to the invention:

Electronic chat communications (which include chat room communications, instant messaging and text messaging, etc.) are an increasingly popular way for two or more people to communicate on-line in virtual real time. For example, chat sessions for conveying a quick note are often used in the workplace in lieu of having a phone call with multiple individuals. In some situations, an ability to quickly communicate with more than one person using chat communications often saves time and avoids costly mistakes in collaborative efforts, such as team projects. When multiple people, however, are participating in a chat session that may be established for a large project, current electronic chat communications are limited due the sequential, read-

only, chronological presentation of exchanges between individuals or groups of individuals associated with the electronic chat session.

Ex-1001 ('731 patent), 2:45-62.

4. The claimed solution is recited in the claims, but at a high level, it involves specific requirements for creating a sidebar for side communications within a channel-based communication system in combination with various indicators and user-selectable controls relating to the sidebar communications. *See generally* Ex-1001.

5. I will refer to Slack and Salesforce.com collectively as “Petitioner” in this Declaration.

6. I have been asked to provide my technical review, analysis, and opinions regarding the merits of the Petitions for *inter partes* review of the '731 patent. My analysis assumes a priority date of November 10, 2015 (the earliest date in the priority chain), which is the date used by the Petition, but my analysis would not change if the date were either a few years earlier or later.

7. The primary reference on which Petitioner relies is U.S. Patent No. 6,630,944, which issued to Fujitsu Limited in 2003, naming Jun Kakuta and two others as inventors. It is entitled “Display Control Method and Device.” In this proceeding, the patent is referred to as the “Kakuta” reference. Kakuta discloses a messaging system based on Internet Relay Chat (“IRC”) that provide threads with

a thread window separate from the main chat window.

8. I have no financial interest in any party to this proceeding or in the outcome of this proceeding. I am being compensated for my work as an expert on an hourly basis. My compensation is not dependent on the outcome of these proceedings or the content of my opinions.

II. SUMMARY OF OPINIONS

9. Although Petitioner asserts that Kakuta anticipates the majority of the claims of the '731 patent, Kakuta and the '731 patent are considerably different, and Kakuta lacks a number of limitations of the claims. Petitioner's backup argument is that Kakuta, either alone or in combination with one or two other references (Matsumoto (Ex-1006) and Chang (Ex-1007)) renders the '731 patent claims invalid for obviousness. However, as I explain later in this Declaration, a person of ordinary skill in the art (which I will call a "Skilled Artisan") would not have modified Kakuta to arrive at the claimed invention. The reasons provided by Petitioner for the alleged modifications, which are largely conclusory, are based on hindsight. In addition, even assuming the teachings of Matsumoto and Chang were applied to Kakuta, the result would be something other than the invention of the '731 patent.

10. Essentially, Petitioner first asks what changes would have to be made to Kakuta to meet the claim language, and then states that those changes would

have been obvious. But Petitioner fails to articulate a valid reason that would have led someone to make the proposed changes. Further, although Petitioner cites to the Declaration of Dr. Cockburn in alleged support, Dr. Cockburn largely just repeats the petition text and fails to provide reasoned analysis to support the conclusory statements found in both the Petition and the Cockburn declaration.

III. QUALIFICATIONS

11. I am currently a Professor Emeritus in the Department of Computer Science at the University of California, Santa Barbara (UCSB). While at UCSB, I held faculty appointments and was a founding member of the Computer Engineering (CE) Program, Media Arts and Technology (MAT) Program, and the Technology Management Program (TMP). I also served as the Associate Director of the Center for Information Technology and Society (CITS) from 1999 to 2012. I have been a faculty member at UCSB since July 1997.

12. I hold three degrees from the Georgia Institute of Technology: (1) a Bachelor of Science degree in Information and Computer Science (with minors in Economics, Technical Communication, and American Literature) earned in June 1992; (2) a Master of Science degree in Computer Science (with specialization in Networking and Systems) earned in June 1994; and (3) a Doctor of Philosophy (Ph.D.) degree in Computer Science (Dissertation Title: Networking and System Support for the Efficient, Scalable Delivery of Services in Interactive Multimedia

System, minor in Telecommunications Public Policy) earned in June 1997. During my education, I took a wide variety of courses as demonstrated by my minors. My undergraduate degree also included a number of courses more typical of a degree in electrical engineering including digital logic, signal processing, and telecommunications theory.

13. One of the major concentrations of my research over the past 30+ years has been the delivery of multimedia content and data between computing devices, including through various network architectures. In my research, I have studied large-scale content delivery systems, and the use of servers located in a variety of geographic locations to provide scalable delivery to hundreds or thousands of users simultaneously. I have also studied smaller-scale content delivery systems in which content is exchanged between individual computers and portable devices. My work has emphasized the exchange of content more efficiently across computer networks, including the scalable delivery of content to many users, mobile computing, satellite networking, delivering content to mobile devices, and network support for data delivery in wireless networks.

14. In 1992, the initial focus of my research was on the provision of interactive functions (e.g., VCR-style functions like pause, rewind, and fast-forward) for near video-on-demand systems in cable systems; in particular, how to aggregate requests for movies at a cable head-end and then how to satisfy a

multitude of requests using one audio/video stream broadcast to multiple receivers simultaneously. This research has continually evolved and resulted in the development of techniques to scalably deliver on-demand content, including audio, video, web documents, and other types of data, through the Internet and over other types of networks, including over cable systems, broadband telephone lines, and satellite links.

15. An important component of my research has been investigating the challenges of communicating multimedia content, including video, between computers and across networks including the Internet. Although the early Internet was used mostly for text-based, non-real time applications, the interest in sharing multimedia content, such as video, quickly developed. Multimedia-based applications ranged from downloading content to a device to streaming multimedia content to be instantly used. One of the challenges was that multimedia content is typically larger than text-only content, but there are also opportunities to use different delivery techniques since multimedia content is more resilient to errors. I have worked on a variety of research problems and used a number of systems that were developed to deliver multimedia content to users. One content-delivery method I have researched is the one-to-many communication facility called “multicast,” first deployed as the Multicast Backbone, a virtual overlay network supporting one-to-many communication. Multicast is one technique that can be

used on the Internet to provide streaming media support for complex applications like video-on-demand, distance learning, distributed collaboration, distributed games, and large-scale wireless communication. The delivery of media through multicast often involves using Internet infrastructure, devices and protocols, including protocols for routing and TCP/IP.

16. Starting in 1997, I worked on a project to integrate the streaming media capabilities of the Internet together with the interactivity of the web. I developed a project called the Interactive Multimedia Jukebox (IMJ). Users would visit a web page and select content to view. The content would then be scheduled on one of a number of channels, including delivery to students in Georgia Tech dorms delivered via the campus cable plant. The content of each channel was delivered using multicast communication.

17. In the IMJ, the number of channels varied depending on the capabilities of the server including the available bandwidth of its connection to the Internet. If one of the channels was idle, the requesting user would be able to watch their selection immediately. If all channels were streaming previously selected content, the user's selection would be queued on the channel with the shortest wait time. In the meantime, the user would see what content was currently playing on other channels, and because of the use of multicast, would be able to join one of the existing channels and watch the content at the point it was currently

being transmitted.

18. The IMJ service combined the interactivity of the web with the streaming capabilities of the Internet to create a jukebox-like service. It supported true Video-on-Demand when capacity allowed, but scaled to any number of users based on queuing requested programs. As part of the project, we obtained permission from Turner Broadcasting to transmit cartoons and other short-subject content. We also connected the IMJ into the Georgia Tech campus cable television network so that students in their dorms could use the web to request content and then view that content on one of the campus's public access channels.

19. More recently, I have also studied issues concerning how users choose content, especially when considering the price of that content. My research has examined how dynamic content pricing can be used to control system load. By raising prices when systems start to become overloaded (*i.e.*, when all available resources are fully utilized) and reducing prices when system capacity is readily available, users' capacity to pay as well as their willingness can be used as factors in stabilizing the response time of a system. This capability is particularly useful in systems where content is downloaded or streamed on-demand to users.

20. As a parallel research theme, starting in 1997, I began researching issues related to wireless devices and sensors. In particular, I was interested in showing how to provide greater communication capability to "lightweight

devices,” *i.e.*, small form-factor, resource-constrained (*e.g.*, CPU, memory, networking, and power) devices. Starting in 1998, I published several papers on my work to develop a flexible, lightweight, battery-aware network protocol stack. The lightweight protocols we envisioned were similar in nature to protocols like Bluetooth, Universal Plug and Play (UPnP) and Digital Living Network Alliance (DLNA).

21. From this initial work, I have made wireless networking—including ad hoc, mesh networks and wireless devices—one of the major themes of my research. My work in wireless networks spans the protocol stack from applications through to the encoding and exchange of data at the data link and physical layers.

22. At the application layer, even before the large-scale “app stores” were available, my research looked at building, installing, and using apps for a variety of purposes, from network monitoring to support for traditional computer-based applications (*e.g.*, content retrieval) to new applications enabled by ubiquitous, mobile devices. For example, my research has looked at developing applications for virtually exchanging and tracking “coupons” through “opportunistic contact” among mobile wireless devices (*i.e.*, communication among devices moving into communication range with each other). In many of the courses I have taught there is a project component. Through these projects I have supervised numerous efforts to develop new “apps” for download and use across a variety of mobile platforms.

23. Toward the middle of the protocol stack, my research has also looked to build wireless infrastructure support to enable communication among a set of mobile devices unaided by any other kind of network infrastructure. These kinds of networks are useful either in challenged network environments (e.g., when a natural disaster has destroyed existing infrastructure) or when suitable support for network communication never existed. The deployment of such networks (or even the use of traditional network support) are critical to support services like disaster relief, catastrophic event coordination, and emergency services deployment.

24. Yet another theme is monitoring wireless networks, in particular different variants of IEEE 802.11 compliant networks, to (1) understand the operation of the various protocols used in real-world deployments, (2) use these measurements to characterize use of the networks and identify protocol limitations and weaknesses, and (3) propose and evaluate solutions to these problems. I have successfully used monitoring techniques to study wireless data link layer protocol operation and to improve performance by enhancing the operation of such protocols. For wireless protocols, this research includes functions like network acquisition and channel bonding.

25. One theme in my wireless network research has been cross-layer solutions and innovations. As mentioned above, with greater wireless device use and network support, we envisioned new application paradigms and services, for

example, when mobile devices come into contact with each other. Instead of relying on existing infrastructure to relay communication, the devices are able to discover each other and communicate directly. Other examples include discovering and using location information to enhance users' experiences. Network support and novel applications use a variety of network architectures supporting users on foot, in vehicles, and across varying terrains and environments. Finally, we studied how communication efficiency can be supported through intelligent handoffs as well as location and movement prediction.

26. Protecting networks, including their operation and content, has been an underlying theme of my research almost since the beginning of my research career. Starting in 2000, I have been involved in several projects that specifically address security, network protection, and firewalls. After significant background work, a team on which I was a member successfully submitted a \$4.3M grant proposal to the Army Research Office (ARO) at the Department of Defense to propose and develop a high-speed intrusion detection system. Key aspects of the system included associating streams of packets and analyzing them for viruses and other malware. Once the grant was awarded, we spent several years developing and meeting the milestones of the project. A number of my students worked on related projects and published papers on topics ranging from intrusion detection to developing advanced techniques to be incorporated into firewalls. I have also used

firewalls, including their associated malware detection features, in developing techniques for the classroom to ensure that students are not distracted by online content.

27. My recent work ties some of the various threads of my past research together. I have investigated content delivery in online social networks and proposed reputation management systems in large-scale social networks and marketplaces. On the content delivery side, I have looked at issues of caching and cache placement, especially when content being shared and the cache have geographical relevance. We were able to show that effective caching strategies can greatly improve performance and reduce deployment costs. Our work on reputation systems showed that reputations have economic value, and as such, creates a motivation to manipulate reputations. In response, we developed a variety of solutions to protect the integrity of reputations in online social networks. The techniques we developed for content delivery and reputation management were particularly relevant in peer-to-peer communication and recommendations for downloadable “apps.”

28. As an important component of my research program, I have been involved in the development of academic research into available technology in the marketplace. One aspect of this work is my involvement in the Internet Engineering Task Force (IETF). The IETF is a large and open international

community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. I have been involved in various IETF groups including many content delivery-related working groups like the Audio Video Transport (AVT) group, the MBone Deployment (MBONED) group, Source Specific Multicast (SSM) group, the Inter-Domain Multicast Routing (IDMR) group, the Reliable Multicast Transport (RMT) group, the Protocol Independent Multicast (PIM) group, etc. I have also served as a member of the Multicast Directorate (MADDOGS), which oversaw the standardization of all things related to multicast in the IETF. Finally, I was the Chair of the Internet2 Multicast Working Group for seven years.

29. My involvement in the research community extends to leadership positions for several academic journals and conferences. I am the co-chair of the Steering Committee for the ACM Network and System Support for Digital Audio and Video (NOSSDAV) workshop and on the Steering Committees for the International Conference on Network Protocols (ICNP), ACM Sigcomm Workshop on Challenged Networks (CHANTS), and IEEE Global Internet (GI) Symposium. I have served or am serving on the Editorial Boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Network, ACM Computers in Entertainment, AACE Journal of Interactive Learning Research (JILR), and ACM Computer Communications Review. I have

co-chaired a number of conferences and workshops including the IEEE International Conference on Network Protocols (ICNP), IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), International Conference on Communication Systems and Networks (COMSNETS), IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS), the International Workshop On Wireless Network Measurement (WiNMee), ACM Sigcomm Workshop on Challenged Networks (CHANTS), the Network Group Communication (NGC) workshop, and the Global Internet Symposium, and I have served on the program committees for numerous conferences.

30. Furthermore, in the courses I taught at UCSB, a significant portion of my curriculum covered aspects of the Internet and network communication including the physical and data link layers of the Open System Interconnect (OSI) protocol stack, and standardized protocols for communicating across a variety of physical media such as cable systems, telephone lines, wireless, and high-speed Local Area Networks (LANs). The courses I have taught also cover most major topics in Internet communication, including data communication, multimedia encoding, and mobile application design. My research and courses have covered a range of physical infrastructures for delivering content over networks, including cable, Integrated Services Digital Network (ISDN), Ethernet, Asynchronous

Transfer Mode (ATM), fiber, and Digital Subscriber Line (DSL). For a complete list of courses I have taught, see my curriculum vitae (CV).

31. In addition, I co-founded a technology company called Santa Barbara Labs that was working under a sub-contract from the U.S. Air Force to develop very accurate emulation systems for the military's next generation internetwork. Santa Barbara Labs' focus was in developing an emulation platform to test the performance characteristics of the network architecture in the variety of environments in which it was expected to operate, and, in particular, for network services including IPv6, multicast, Quality of Service (QoS), satellite-based communication, and security. Applications for this emulation program included communication of a variety of multimedia-based services, including video conferencing and video-on-demand.

32. In addition to having co-founded a technology company myself, I have worked for, consulted with, and collaborated with companies for nearly 30 years. These companies range from well-established companies to start-ups and include IBM, Hitachi Telecom, Turner Broadcasting System (TBS), Bell South, Digital Fountain, RealNetworks, Intel Research, Cisco Systems, and Lockheed Martin.

33. Through my graduate education, leadership with CITS, involvement in TMP, role in the development of the Internet2 infrastructure, and consulting

with ISPs, I have gained a strong understanding in the role of the Internet in our society and the challenges of deploying large-scale production networking infrastructure. CITS, since its inception, has looked at the role of the Internet in society, including how the evolution of technology has created communication opportunities and challenges, including, for example, through disruptive technologies like P2P. TMP looks to focus on non-purely technical issues, including, for example, state-of-the-art business methods, strategies for successful technology commercialization, new venture creation, and best practices for fostering innovation. Through my industry collaborations and Internet2 work, I have developed significant experience in the challenges of deploying, monitoring, managing, and scaling communication infrastructure to support evolving Internet services like streaming media, conferencing, content exchange, social networking, and e-commerce.

34. I am a Member of the Association of Computing Machinery (ACM) and a Fellow of the Institute of Electrical and Electronics Engineers (IEEE).

35. Additional details about my employment history, fields of expertise, courses taught, and publications are further included in my CV filed as Exhibit 2004 in this proceeding.

IV. SUMMARY OF THE LAW OF ANTICIPATION AND OBVIOUSNESS

A. Anticipation

36. In this proceeding, Petitioner relies on patents and published patent applications. As is relevant here, I have been informed that a claim is invalid under 35 U.S.C. § 102(a) if the claimed invention was patented or described in a printed publication (which would include published patent applications) before the effective filing date of the claimed invention.

37. I understand that there is an exception to this rule under 35 U.S.C. § 102(b) for prior disclosures of the inventor of the challenged patent. That exception does not apply here.

B. Obviousness

38. I have been informed that a patent claim is invalid as “obvious” under 35 U.S.C. § 103 in light of one or more prior art references if it would have been obvious to a person of ordinary skill in the art at the time the invention was made, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called “secondary considerations” of non-obviousness, which include: (i) “long felt need” for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) “copying” of the claimed invention by others.

39. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference or references themselves, the knowledge of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

V. LEVEL OF SKILL IN THE ART

40. A person of ordinary skill in the art (which I will refer to as a “Skilled Artisan”) would have a bachelor’s degree in computer science or a related degree and two years of experience designing software for communications systems. More experience can compensate for less education and more education can compensate for less experience. I have more than ordinary skill and can opine on the issues raised in the Petition.

41. The Petitioner defines a Person Having Ordinary Skill in the Art

(“PHOSITA”) with respect to the ’731 patent as one with experience with “graphical user interfaces” and “at least two years of experience in the graphical user interface field.” However, when discussing the technology, the Petition refers to the “field of electronic chat communications” and the “field of messaging” Petition, 6-7 & 117. Thus, I disagree that the level of skill should be tied to the “user interface field.” Even assuming Petitioner were correct, I would still have more than ordinary skill and be qualified to opine on the issues raised in the Petition.

VI. CLAIM CONSTRUCTION

42. I have been asked to apply the ordinary meaning of claim terms within the context of the ’731 patent. I have not been provided any claim constructions to consider.

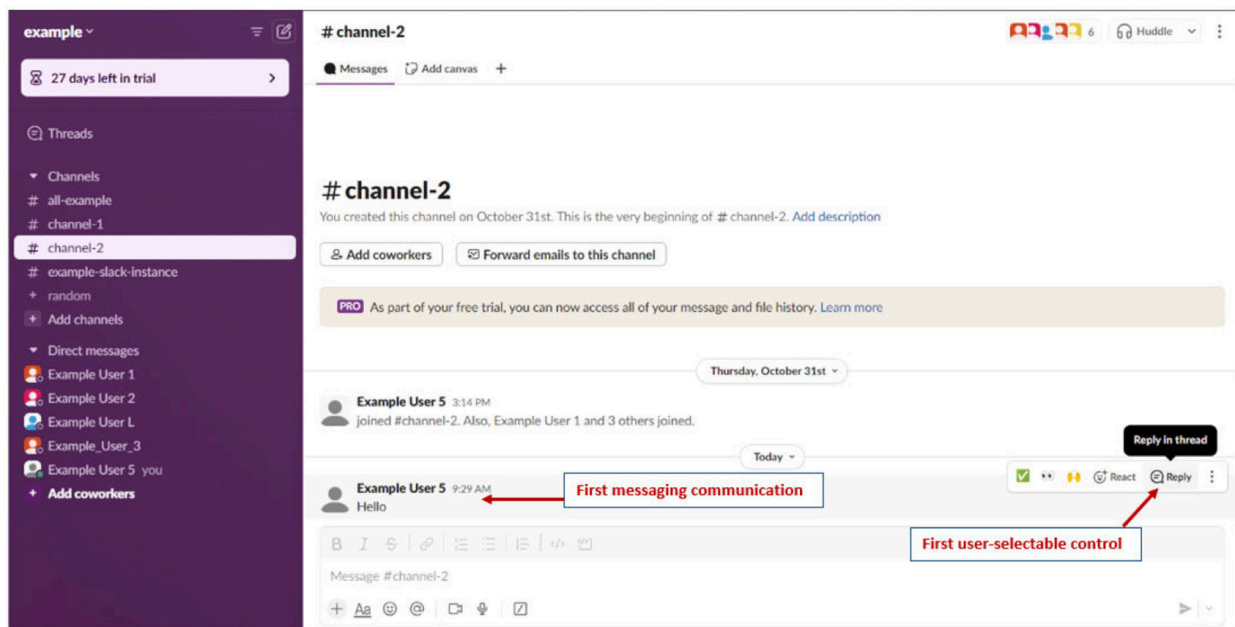
VII. DESCRIPTION OF THE TECHNOLOGY

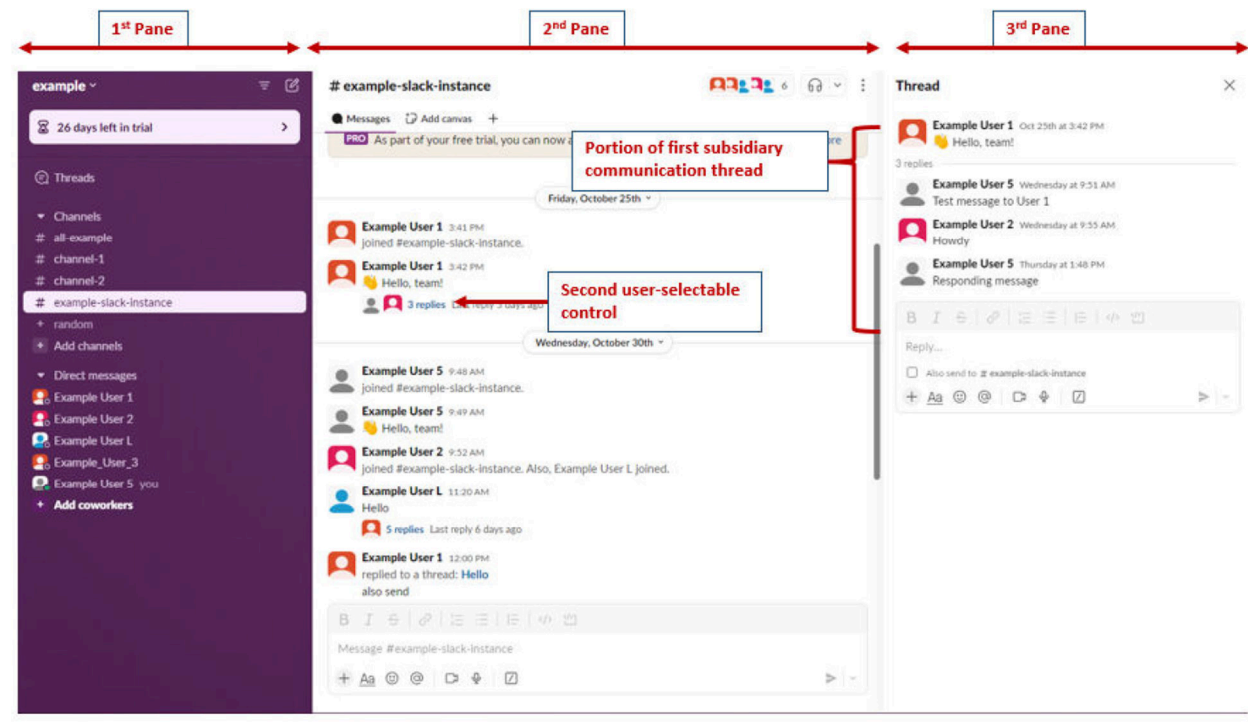
43. The ’731 patent provides a group-messaging system with an improved user interface that greatly facilitates the spawning of, and user interaction with, subsidiary threads or “sidebars” within a larger group channel. *See, e.g.*, Ex-1001, 16:59-17:19. In the ’731 patent, a thread-spawning first messaging communication is displayed in the main chat channel in a manner so as to be easily identifiable as an anchor message, and in a manner that invites further user interaction with controls associated with the first messaging communication to display at least a

portion of the thread associated with that message.

A. Petitioner's Statements Concerning the Accused Technology

44. I understand that Wrinkl, Inc. has asserted in District Court litigation that the '731 patent is infringed by the Slack messaging platform. Ex-2006 (Wrinkl Infringement Contentions, Ex-A). The following annotated screenshots from a Slack interface illustrate how Slack implements its “Threads” feature: Slack has (a) three panes (a first pane including a plurality of channels, a second pane with messages from a selected channel, and a third pane to display a thread that is opened via a first user-selectable control) and (b) an anchor message in the third pane that is also displayed in the second pane in association with an indication and a second user-selectable control that causes the third pane to open if the third pane is not already open. Ex-2006, 10, 20:





45. I discuss details of the '731 patent in the next section, and the similarities between the '731 patent and the Threads feature of Slack are numerous. Based on my review of the infringement contentions concerning claim 1, it is apparent that Slack's Threads feature practices each limitation of the claims asserted in the litigation and as charted in the infringement contention claim chart. I do not believe Petitioner provided non-infringement contentions in the litigation.

46. Given that Slack's Threads feature appears to practice (based on my review of the infringement contentions) the claims of the '731 patent, Petitioner's comments regarding the Threads feature provides some evidence to guard against hindsight bias when considering the question of obviousness of the invention of the '731 patent.

47. For example, in an article published in TechCrunch¹ entitled “Slack launches threaded messaging to take conversations off to the side,” the Threads feature is described as follows:

With threads, users can pop out of the chaos of an ongoing chat and pin bits of conversations off to the side that they can address on a more regular basis. Users can jump in and out of their existing threads much like they do channels and direct messages, and when they want to share new comments back into a chat stream, they can do so by hitting a small checkbox at the bottom of their new message. . . . Starting a thread is as simple as clicking on a button next to a message in a room, much like you would add a reaction, and then it peels off into a new conversation within the flexpane.

Ex-2001 (TechCrunch article).

48. In the article, Slack’s VP of Product commented on the newly released Threads feature, stated: “We’re giving people the choice to have sidebars, where they don’t feel like the only option is to take things into a private channel or direct message.” Slack’s VP of Product further discussed how, although the thread sidebar looks simple in retrospect, “it was nontrivial and non-obvious”:

¹ TechCrunch is a well-known online news platform focused on the technology, and in particular emerging technologies. I was familiar with TechCrunch prior to this proceeding, and have read its articles in the past. As a news organization, TechCrunch provide articles with quotes from its sources. I have no reason to doubt that the quotes of Slack employees is accurate.

Threads is an example of where we've been hearing this demand for customers, and felt the need to have sidebar conversations. But it was nontrivial and non-obvious on how to get it there. It's one of our values as a company — not just build products, but everything we do we value our craft.

Ex-2001 (TechCrunch Article).

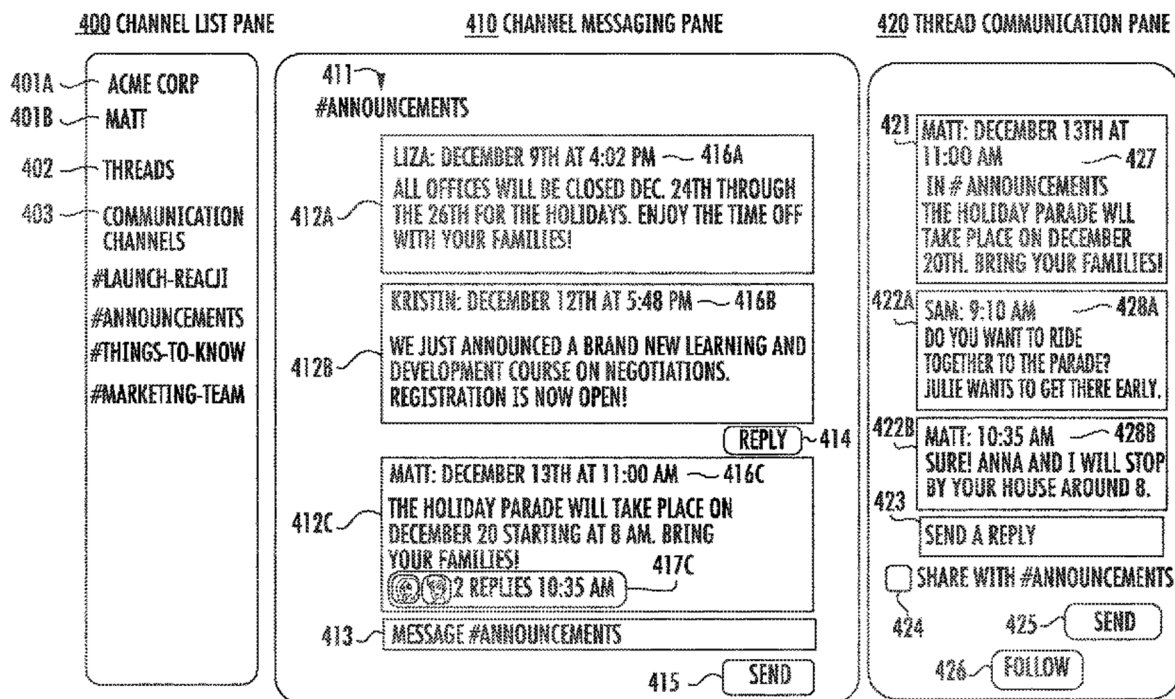
49. Another article from Wired Magazine² interviewed the Slack product lead for Threads and reported that the Threads feature went through at least four iterations and two years of development. Ex-2009 (Wired Article).

50. These comments from Slack roughly eight years ago and shortly after Slack's release of its accused Threads feature is evidence of the non-obviousness of the invention of the '731 patent. At the time Slack believed the inventive feature was not obvious, and indeed it took Slack years to figure out how best to provide thread functionality to users.

51. In addition, Petitioner's own patent filings suggests that the invention is non-obvious. For example, Slack Technologies, LLC filed for, and received, U.S. Patent No. 11,539,649, entitled "Group-Based Communication Interface With Subsidiary Channel-Based Thread Communications." Ex-2010. The patent

² Wired Magazine is a well-known source of technology-related articles. I am familiar with Wired Magazine and have read articles in Wired over the years. As with TechCrunch, I have no reason to doubt the accuracy of the quotes in the article.

concerns the infringing Threads feature of Slack, and the figure on the cover of the patent highlights the three panes, with channels in the first pane, thread indicators and second user-selectable controls in the second pane, and thread messages in the third pane:



Ex-2010 (US Patent No. 11,539,649).

52. Claim 1 of the Petitioner's '649 patent is identical to claim 30 of the '731 patent. It appears that claim 1 is still in force, and has not been disclaimed. Thus, Petitioner's position on obviousness in this proceeding is contrary with the position it (or its sister entity) has taken with regard to its own patents.

B. The '731 Patent

53. The '731 patent explains that messaging systems suffer from "a loss

of context and continuity of communication” and have a “problem of reply-to-all clutter,” noting that “once a message is off the top of the screen, it is rarely seen again.” The ’731 patent offers a solution that responds to these problems “while retaining context and continuity (place, time, meaning), and the flow of the conversation for each unique user.” Ex-1001, 1:43-66. For example, a user may view private messages isolated in a third pane while simultaneously viewing the private messages in context in the main messaging pane. Thus, a user “can reply directly to someone in a group chat that can be kept private from other group members without having to engage the entire group, while retaining context and relativity to the broader conversation for the participants of the 1 to 1 conversation.” *Id.*, 16:65-17:2.

54. For example, Figure 10A depicts a user hovering over an arrow control associated with a specific message in the main channel screen to start a sidebar conversation:



FIG. 10A

55. This arrow is an exemplary “first user-selectable control.” This control is associated with a specific message in the channel pane as would be readily apparent to a Skilled Artisan by the relative location of the control to the Ali T. message at 8:42 AM. In Figure 10A, the Ali T. message has also been highlighted with the highlighting extending to and touching the arrow control. When a user presses the arrow control, the Ali T. message is selected as the anchor message for the thread and a sidebar pane opens (if not already open) to display the thread. Thus, the act of clicking the arrow button accomplishes the selection of the message and the opening of the sidebar pane (the third pane) in a single step.

Additional user actions or selections are not necessary.

56. In the exemplary system, after pressing the arrow control associated with the message, a sidebar is started as shown in FIG. 10B:



FIG. 10B

57. Figure 10G (below) depicts the screen of a user who is a member of a sidebar, but whose sidebar pane is not open. Figure 10G's has a sidebar conversation that is based on Frank M.'s message at 8:30 AM, which acts as the anchor message for the sidebar. In the second pane (the main chat pane), Frank M.'s message has an indicator identifying it as an anchor message of a sidebar and a user-selectable control to cause the sidebar pane to open.

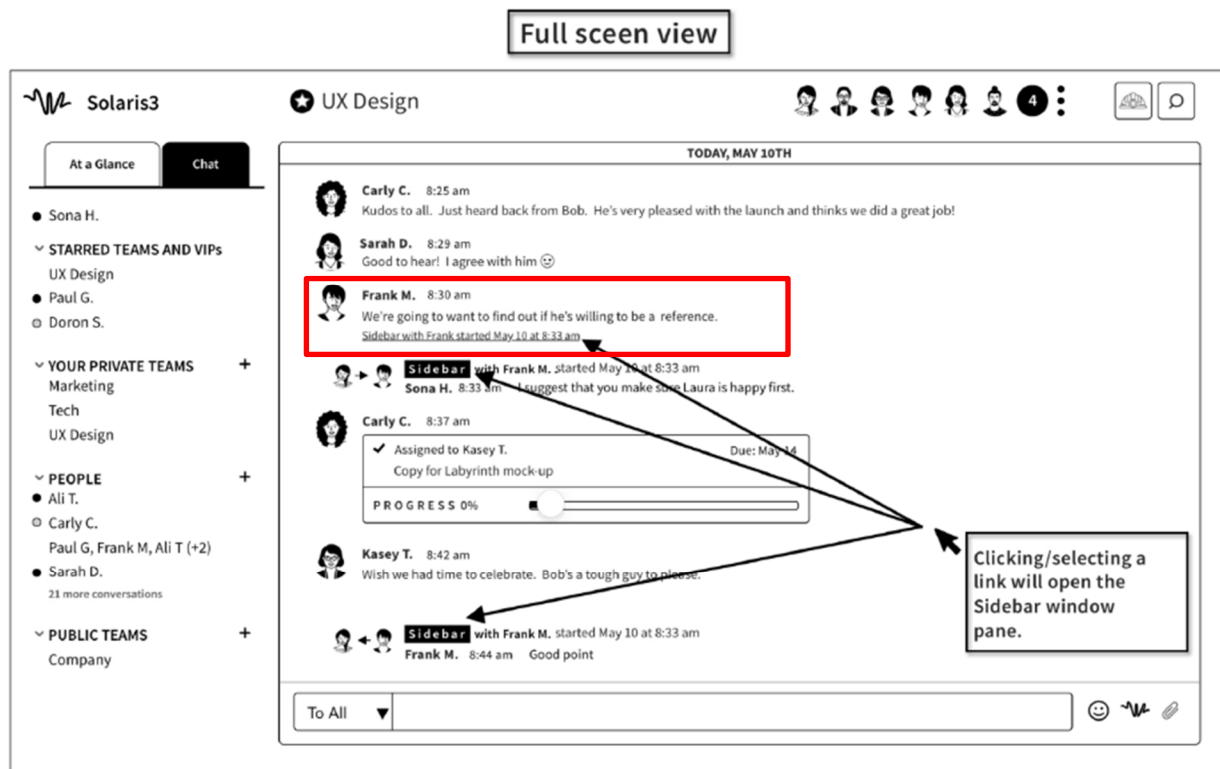


FIG. 10G

58. The indication is a note stating “Sidebar with Frank started May 10 at 8:33 am.” Additional messages entered within the sidebar thread may be provided to the user in the main chat pane, and are identified as sidebar messages in the figure above by the conspicuous “Sidebar” symbol and also by being displayed using indented text.

59. The indications also have links that provide controls that when clicked will cause the sidebar pane to open if it is not open, as depicted in FIG. 10H:

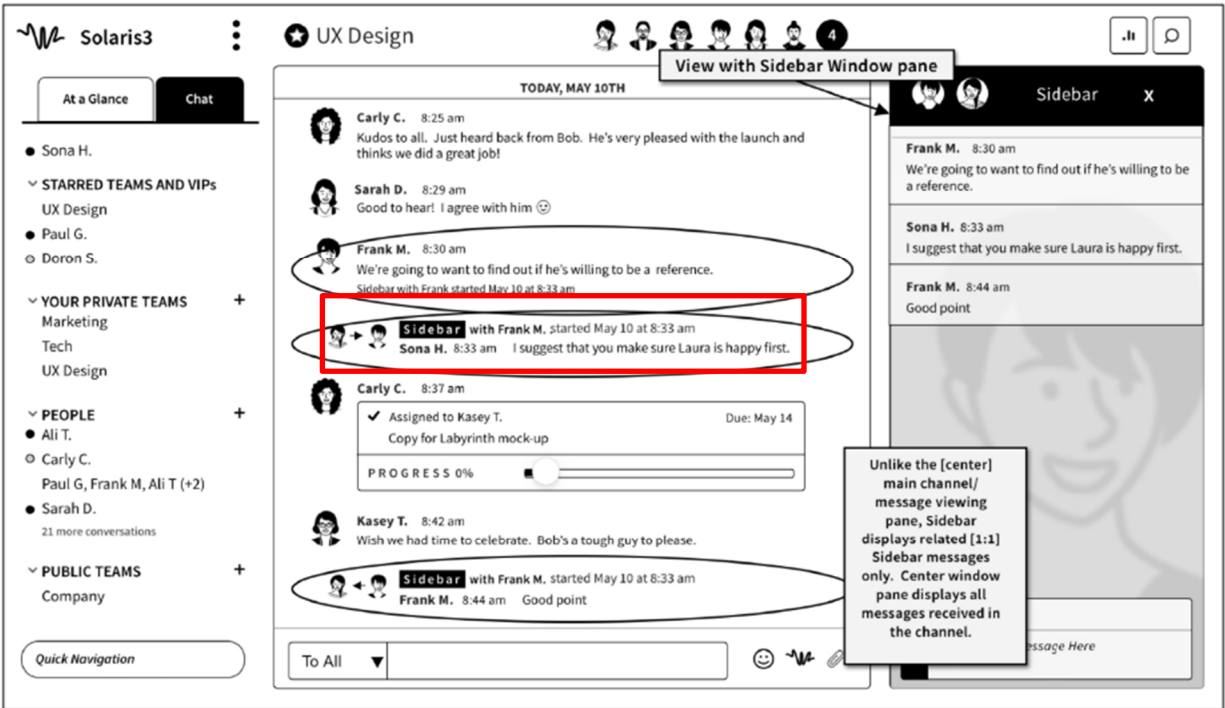


FIG. 10H

60. As shown in FIG. 10H (above), various channel titles are provided in the left-most pane (a “first pane”), the messages in the selected channel (“UX Design”) are shown in the middle pane (a “second pane”), and the sidebar is shown in the right-most pane (a “third pane”). The sidebar shown here would have been created when Sona H. clicked the arrow control associated with Frank M.’s message, similar to that shown in FIG. 10A (a “first user-selectable control”). The selected message—Frank M.’s message at 8:30 am—became the “anchor message” for the sidebar. Frank M. would see the indicator on his screen that a sidebar was started and can open the sidebar pane by clicking the associated link (a “second user-selectable control”) to cause the sidebar pane to open.

C. The Prior Art

1. IRC

61. Kakuta builds on traditional Internet Relay Chat (“IRC”) technology, and I therefore provide a brief background concerning IRC. I will refer to Exhibit 1014 to provide this background, which is an exhibit provided by Petitioner. My discussion here is fully consistent with Exhibit 1014 and should be non-controversial.

62. In IRC, users could join various topic-based “channels.” As explained in Petitioner’s own Exhibit 1014, IRC supported “community communication through channels where a participant can post a line of text (a ‘turn’) to the channel space which is visible to all other participants.” Ex-1014, 4. “The basic chat-channel user interface adopted and popularized by IRC” was “utilized by many of the other open chat-systems” developed in the 2000s. *Id.* A “typical IRC User Interface” is reproduced here from Exhibit 1014:

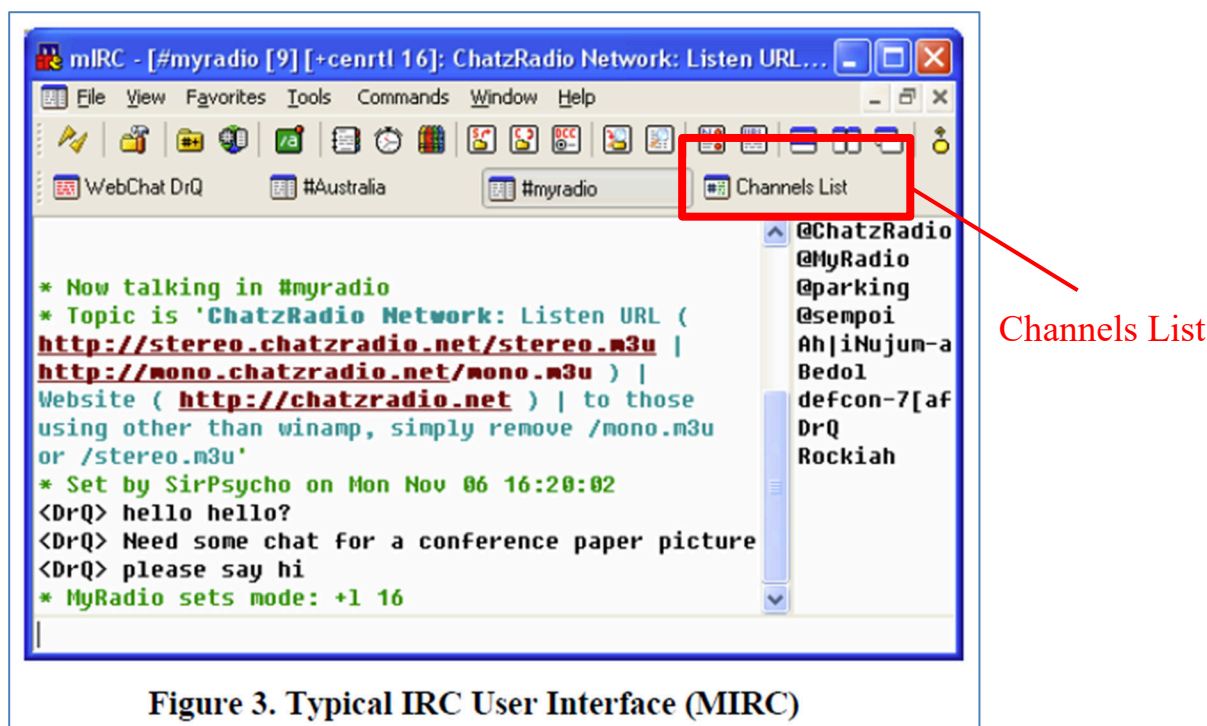
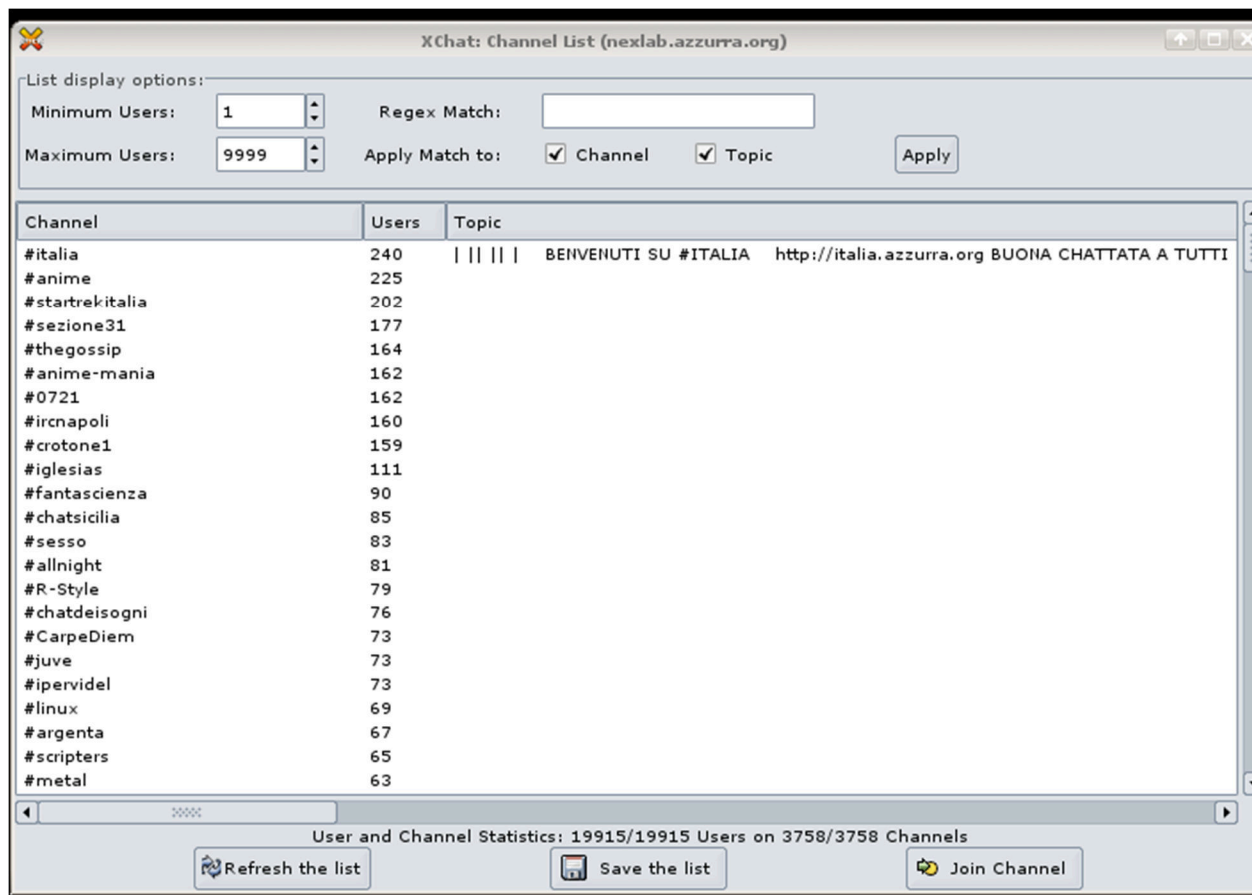


Figure 3. Typical IRC User Interface (MIRC)

Id. As shown here, after a channel is selected, in this case “#myradio” (channels in IRC begin with the hash symbol), a chat window (left) and user window (right) are displayed. Other available channels are not displayed in the IRC chat window.

63. Instead, IRC systems typically had a channel list button (shown in the screenshot above) that would switch to a new screen with a list of channels. The exhibit provided by Petitioner does not have a screen shot of the Channels List screen, and therefore I provide a screen shot obtained from “wikimedia.org.” The following screen shot is described as one from the “XChat” messaging platform. It provides a list of channels and the number of users who are members of each respective channel. The screen shot further states that there are 3,758 channels at the time of the screen shot:



Ex-2008,

(https://commons.wikimedia.org/wiki/Category:IRC_screenshots#/media/File:Azzurra_lista_canali.png)

2. Kakuta

64. Kakuta “relates to technology that makes conversational display easy to view, in a textual conversation system such as IRC (Internet Relay Chat).” Ex-1005, 1:9-19. Kakuta is a channel-based system, and explains that “a ‘channel’ means a virtual space that chat clients share in IRC.” *Id.* Kakuta states that “in IRC creating channels that differ according to topic is easy. In reality, however,

multiple topics often break out within a single channel.” Ex-1005, 1:56-58.

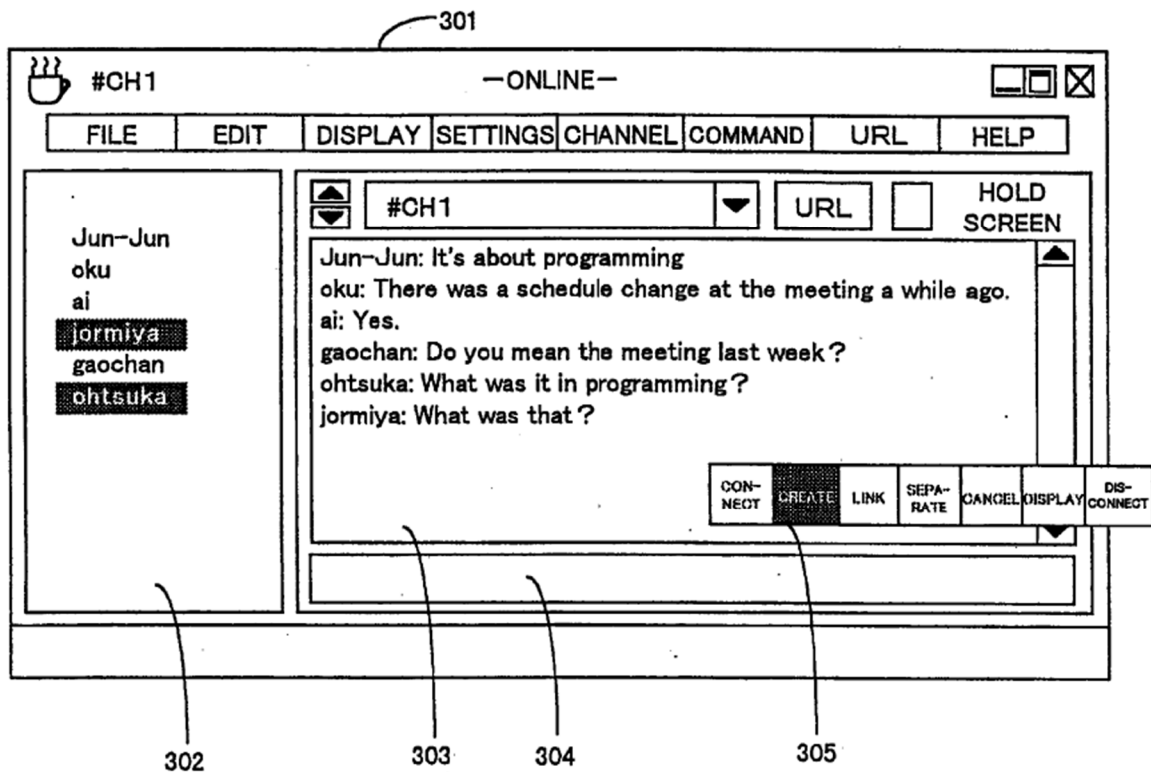
Kakuta improves on conventional IRC by providing a pop-up “thread” window where a subset of users (“message groups”) can have their own conversation, much like a breakout meeting within a larger conference.

65. Kakuta therefore proposes extending an IRC system by introducing a floating toolbar and thread window onto known IRC systems, but Kakuta does not propose modifying the behavior of the main chat window, which operates like a traditional IRC window. Rather Kakuta explains that the main chat window is a traditional IRC chat window. Kakuta states “FIG. 4 depicts a screen just after booting a chat client relating to the present embodiment. A main screen 301, a user window 302, and chat window 303, and message input field 304 are displayed in the FIG. 4 screen. These windows are likewise as with general IRC clients, therefore explanation is omitted.” Ex-1005, 8:1-3 (emphasis added).

66. Kakuta teaches three different ways to create a thread. (1) A user may select thread participants from the channel user list, (2) a user may select messages rather than users, or (3) a user may generate an empty thread and thereafter select users.

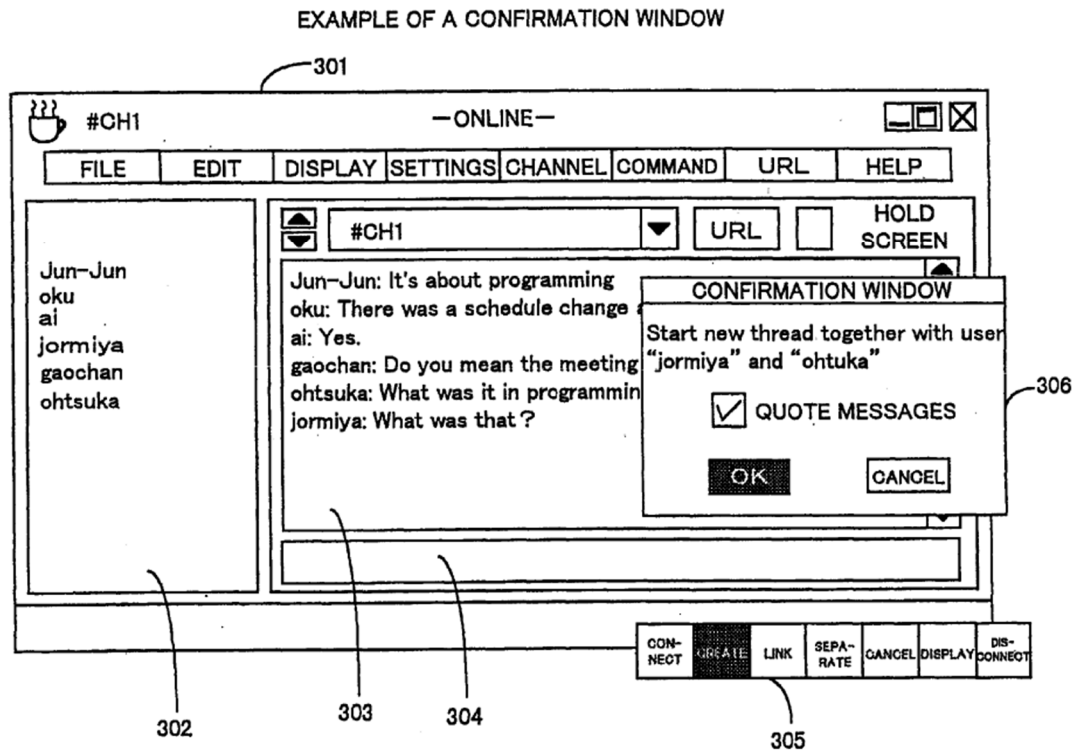
67. As depicted in Fig. 5, a user may start the process of thread creation by selecting participants from the user list and press the “Create” button:

SCREEN EXAMPLE: SELECTING USER CREATES THREAD

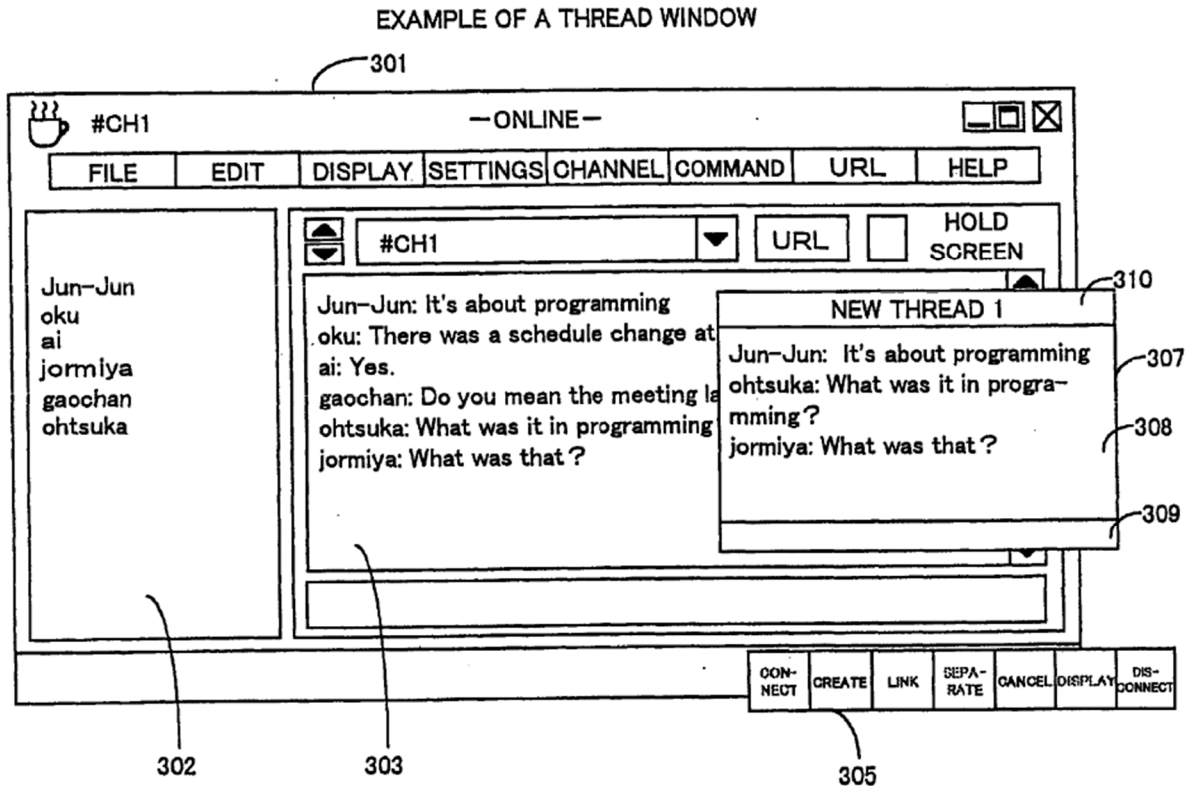


68. Figure 5 depicts a point in time where Jun-Jun has selected jormiya and ohtsuka and has then pressed the “Create” button from toolbar 305.

69. Figure 6 depicts the response of the system to the user pressing the “Create” button in Figure 5, which is to pop up a confirmation window 306. In this confirmation window, Kakuta prompts the user to confirm whether the user wishes to start a new thread and to choose whether past messages in the channel from the group members should be included (by clicking “Quote Messages”):



Ex-1005, Fig. 6, 8:12-29. Pressing “OK” in the “confirmation window” causes Kakuta to generate a new window on Jun-Jun’s screen for the chat group, as shown in Fig. 7:



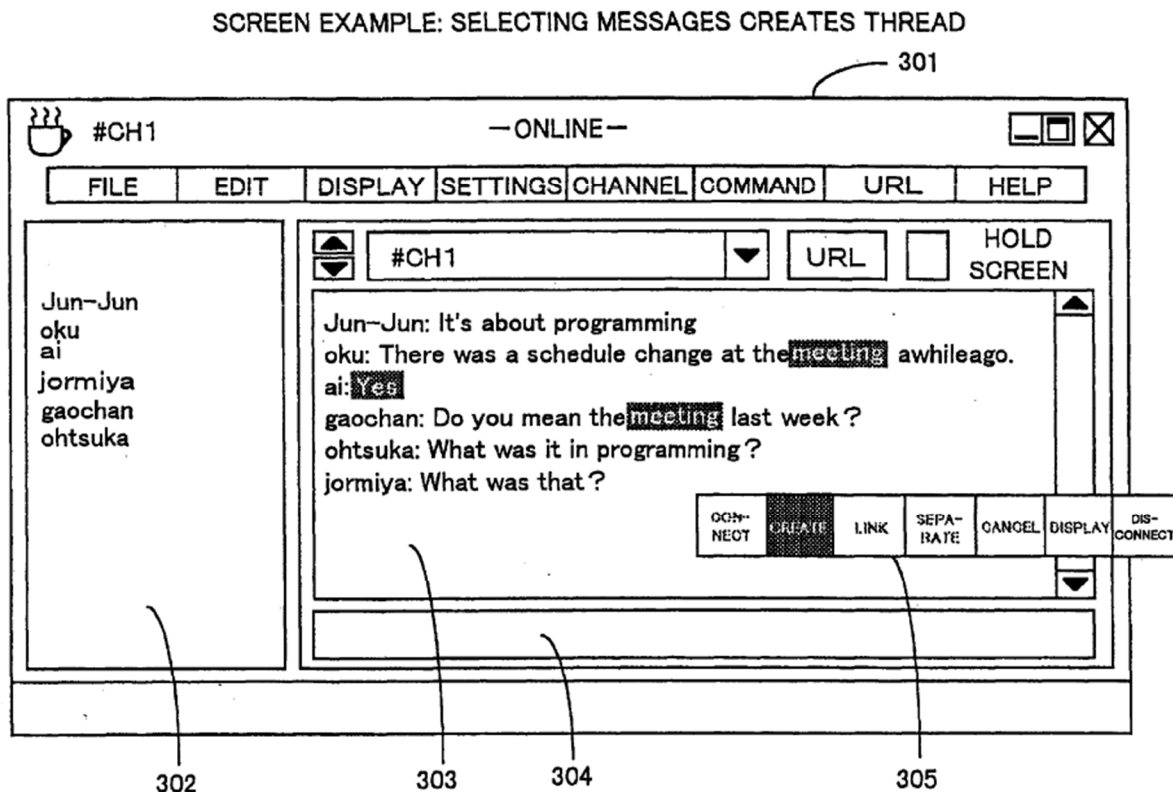
Ex-1005, Fig. 7, 8:38-42.

70. As seen in Figure 7, the **past** messages from the selected users are copied **from** the chat window 303 **to** the new thread window, which is labeled “New Thread 1.” This thread is by default marked “private,” which means only the selected users may see posts to the thread. Ex-1005, Fig. 3, 6:34-67. If marked “public,” then all users in the channel can see messages posted to the thread. Kakuta explains that in other embodiments, an option to modify the disclosure level is added to the confirmation screen. Ex-1005, 15:18-22, 3:56-63, 3:36-38, 8:50-51, 12:39-41.

71. Kakuta discloses a similar approach when messages are selected instead of users. As shown in Figure 8, a user would select messages and then

press the “Create” button, which would then cause a similar confirmation window to appear with a “Quote Messages” checkbox. Ex-1005, FIG. 8, FIG. 16, 8:43-51, 12:10-49.

Fig. 8



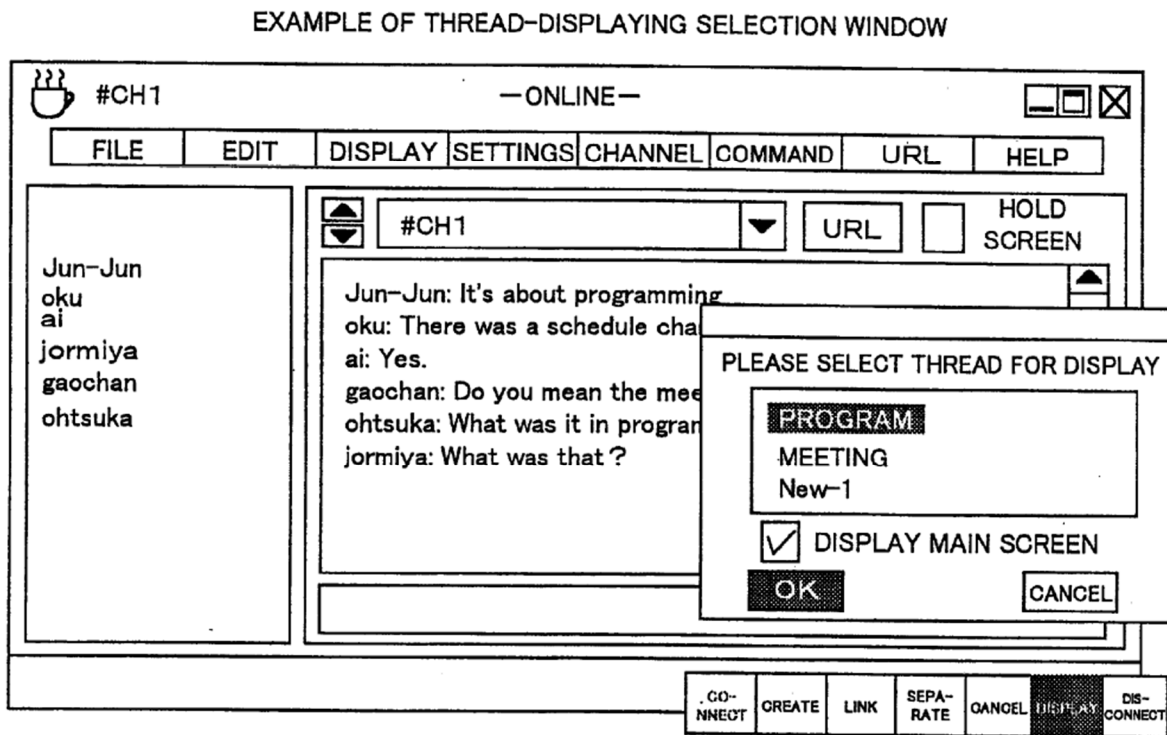
72. Kakuta thus requires at least three steps (depending on how one counts the steps) to open a thread window. In addition to the multiple steps, Kakuta’s Figures 5-8 embodiments differ from the ’731 patent in that other users are not informed when they are selected as members of a chat group.

73. Rather, Kakuta teaches that it provides a “Display” button that allows

users to see what groups they have access to (either because the threads are public or because the users are members of a private thread).

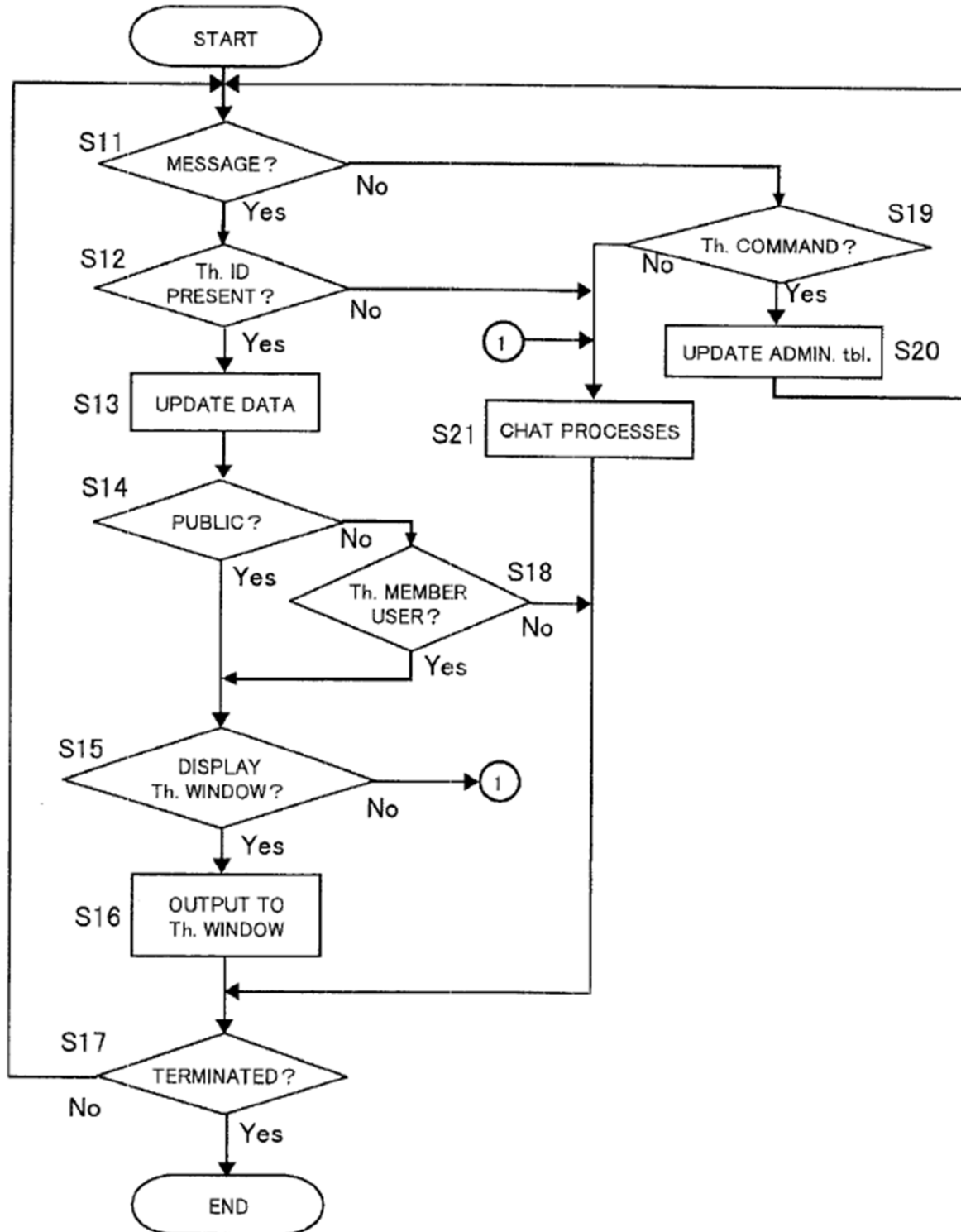
74. Kakuta explains that the “Display” button triggers the “Displaying Process Subroutine.” Ex-1005, 14:45-15:14, 9:23-31, Figs. 19, 12. A user pressing the “Display” button will see a listing of thread titles that they may access, as depicted in Figure 12. As shown in Figure 12, the user is presented with the three threads to choose from and the top thread (“PROGRAM”) has been selected for display (column 9, lines 30-31). As shown in the figures and as described in the text, Kakuta provides a “thread title” list (column 9, line 25), but does not indicate whether each respective thread is public or private. Nor does it indicate what messages are associated with particular threads. Just the titles are provided.

Fig. 12



75. Also, although Petitioner identifies messages in Kakuta's figures as being "additional messages" (thread messages), Kakuta's screen shot figures do not depict any messages that were entered into the thread message text input field 309 (shown in Kakuta, Figure 7). Rather, the process flow for additional messages is shown in Figure 14, the "Output Control Process." Kakuta explains that if a new thread message is received from the system, the client device will determine whether the thread window is open or not, as depicted in Figure 14.

OUTPUT CONTROL PROCESS



76. As shown in Figure 14, step S12 determines if the message received

over the network is a thread message. If so, the system determines whether the thread is public, and if not, whether the user is a member of the thread (steps S14 and S18). If either are true, then the user may see the message, and Kakuta asks whether the user has the thread window open (at S15). If the thread window is open, then the thread message is output to the thread window (S16). The process thus far is discussed in Kakuta at column 9, line 65 to column 10, line 67. If the thread window is not open, then the process continues at “chat processes” (S21). Kakuta describes step 21 as follows: “Herein, in Step S21, the administration module 23 sends the thread message to the control module 32. The thread message not being displayed in the thread window is therefore displayed in the main screen.” Ex-1005, 10:67-11:3.

77. Thus, the thread message is displayed in the thread window (if the thread window is open) or alternatively, the thread message is displayed in the main chat window (if the thread window is not open). The flowchart is clear that this is an “either or” situation. Either the received message is displayed in the thread window or the received message is displayed in the main chat window. The received message is not displayed in both windows on the same screen.

78. For example, if Jun-Jun sends jorymiya a thread message and jormiya’s thread window is not open, Jun-Jun’s message appears in the main chat window, without any indication that it forms part of a thread, or whether the thread

is marked public or private. Thus, jormiya may inadvertently reply to Jun-Jun's private message in the main chat window and thus send the reply to all the users of the channel (because jormiya does not know the message received was private).

79. At the end of Kakuta, Kakuta offers a solution of sorts to this problem. Kakuta discloses "Other Embodiment (e)":

(e) Wherein threads are created, it is possible to make all members display the same by sending to the selecting parties the selected message ID together with the thread creation command.

Ex-1005, 15:32-36. This passage suggests that other members of the thread will see the same screen as the user who created the new thread. However, how the thread creation command causes displays to be "the same" is not explained. Thus, presumably, a window would pop up on the screen unexpectedly, and the user would not be informed as to whether the thread was public or private or who the thread members are. Ex-1005, 15:33-36, 3:36-38, 8:50-51.

D. Matsumoto and Chang

80. Matsumoto and Chang bear little resemblance to either the '731 patent or Kakuta. Matsumoto seeks to display multiple channels at once and intermingles messages from multiple channels. Matsumoto describes a "decision means" that "decides whether or not to send [a message on a first network] to the second network based on the conditions table." Ex-2006, 3:10-18, 4:53-67. Whereas Kakuta provides for threads within a single channel, Matsumoto copies messages

from one channel to another based on conditions. Both Kakuta and Matsumoto have channel identifier elements where the channel title is provided for the main chat window. In both Kakuta and Matsumoto, there is a single channel title provided. The “channel window 36” of Matsumoto is used to list all the channels on the screen—the aggregate of the main channel window and the sub-channel window. Because Kakuta does not have a “sub-chat window” and does not display multiple channels at once, there is no reason a skilled artisan would modify Kakuta based on Matsumoto’s “channel window 36.”

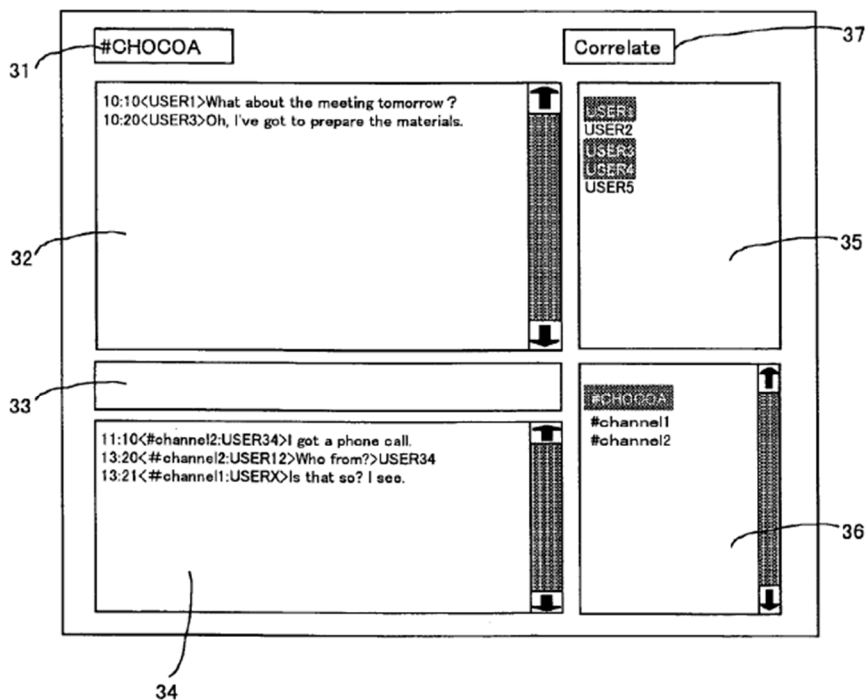
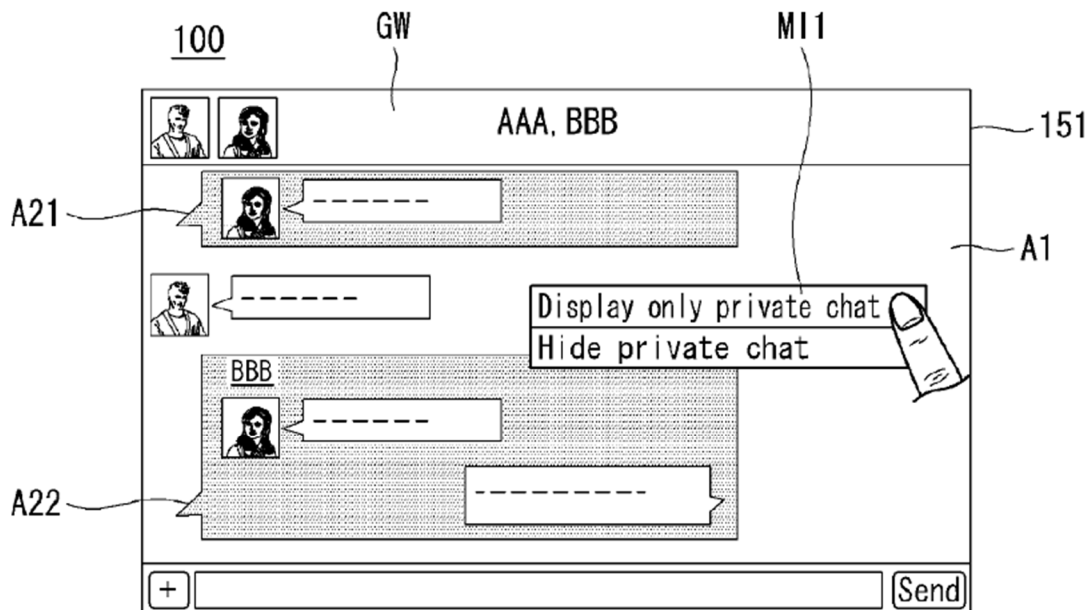


FIG. 3

Ex-2006 (Matsumoto), FIG. 3.

81. Chang provides a user interface for traditional chat tailored for mobile devices. Chang explains that “[t]erminals can be divided into mobile terminals and stationary terminals,” but in mobile terminals “screen size is limited,” and therefore, according to Chang, two or more chat windows may not be displayed simultaneously. Ex-1007, [0005], [0007]. Thus, in the case “where the user simultaneously performs private chat with another party while group chatting together with another parties, the user feels inconvenience when chatting while alternately moving chat windows.” Ex-1007, [0007]. To address this problem, Chang proposes techniques using various gestures to provide both a “group chat area” and “private chat areas” within the same chat window. Ex-1007, [0010].

FIG. 29



(a)

Ex-1007 (Chang), FIG. 29(a).

82. Neither Matsumoto nor Chang fills the multiple gaps between Kakuta and the '731 patent claims at issue in this proceeding.

VIII. GROUND 1: KAKUTA DOES NOT ANTICIPATE THE CLAIMS

A. Claim 1

1. Kakuta Does Not Disclose Limitation 1[c]—A “first pane configured to display a plurality of channel identifiers”

83. Limitation 1[c] requires:

a first pane configured to display a plurality of channel identifiers, each of said plurality of channel identifiers corresponding to a respective one of a plurality of group-based communication channels, each of such channels having an associated group of members who are users of said system

84. Figure 10B depicts an exemplary first pane, which is the pane on the left. The pane being displayed is the “Chat” pane, which displays multiple channels (e.g., private channels “Marketing,” “Tech,” and “UX Design” and public channel “Company”). In Figure 10B, the UX Design channel has been selected to be displayed as indicated by the “UX Design” name above the main chat pane (the “second pane” in the claims).



FIG. 10B

Ex-1001, Fig. 10B.

85. Kakuta has no “first pane” as claimed. In Kakuta, the left side of the screen is a “user window 302” that lists users for a particular channel. Multiple channels are not listed there.

86. Petitioner asserts that Figure 8 of Kakuta discloses a “first pane.” However, for this alleged “first pane,” Dr. Cockburn identifies a channel identifier element of Kakuta combined with a down arrow that he asserts is a “drop down menu.” Ex-1003, ¶59.

87. Petitioner is incorrect because a Skilled Artisan would not consider the channel identifier element or the alleged drop-down menu (or their

combination) to be a “pane.” Figure 10B of the ’731 patent, for example, clearly depicts the three “panes” of the invention:



First Pane

Second Pane

Third Pane

FIG. 10B

I note that Petitioner’s ’649 patent uses the same three-pane structure, and specially identifies each respective pane as a “pane” (“400 CHANNEL LIST PANE”, “410 CHANNEL MESSAGING PANE”, and “420 THREAD COMMUNICATION PANE”):

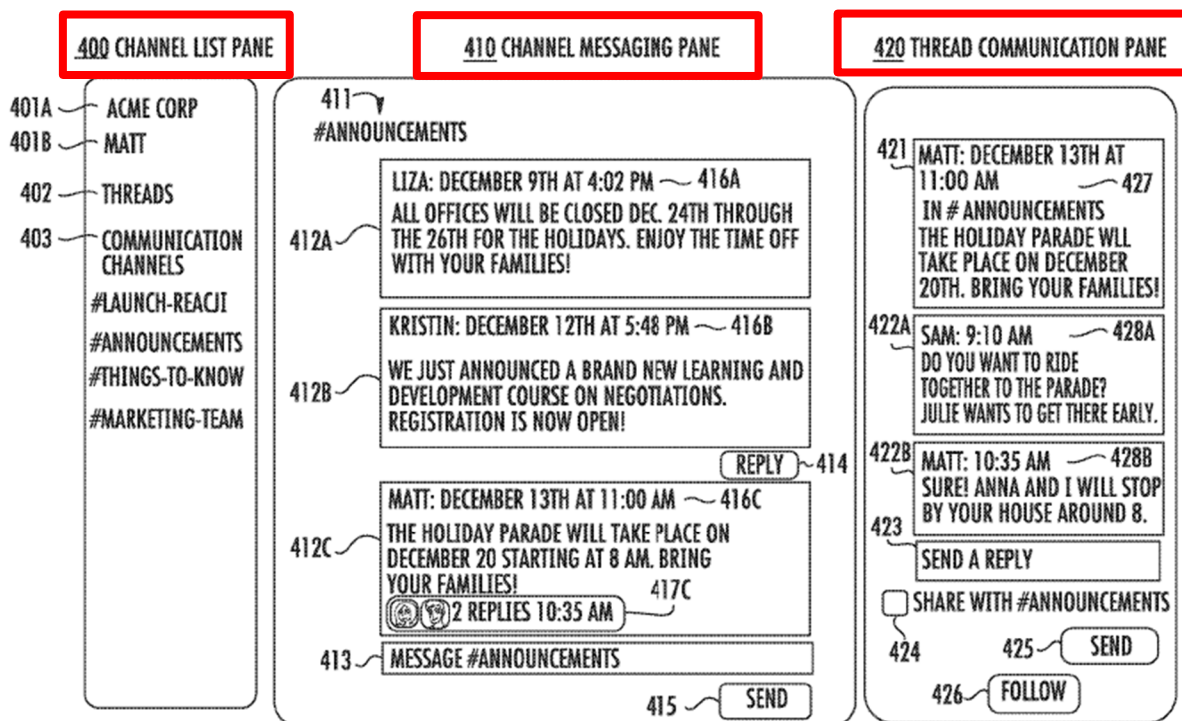


FIG. 4

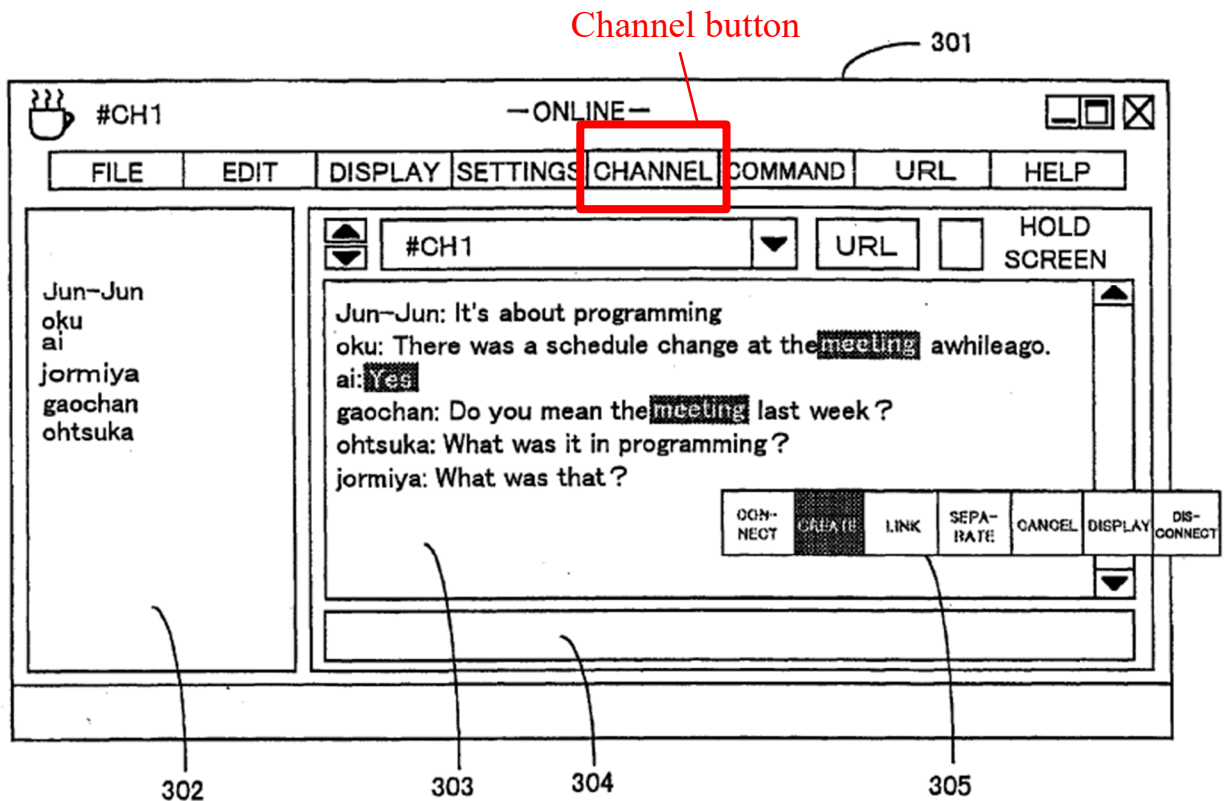
Ex-2010 (U.S. Patent No. 11,539,649).

88. The channel identifier in Kakuta is merely an area within a window, and not a pane. Kakuta does not disclose a drop-down menu, *but even assuming that it did*, a drop-down menu is also not a pane, and would not have been understood by a Skilled Artisan as a “pane.” For example, Petitioner cites to Exhibit 1012 as evidence of how a drop-down menu would operate. That Exhibit, however does not refer to its drop-down menu or combo boxes as “panes” and does not offer a description consistent with how a Skilled Artisan would understand the term.

89. On the contrary, the exhibit on which Petitioner relies calls these

things “widgets.” A Skilled Artisan would have understood that widgets in the context of Kakuta are user interface components that might be included within a pane, but are not themselves panes. For example, Petitioner (or its sister company) understood, when prosecuting its own patent in this area, that “Kakuta is silent as to causing ‘rendering of first group-defined titles associated with ... communication channels’ and presenting them via ‘a first sidebar pane of a user interface.’” Ex-2005 (Slack’s response during prosecution of patent application no. 16/399,349, where Slack distinguished Kakuta to obtain its claims), p. 14. I agree—Kakuta’s channel identifier element (1) identifies a single channel and (2) is not a “pane.”

90. In addition, even setting aside the “pane” issue, the channel identifier identifies a single channel, and not a plurality of channels. Regarding the arrows near the channel identifier, a logical interpretation of a Skilled Artisan would have been that the up and down arrows to the left of the channel identifier might allow the user to go up or down one channel at a time (and displaying only one channel at a time). But, a Skilled Artisan would not conclude that the down arrow generates a drop-down menu that provides a channel listing. This is because Kakuta discloses a “Channel” button that would provide the channel listing, as would be conventional in an IRC system (and Kakuta builds on conventional IRC systems).



91. Along the top of the main screen is a row of buttons that a user may expect to find in a traditional IRC system. One of those buttons is a “Channel” button. Pressing a “Channel” button on a traditional IRC system would cause the system to switch to a channel selection screen. For example, in Exhibit 1012 filed by Petitioner, a traditional IRC screen is shown:

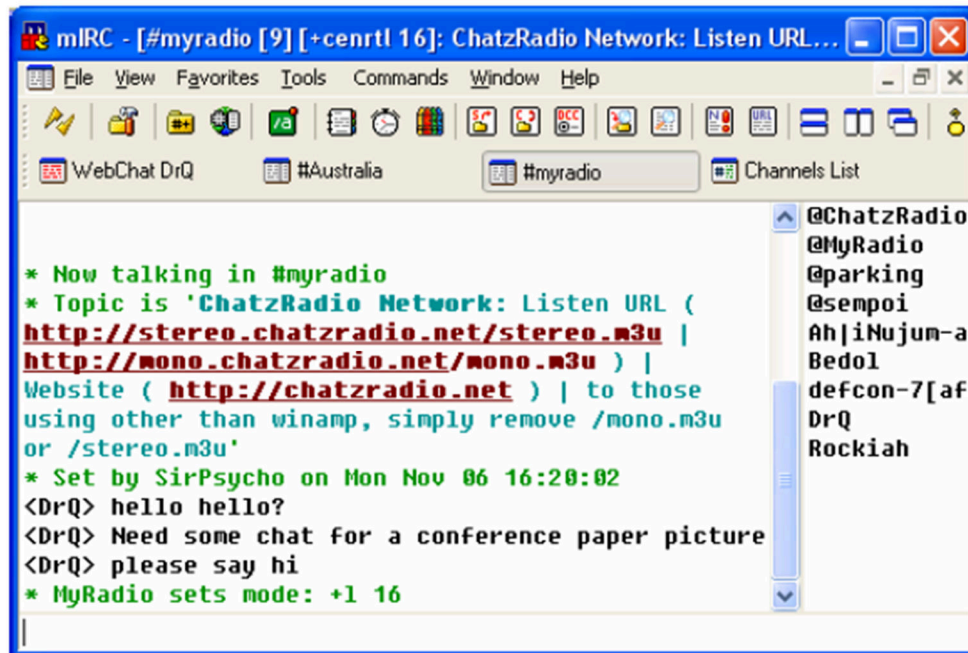
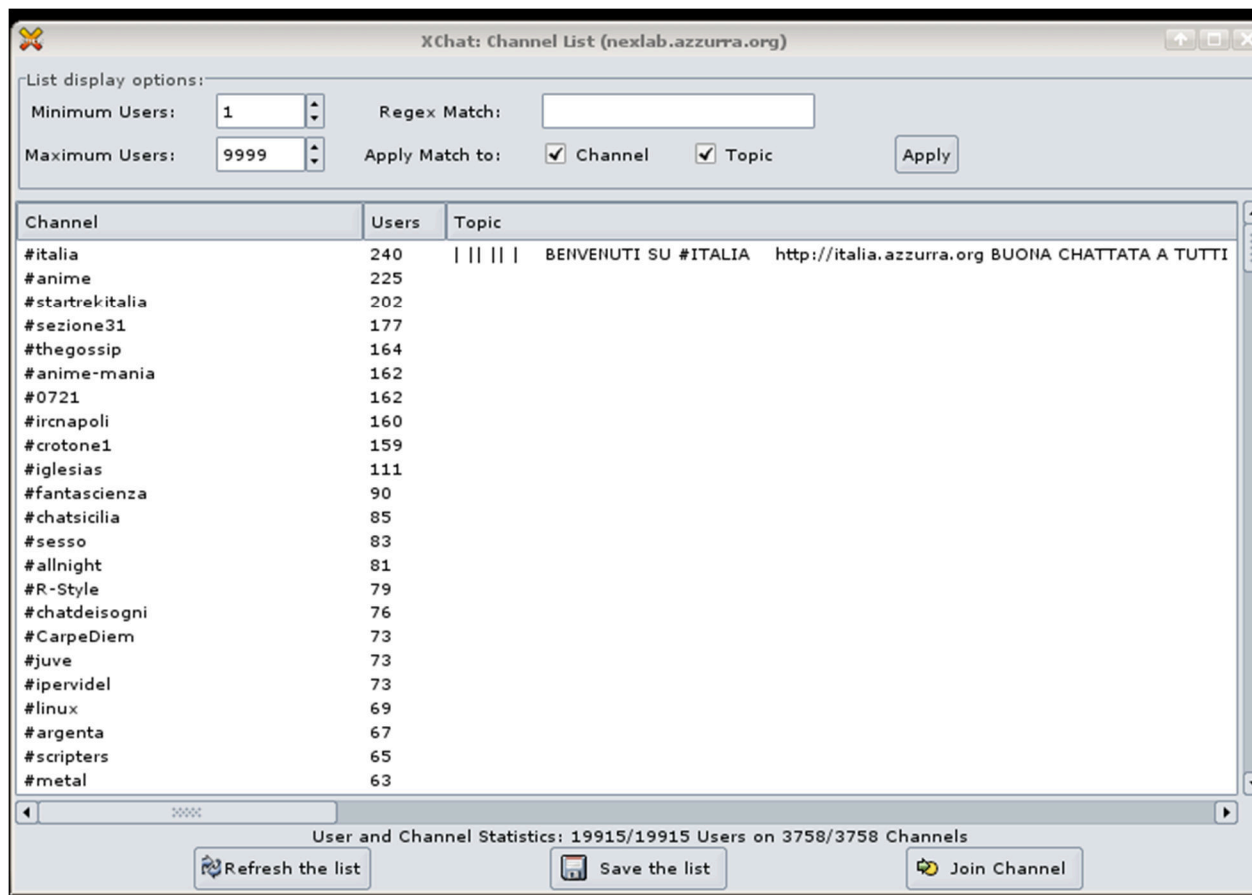


Figure 3. Typical IRC User Interface (MIRC)

Ex-1012, Fig. 3. Similar to Kakuta, this screen has a row of buttons above the chat window. The rightmost button is “Channels List.” A Skilled Artisan would understand that this “Channels List” button would cause the system to switch from the screen shown above to a channel selection screen similar to what is shown below (although these screenshots come from two different IRC systems). Once a new channel selection is made, the screen would revert back to the main chat screen as shown above.



Ex-2008

(https://commons.wikimedia.org/wiki/Category:IRC_screenshots#/media/File:Azzurra_lista_canali.png).

92. Even if Kakuta had not disclosed a “Channel” button, a Skilled Artisan would not have assumed or concluded that Kakuta intended to use a drop-down menu for channel selection. There is no evidence that IRC systems around the time of the invention used drop-down menus from the main chat screen to handle channel selection, and I am aware of none. Kakuta’s silence on this issue would have informed a Skilled Artisan that Kakuta used a traditional “Channel”

button depicted in Kakuta's figures. The set of channels in a system such as Kakuta is both unbounded and numerous. For example, the channel listing shown in the above screenshot has 3,758 channels at the time of the screenshot, and that number would likely grow over time. A Skilled Artisan would not use a drop-down menu in Kakuta for a channel listing in the thousands or even hundreds. On the contrary, the full screen real estate would be used, as was typical, to provide channel information. This is particularly true where main chat screen concerns a single selected channel and therefore once a channel is selected, there would be no reason to take up valuable real estate with an irrelevant channel listing.

93. Petitioner and Dr. Cockburn read too much into the down arrow to the right of the channel identifier. Ex-1003, ¶59. Even assuming that the down arrow triggered a drop-down menu, a Skilled Artisan would not conclude the menu contains a channel listing (alleged by Dr. Cockburn to be "necessarily" present) because there is a "Channel" button elsewhere on the Kakuta screen. Given Kakuta's focus on threads, a more logical conclusion is that such a menu would provide a shortcut for a thread listing or some other thread-related functionality. Such a list would have a direct relationship to the selected channel and would provide functionality in addition to traditional IRC functions.

2. Kakuta Does Not Disclose Limitation 1[h]— “display a first user selectable control associated with said first messaging communication”

94. Limitation 1[h] requires:

display a first user-selectable control associated with said first messaging communication

95. This limitation provides a specific way to trigger the display of a third pane. For example, in Fig. 10A of the '731 patent, a user hovers over Ali T.'s message at 8:42 am and is presented with a menu of user-selectable controls associated with that specific message. In the example, the arrow control is selected (a “first user selectable control associated with said first messaging communication”) which causes the display of the third pane with Ali T.'s message from 8:42 AM, as shown in Fig. 10B.

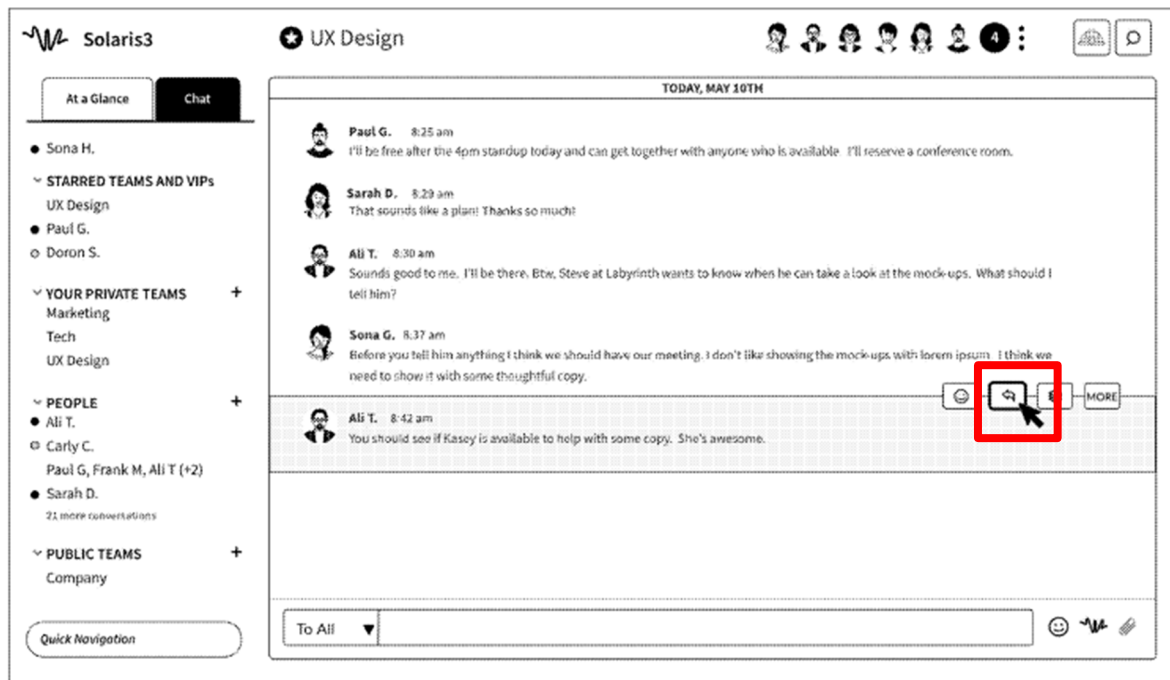


FIG. 10A

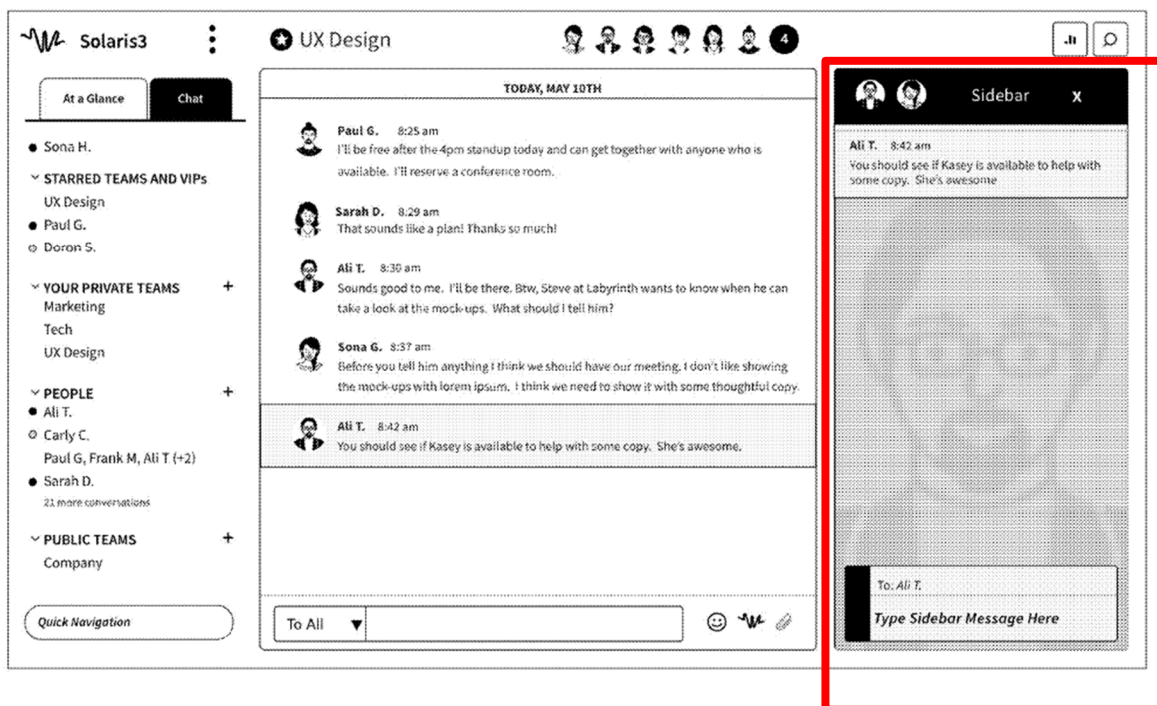


FIG. 10B

Ex-1001, FIGS. 10A, 10B, 24:16-44. From the positioning of the control relative to the message, a reader immediately understands that the control provides functions related to that specific message and not other messages.

96. Kakuta lacks any such control and Petitioner identifies none. Kakuta operates in a fundamentally different manner by using a generic floating toolbar to provide system-level functionality that is not tied to any specific messaging communication as required by the claims. For example, the “Create” button is used to create threads and the “Display” button is used to choose a thread for display. These buttons are not associated with a particular messaging communication and nothing in the positioning of the toolbar provides an association between the toolbar (and its buttons) and a particular messaging communication. In fact, the toolbar is floating and is shown in various positions on the screen throughout the various figures. In contrast to the ’731 patent, there is no correlation between the position of the toolbar (and its buttons) and messages on which the toolbar or its buttons operate. For example, in each figure of Kakuta depicting a screenshot, the position of the buttons on the floating toolbar bears no relationship to any particular message and the position does not indicate an association between the button and any particular message, and the toolbar is present even when no messages are displayed. I provide Kakuta’s figures depicting the screens with toolbars below:

Fig. 4

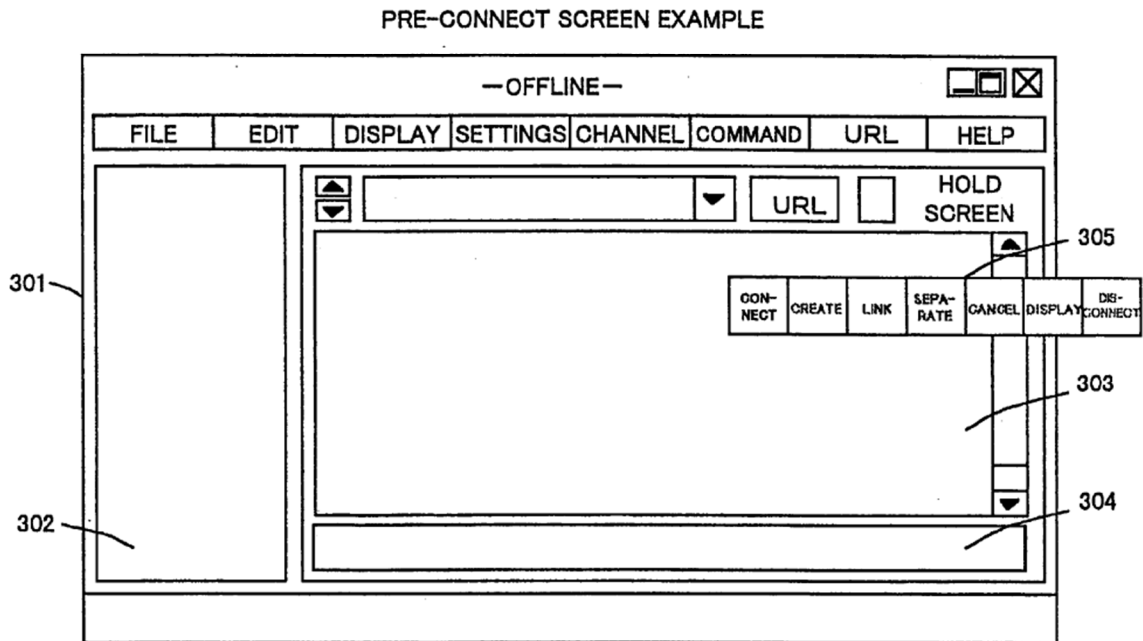


Fig. 5

SCREEN EXAMPLE: SELECTING USER CREATES THREAD

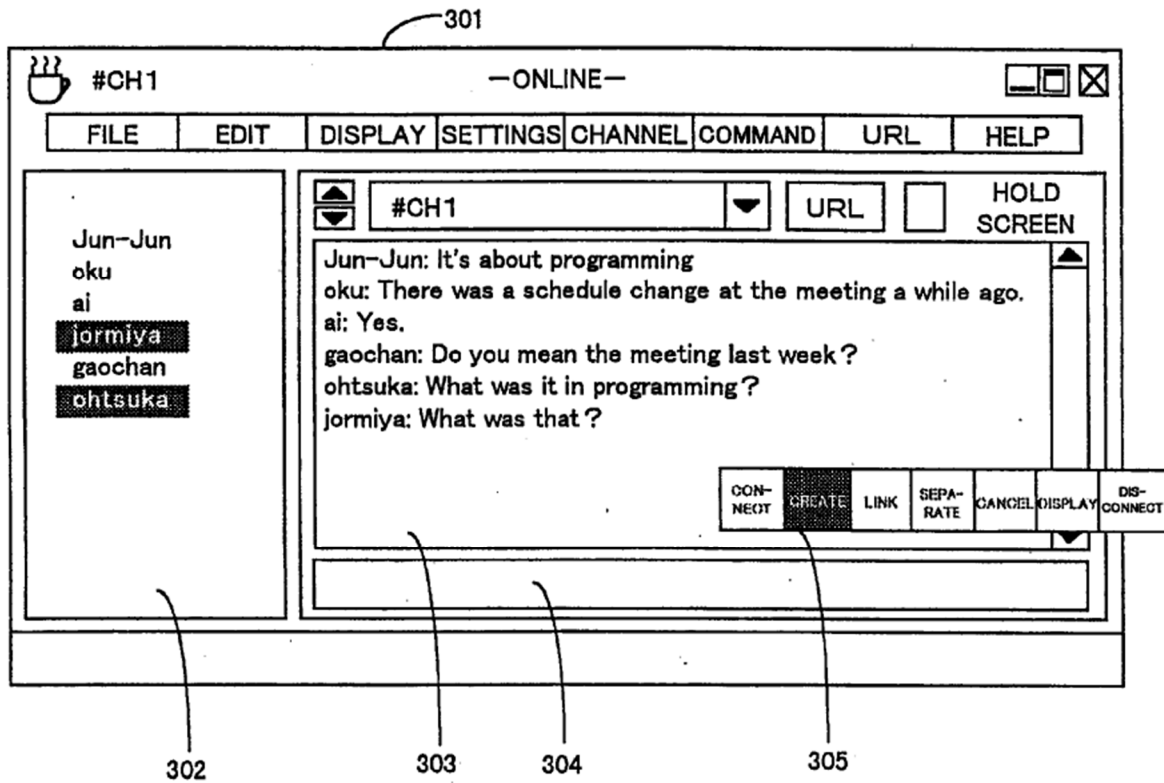


Fig. 6

EXAMPLE OF A CONFIRMATION WINDOW

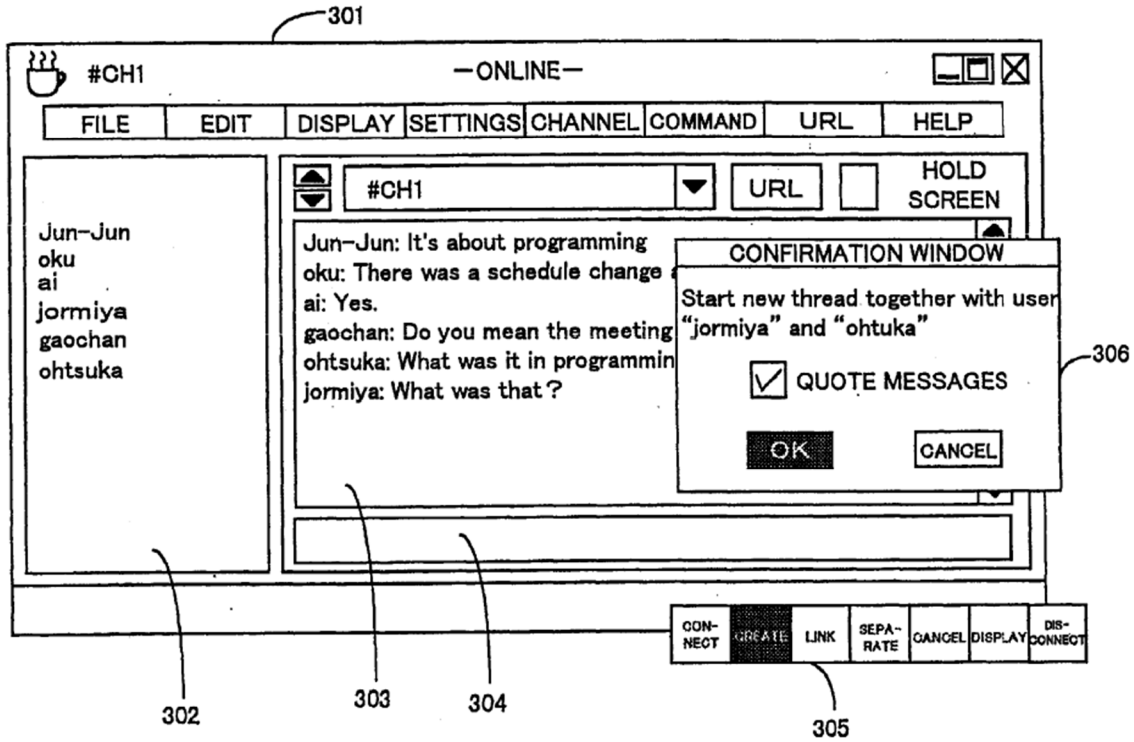


Fig. 7

EXAMPLE OF A THREAD WINDOW

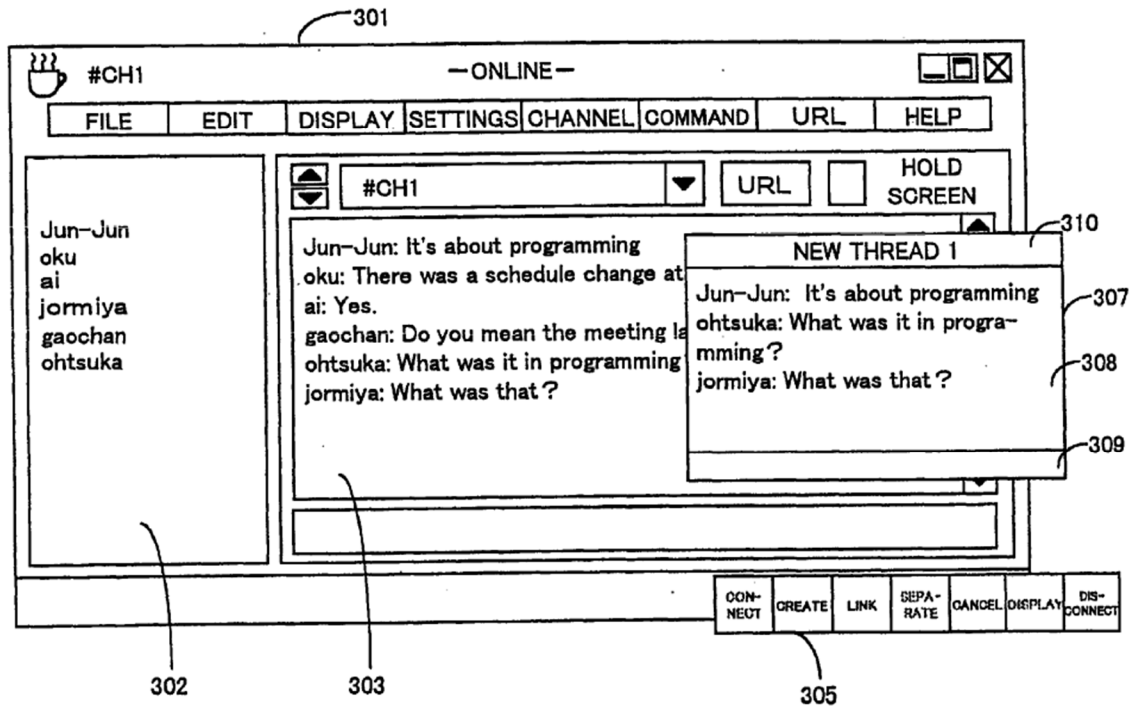


Fig. 8

SCREEN EXAMPLE: SELECTING MESSAGES CREATES THREAD

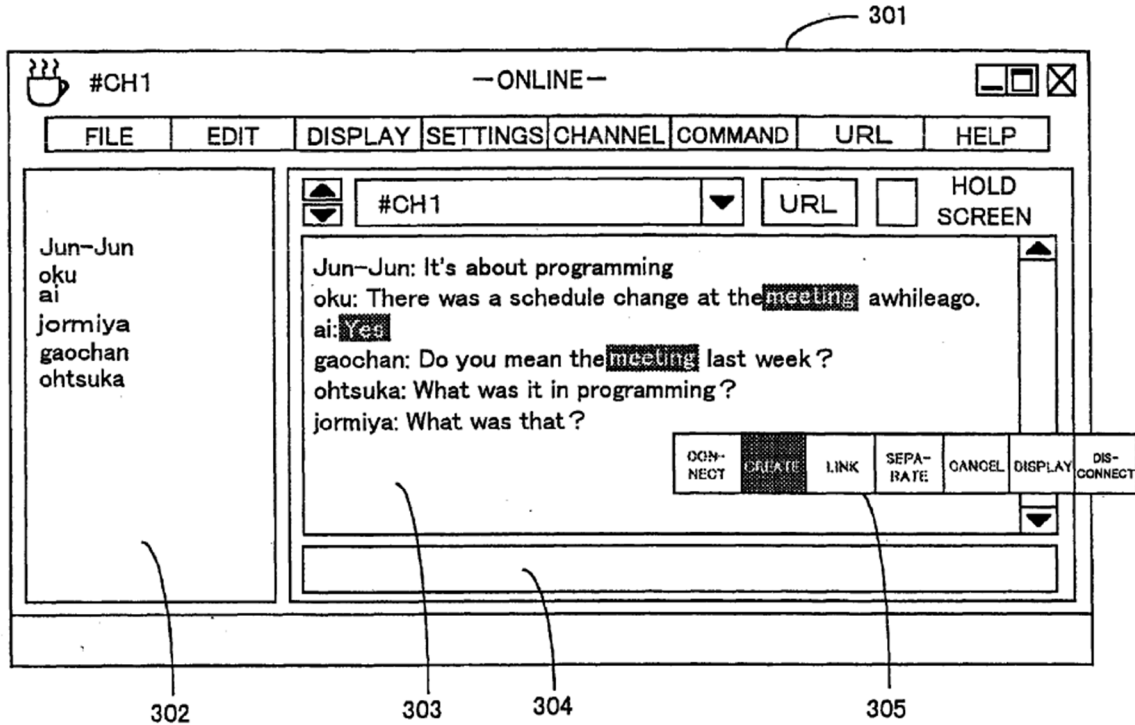


Fig. 9

EXAMPLE OF CREATED THREAD

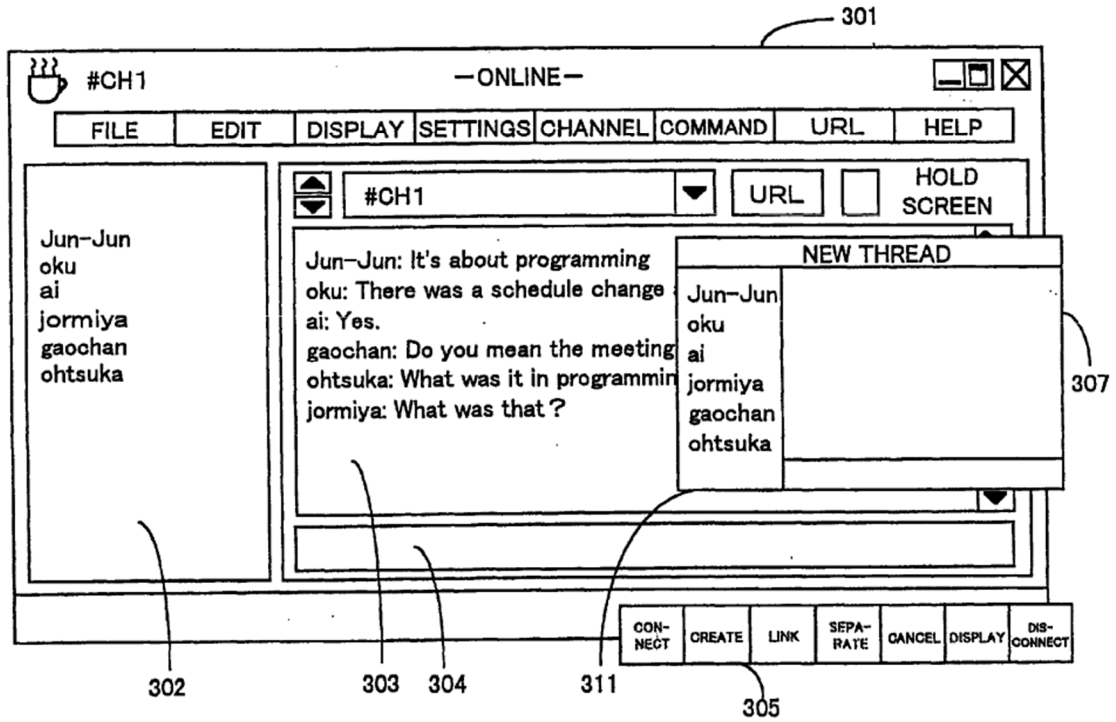


Fig. 10

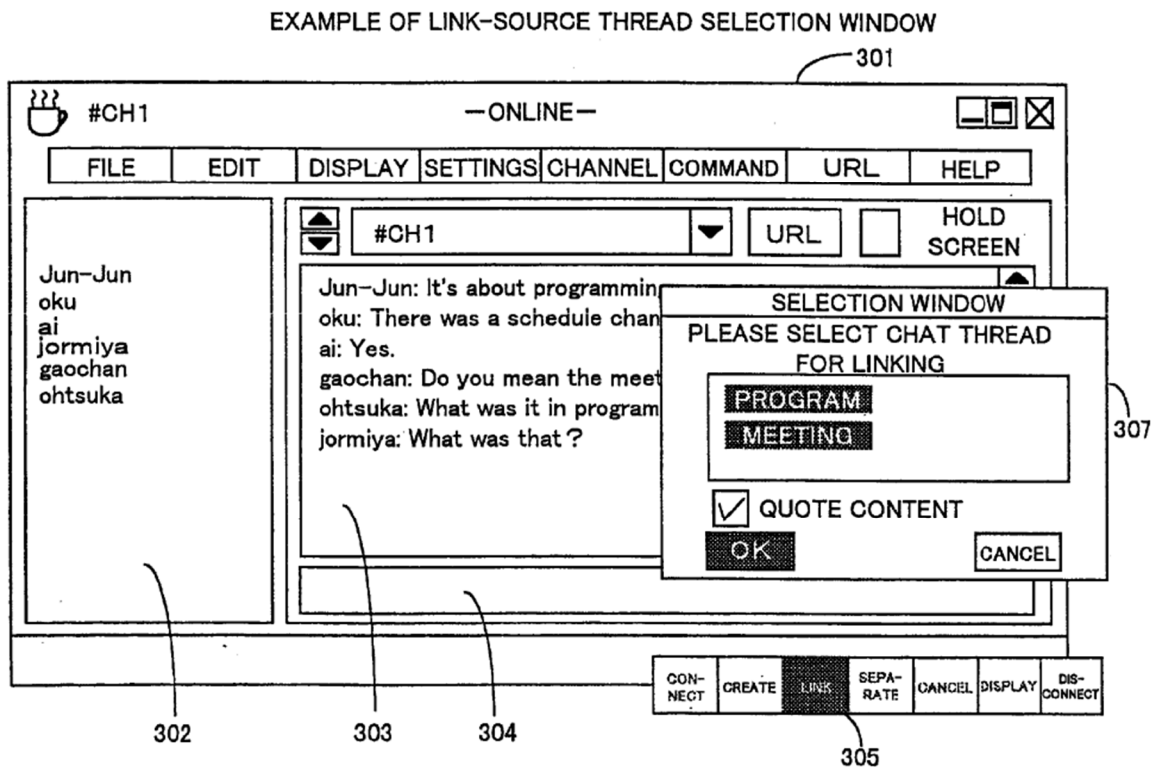
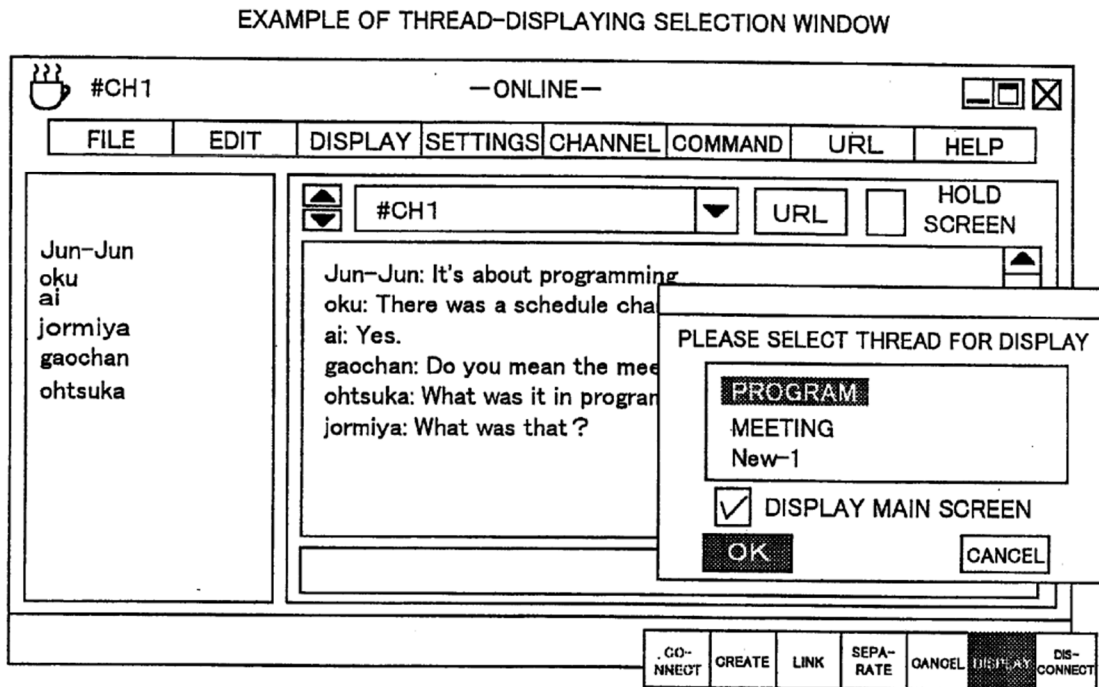
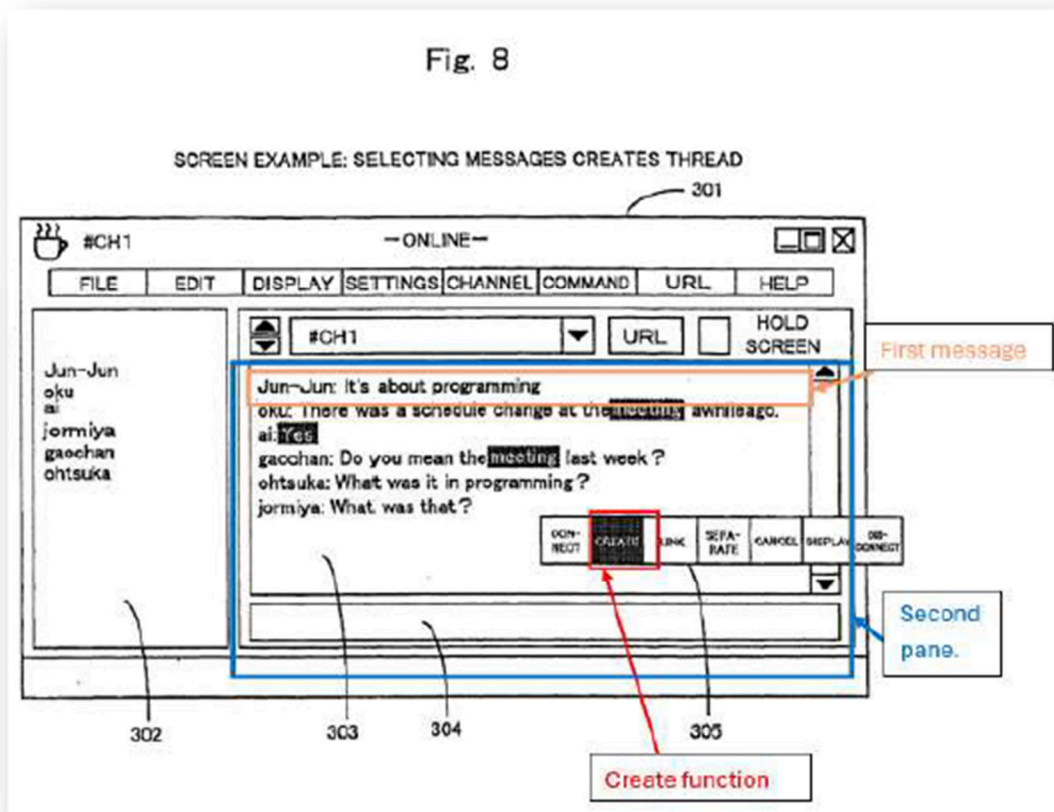


Fig. 12



97. In Kakuta, a user creates a thread in one of three ways: selecting users and then pressing the “Create” button, selecting messages and then pressing the “Create” button, or by pressing the “Create” button and later selecting users. In each option, the position of toolbar has no relationship or association with any particular message. Petitioner and Dr. Cockburn assert that the message-selection embodiment of Figure 8 “illustrates the use of a ‘first user-selectable control’ (Create) among the toolbar (305) that is associated with a first communication message [sic] (i.e., highlighted selected messages).” However, no further explanation is provided. Petitioner does not identify which of the three messages

is the alleged “first messaging communication” or explain how the alleged association is displayed. Petitioner instead identifies a “First message,” but that message has not been selected for the thread and thus cannot be the “first messaging communication.”



Ex-1003, ¶80.

98. In addition, the highlighting in the chat window 303 of Figure 8 was not generated by the system to inform the user which messages the toolbar is associated with. On the contrary, it was caused by the user clicking on words to select messages for the thread, and the order in which the user clicked the words is

not known. Notably, even when selecting messages, the toolbar does not relocate to a position near the selected message to convey some sort of association.

Nowhere in Kakuta is there a suggestion that the toolbar's position is related in any way to the selected messages when message selection is used to create a thread.

3. Kakuta Does Not Disclose Limitation 1[i]—Kakuta Does Not Display a Third Pane “In Response To” a User Selection of “Said First User-Selectable Control”

Limitation 1[i] requires:

in response to a user selection of said first user-selectable control, display a third pane, said third pane displaying said first messaging communication and having an associated sidebar text-entry field that is different from said channel text-entry field and operable to receive user input of an additional messaging communication

99. Because Kakuta does not disclose the “first user-selectable control,” of limitation 1[h], Kakuta cannot take any action “in response to” a selection of a non-existent control. Therefore, because limitation 1[h] is missing, limitation 1[i] is necessarily missing as well.

100. Also, even assuming Kakuta's “Create” button could be the “first user-selectable control” of limitation 1[h], it would still fail to meet limitation 1[i]. The “first user-selectable control” must cause the display of the “third pane” (if not already being displayed). That is, the pane must be displayed “in response to” selection of the control. However, in Kakuta, the third pane is not displayed “in

response to” the “Create” button. Rather, it is the “confirmation window” that is opened in response to the pressing of the “Create” button. In addition, a user must press the “OK” button on the “confirmation window” before the thread window (which Petitioner identifies as the third pane) will open. Dr. Cockburn’s statement that the thread window opens “in response to” the pressing of the “Create” button (Ex-1003, ¶83) ignores the “confirmation window” depicted in Figure 6, which is the window displayed in response to a click of the “Create” button.

101. In addition, claim 1 of the ’731 patent claims recite a one-step process in which a user selects a message displayed in the second pane (by selecting a control associated with the message), and in response, the system opens a third pane and displays the selected message in the third pane.

102. But, Kakuta fails to disclose this one-step process. In order to create a separate thread in Kakuta based upon message selection, a user must engage in at least three steps. If the user wishes to have specific messages displayed in the thread, the user must first select each message and then select the “Create” button. And even then, the user selection of the “Create” button does **not** cause the display of the thread window (the alleged “third pane”). Instead, it causes the display of a “confirmation window.” The “confirmation window” is a dialog box which requires further user action, a third step of pressing “OK,” before a thread window is displayed. If the user does not press “OK,” (e.g., presses “cancel” instead) the

thread window is never displayed. The “confirmation window” provides for further user selections, such as allowing a user to change the default setting of the “Quote Messages” check box or change the default privacy setting for the thread. Ex-1005, 15:18-22. Thus, Kakuta has a different structure than the ’731 patent, providing for intermediate steps and decision making that distinguishes Kakuta from limitation 1[i] of the ’731 patent.

**4. Kakuta Does Not Disclose Limitations 1[n] and 1[o]—
Kakuta Does Not Have an Anchor Message and Has No
“Second User-Selectable Control” and No “Indication” As
Claimed**

103. Limitations 1[n] and 1[o] require:

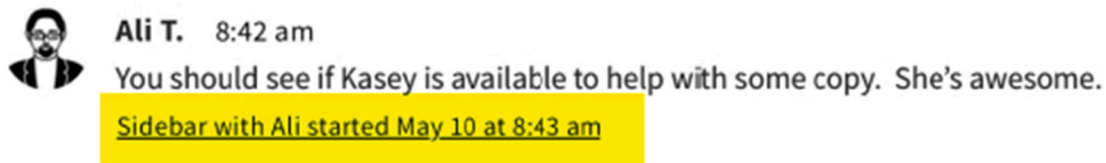
display, in said third pane, a first subsidiary communication thread having an anchor message, said thread comprising said first messaging communication together with said additional messaging communication, wherein said first messaging communication is the anchor message of said first subsidiary communication thread;

display a second user-selectable control associated with said first messaging communication in said second pane, said second user-selectable control being associated with an indication that said first messaging communication is an anchor message of a subsidiary communication thread

104. These limitations concern the anchor message of the ’731 patent and its associated indications and controls. In the claims, the “anchor message” is the first messaging communication that had been selected by the user to start the

thread. It will be displayed in the third pane and also in the second pane, where it is associated with a second user-selectable control that is, in turn, associated with an indication that informs the user that the first messaging communication is an anchor message.

105. The '731 patent illustrates the display an exemplary anchor message in the second pane with the associated control and indication:



Ex-1001, FIG 10E (excerpt).

106. Kakuta lacks any such anchor message.

107. Dr. Cockburn states that “Kakuta teaches that the first messaging communication of both pane two and pane three is the anchor message of pane three. This is illustrated in Figure 7” and annotates Figure 7 to identify the top-most message. Ex-1003, ¶100. But Figure 7 is an example of a thread window created based on the selection of users, not messages. The messages in the thread window are simply a portion of the historical messages sent by the members of the thread before the thread was formed. Nothing in Kakuta suggests that there is anything special about the message that happens to be displayed at the top of the thread window. As messages are added, messages will scroll up and off the

window. If there are more historical messages than space in the window, then the top-most message is an arbitrary message in the middle of the message stream. Thus, the message in the top-most position in the window does not indicate a message in an anchor message. In addition, the top message in Figure 7 happens to be a message prior to the creation of a thread. However, that is not apparent from the Kakuta user interface. For example, if “Quote Messages” were not selected, then the top message would be the first thread message input in the message input field 309. Such a message is not an anchor message as it cannot be the “first messaging communication” from earlier in the claims.

108. Dr. Cockburn suggests that the top-most message “can serve as the anchor message.” Ex-1003, ¶101. However, I understand that for anticipation, a claimed feature such as an “anchor message” must actually be present in the prior art. It is thus not sufficient that a message “can serve” as an anchor message. The question is whether it is an anchor message. Even for obviousness, the test is what a Skilled Artisan “would” do and not what a Skilled Artisan “could” do with a reference.

109. In Kakuta, no message serves as an anchor message, as the concept is foreign to Kakuta. Messages are only copied from the chat window to the thread window if the “Quote Messages” checkbox is checked, and when copied, they serve no function in the system (they are provided to give the user context for the

thread). Nowhere does Kakuta suggest that a user selects a message to start a thread and then is provided with a control in the second pane that is both associated with the message, and with an indication that the selected message is an anchor message. In fact, as Kakuta states, even when messages are selected, “it is preferable that selecting messages means selecting users.” Ex-1005, 8:50-51. Thus, message selection is just another vehicle to identify the members of a thread, and Kakuta is not concerned with identifying anchor messages. In fact, when selecting users, a thread is created with the thread creator plus the selected users. Thus, Jun-Jun selects two users and a thread with three members is created. Ex-1005, FIGs 5 & 7. However, when messages are selected, the thread creator is not presumed to be a member. For example, Oku (the thread creator) selects messages authored by Oku and others. Ex-1005, FIG. 8 & 11A. Because threads would generally have at least two members, a Skilled Artisan would understand Kakuta as contemplating that at least two messages would be selected to create a thread to identify at least two thread members. This is in contrast to the ’731 Patent where a single message is selected to be the anchor message for a new thread.

110. Dr. Cockburn also relies on Figure 2 of Kakuta as disclosing an anchor message (the message with Message ID of “1”). Ex-1003, ¶102. But Figure 2 is a “conceptual illustration of a message list,” not something shown to the user. Ex-1005, 4:57. The message with Message ID “1” was created prior to

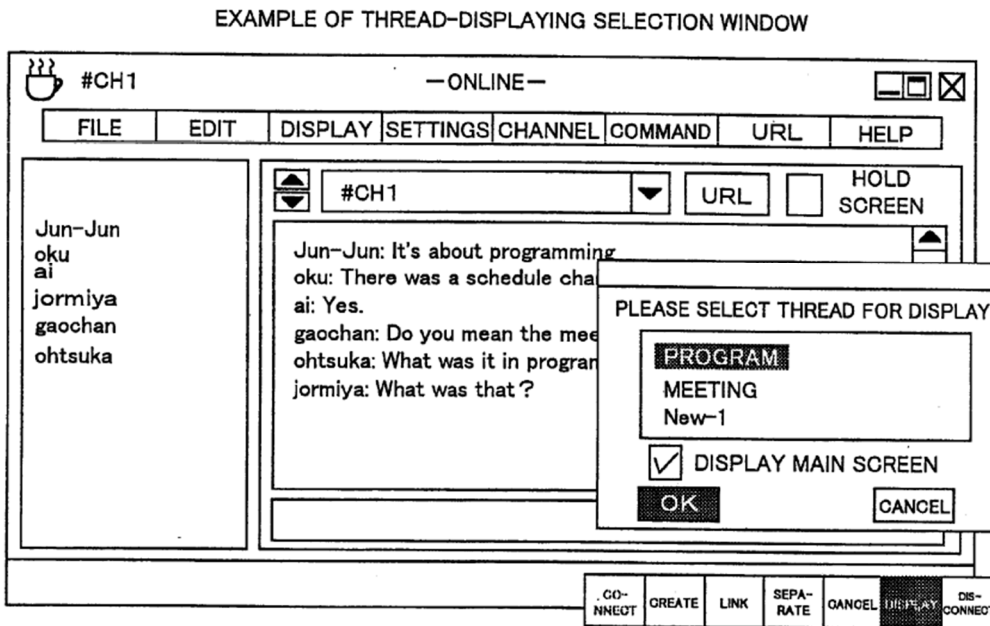
any thread creation and was assigned to thread 1 based on the selection of users and the checking of the “Quote Messages” checkbox. Ex-1005, 7:19-22. Thus, Kakuta is clear that the message with Message ID “1” is a historical message like any other historical message that may be included at the time of thread creation. Kakuta discloses the sequential assignment of Message IDs and thus the ID of “1” signifies nothing more than that the message is the first message entered into the system. A Skilled Artisan would have understood that the ID of “1” says nothing about whether the message is in a thread or not, and the ID number was assigned well before any thread was created. Finally, the particular Message ID number is not relevant to a user and does not appear on the user interface and thus a Skilled Artisan would have understood that the Message IDs would not be presented to the users.

111. Kakuta also lacks the claimed second user-selectable control and indication. Dr. Cockburn identifies the “Display” button on the floating “toolbar 305” as the alleged second user-selectable control. Ex-1003, ¶104. However, the claim requires that the control be “in said second pane.” Dr. Cockburn maps Kakuta’s main chat window 303 onto the “second pane,” but the “Display” button is not in that window. Rather, it is displayed in a floating toolbar 305 that is distinct from the chat window 303 and that is displayed outside of it. The “Display” button is not “display[ed] ... associated with said first messaging

communication in said second pane.” It’s placement on the floating toolbar and the placement of the toolbar within the overall screen has no association with the “first messaging communication in second pane.” This is in contrast to the ’731 patent where the control is placed near the first messaging communication in a manner that conveys to the user an association between the two.

112. In addition, the “Display” button is not “associated with an indication that said first messaging communication is an anchor message.” To the contrary, as shown in Fig. 12 reproduced below, the “Display” button is a general system function that does not operate in relation to displayed messages. Selecting it causes the display of a selection window that contains a list of existing thread titles (not displayed messages) and includes no information at all about specific messages in the second pane, let alone their status as anchor messages. Ex-1005, 9:23-26; Fig. 12.

Fig. 12



113. Alternatively, Dr. Cockburn asserts: “The first communication message is indicated as an anchor message because it is presented as the first (i.e., top-most) message within the thread.” Ex-1003, ¶106. That is incorrect because in Kakuta, the “top-most” spot does not convey information about the status of the message residing there. The message may be a past message (when “Quote Messages” is checked) or a new additional message entered into the thread message input field 309 (when “Quote Messages” is not checked). In Kakuta, the “top-most” message is not selected by the user, but is the result of (a) the system operation, (b) various options selected by the user, and (c) the size of the window in which the message is being displayed. In two of the three ways that threads can be formed, there are no messages selected, and even in the message selection

embodiment, the messages are being selected to provide a topical context, and not to identify the first message in a thread. Moreover, if the window is too small to hold all messages, then the “top-most” message is not even the earliest message in the thread.

114. Finally, Dr. Cockburn relies on the thread titles shown in Figure 12 of Kakuta. Ex-1003, ¶¶104-106. But, these titles are arbitrary and are chosen by the user to provide other users a contextual clue as to the topic of the thread. The title signifies nothing about any particular message in the thread and thus cannot provide an “indication” that any particular message is an anchor message.

Petitioner asserts that the thread title “Programming” “corresponds” to the first message about programming. However, both the first and second message use the word “programming.” A Skilled Artisan reading Kakuta would not have viewed the title of the thread “Programming” to correspond to the first message in the thread. On the contrary, a Skilled Artisan would have recognized that the thread title corresponds to the entire thread as a whole, and not to any individual message.

5. Kakuta Does Not Disclose Limitation 1[p]—The “Display” Button Does Not Cause a Portion of the Subsidiary Thread to be Displayed in the Third Pane

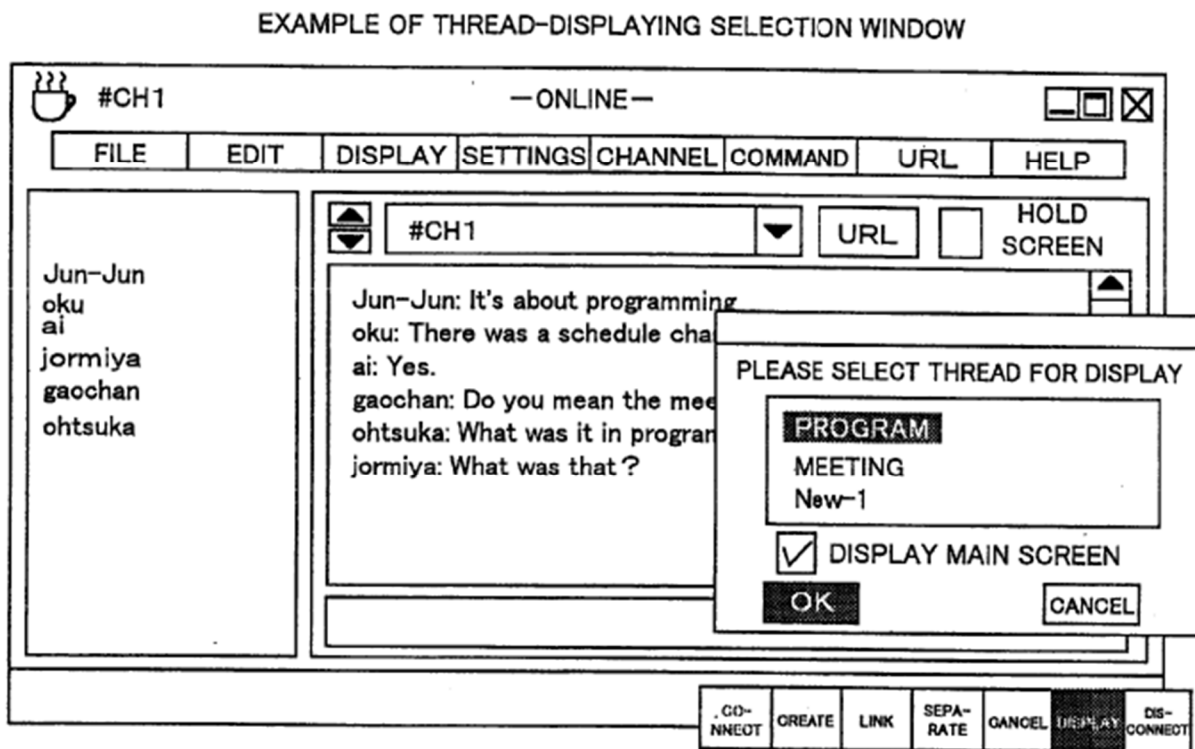
115. Limitation 1[p] requires:

in response to a user selection of said second user selectable control, cause at least a portion of said first subsidiary communication thread

containing said first messaging communication to be displayed in said third pane.

116. Dr. Cockburn asserts that the “Display” button meets this limitation. Ex-1003, ¶108. However, as I already discuss for limitation 1[o], the “Display” button cannot qualify as a “second user selectable control.” Also, the “Display” button does not cause any thread to appear in the thread window (the alleged “third pane”). Rather, as shown in Figure 12 of Kakuta, the “Display” button causes a thread selection window to appear.

Fig. 12



Ex-1005, Fig. 12.

117. A user must first select a particular thread title from the thread title list and then select “OK” before the selected thread window will appear. If the user selects cancel, no thread is displayed. Thus, Kakuta’s system does not “cause” a portion of a thread to appear in the thread window “in response to” user selection of the “Display” button. The intervening steps and user actions breaks any causal chain. A Skilled Artisan would have recognized that the pressing of the “Display” button starts the Displaying Process” of Figure 19 and discussed at Kakuta, 14:46-15:9. According to that description, the pressing of the button “directs the display of the FIG. 12 selection window.” Id. Because a user must take actions to continue and may press the “cancel” button, a Skilled Artisan would have recognized that the “Display” button cannot “cause” the display of a thread window.

B. Claims 2-27

1. Claim 2

Claim 2 requires:

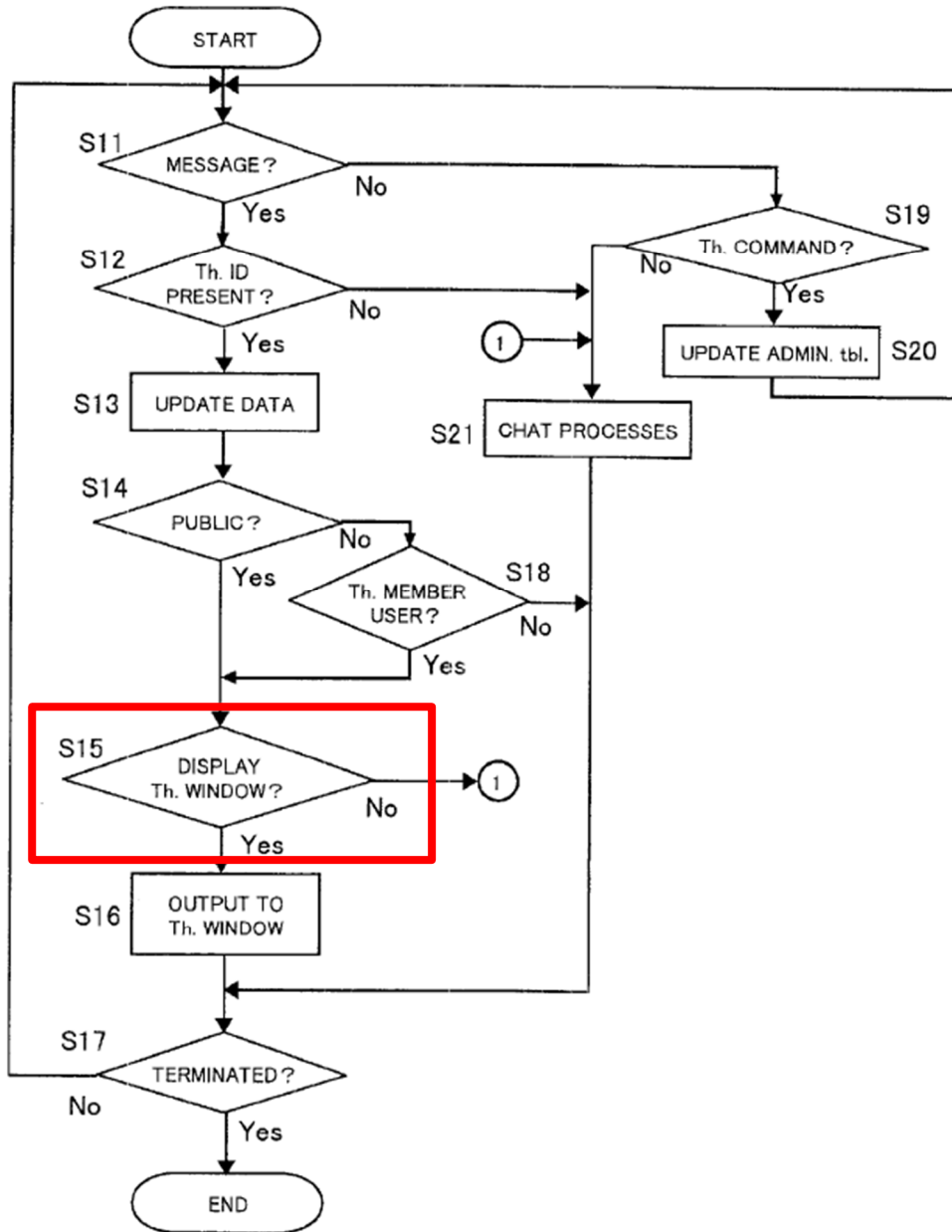
The apparatus of claim 1, further comprising instructions for causing the apparatus to display said additional messaging communication in said second pane while said additional messaging communication is being displayed in said third pane.

118. In Kakuta, when a thread is created and “Quote Messages” is checked, messages are copied **from** the chat window 303 **to** the thread window 307 (the

putative “third pane”). Ex-1005, 12:42-49, 8:20-29. These messages existed prior to thread creation and thus cannot be the “additional messaging communications” as claimed. Thus, none of Kakuta’s figures disclose “additional messaging communications” in both the alleged second and third panes.

119. Also, Kakuta is clear that additional messages are displayed **either** in the thread window 307 **or** in the chat window 303, **but never both**. This is shown in the flow diagram of Fig. 14, at the decision box S15.

OUTPUT CONTROL PROCESS



Ex-1005, Fig 14.

120. Kakuta explains: “Further, wherein in aforementioned step S15 the decision is that the window for the object thread is **not** being displayed, Step S21 ensues. Herein, in Step S21, the administration module 23 sends the thread

message to the control module 32. The thread message **not being displayed in the thread window is therefore displayed in the main screen.**” Ex-1005, 10:65-11:3. Thus, a Skilled Artisan would have understood that if the thread window is not open, only then would the system display the thread message in the main chat window 303. Using Figure 4 as exemplary, a Skilled Artisan would see that when the thread window is closed, main screen 301 has a single location to view messages, which is chat window 303, and therefore a Skilled Artisan would have understood that the messages would be displayed in chat window 303 in the eventuality that the thread window is not open.

121. Dr. Cockburn cites to Kakuta at 15:11-13, 9:23-31, and 15:18-22 to argue that Kakuta copies messages from the thread window to the chat window. Ex-1003, ¶112. That is incorrect. At 15:11-13, Kakuta refers to FIG. 7, in which messages were copied **from the main chat window to the thread window.** Kakuta states that Fig. 7 is “[a]n example of a thread window 307 when a new thread has been created.” Ex-1005, 8:30-31. Kakuta explains that a user can choose “whether or not to display **past messages** in the freshly created thread window.” Ex-1005, 8:20-22, 8:30-31. Therefore, the messages in the thread window of FIG. 7 cannot be the claimed “additional messaging communications,” which must be typed in after a thread is created. Ex-1005, 10:65-11:3.

122. Kakuta’s passage at 9:23-31 does not suggest that messages will

appear in both the thread window and main chat window. That passage describes Figure 12, which does not show simultaneous display of thread messages in two panes. Nothing in the passage suggests that thread messages entered into thread message input field 309 are simultaneously displayed in two windows. On the contrary, that passage discusses the opening of the thread window, in which case all thread messages are displayed in the thread window per the Figure 14 flowchart. I note that Petitioner and Dr. Cockburn may have misread column 9, lines 29-30. That passage states that the “thread window” is displayed on the main screen. Because the thread window is displayed, thread messages are displayed exclusively in that window. That passage does not state (and nor would it suggest to a Skilled Artisan) that threads are displayed in both the thread window and the chat window. In paragraph 114, Dr. Cockburn cites Kakuta, 14:51-58 and Figures 12 and 19, but these citations do not support Dr. Cockburn and the display of selection menus.

123. Finally, the passage at 15:18-22 of Kakuta does not support Petitioner’s position. *Id.* That passage concerns “disclosure levels” (whether threads are public or private) and not the specific windows in which thread messages are displayed. *Id.*

2. Claim 3

124. Claim 3 requires:

The apparatus of claim 2, wherein displaying said additional messaging communication in said second pane comprises providing a third user-selectable control associated with said additional messaging communication in said second pane, wherein said third user-selectable control is configured to:

in response to user selection of said third user-selectable control, cause at least a portion of said first subsidiary communication thread containing said additional messaging communication to be displayed in a third pane.

125. This is illustrated in Fig. 10G of the '731 patent, for example, which depicts an “additional messaging communication” by Sona H. at 8:33 am (the message “I suggest that you make sure Laura is happy first.”). This additional messaging communication is displayed in the second pane with a “SIDEBAR” icon that, when selected, causes the third pane to open to display the thread (as shown in Fig. 10H):

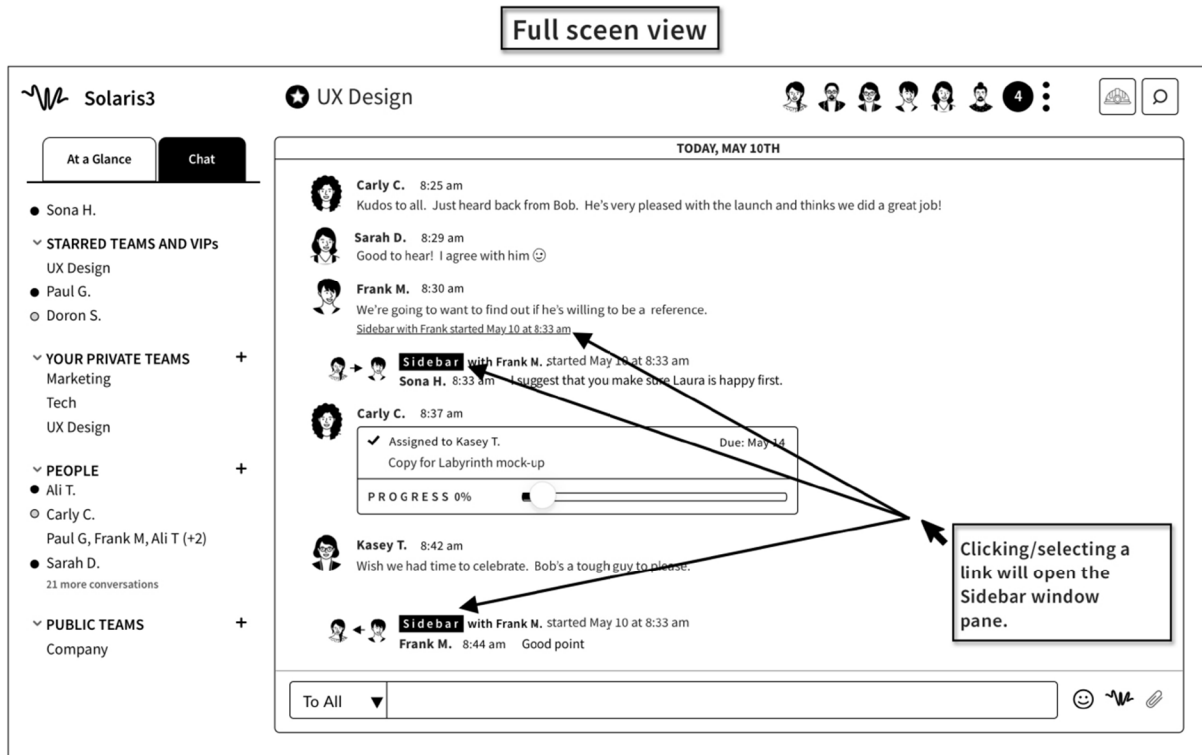


FIG. 10G

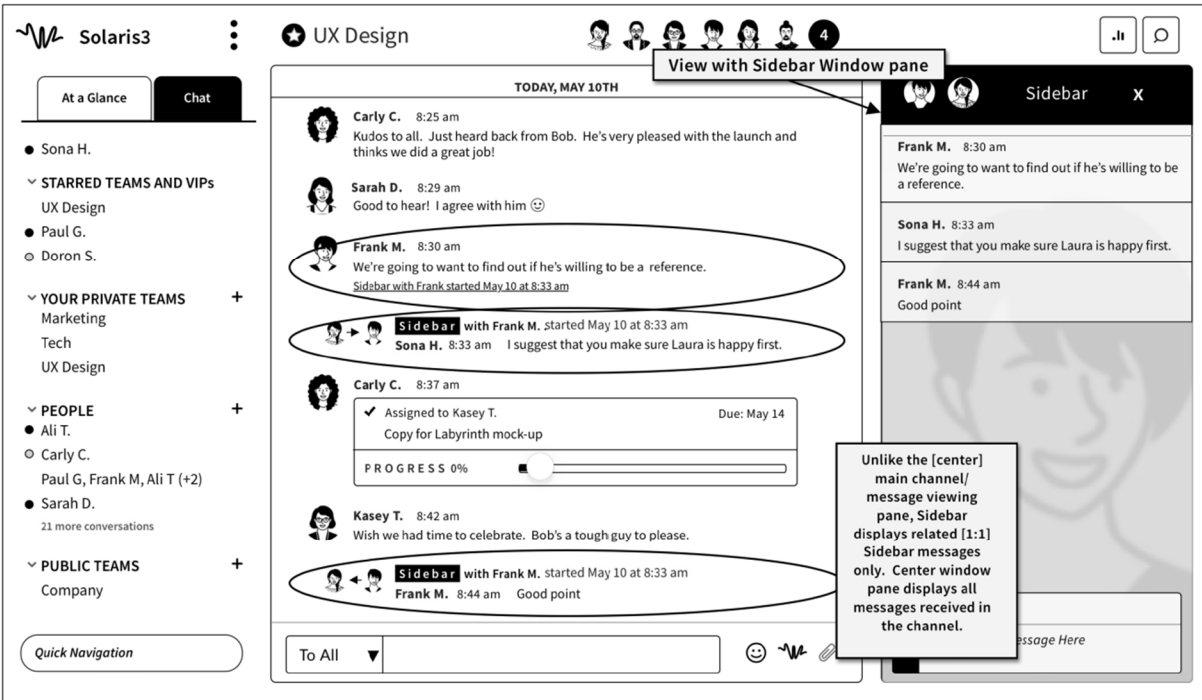


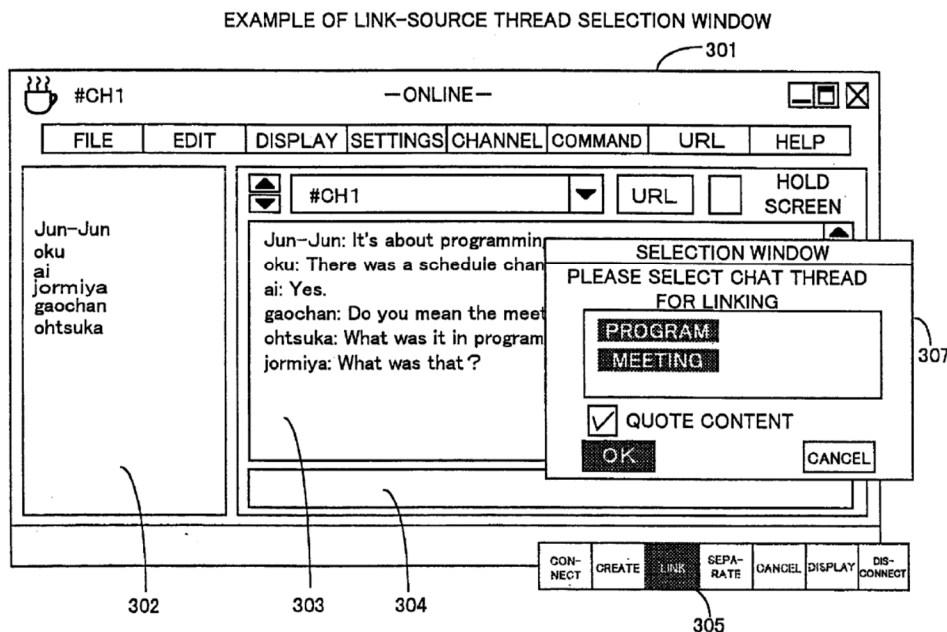
FIG. 10H

126. Kakuta does not disclose any third user-selectable control associated with “additional” thread messages displayed in the second pane that would trigger the opening of third pane containing at least a portion of the associated thread. In Kakuta, thread messages are displayed in the main chat window 30 (purported second pane) **only** if the thread window is not open, and Kakuta does not suggest there are any controls associated with the message when displayed in the main chat window.

127. Petitioner identifies the “Link” button on toolbar 305 as the alleged third control. See Fig. 10 reproduced below. But the “Link” button, like the other buttons on the floating toolbar of Kakuta, is not associated with particular

messages in the main chat window. The “Link” button causes display of a pop-up window with a list of the existing threads. Ex-1005, 9:5-17. The “Link” button thus performs a single function—it provides access to the thread selection window as shown in FIG. 10 of Kakuta, and it is not associated with any particular message or even with any particular thread.

Fig. 10



128. The selection window simply provides a listing of all threads available to the user so that a user may choose those threads the user wishes to link together. The operation of the “Link” button is in contrast to the “association” of the ’731 patent, which displays a third user-selectable control in a manner that associates it with the additional messaging communication. A Skilled Artisan would understand that the association claimed and discussed in the ’731 patent is such

that the user recognizes that the control operates on the associated communication, and causes the display of at least a portion of the thread containing the associated message. Kakuta has no control that performs such a function.

129. Petitioner and Dr. Cockburn also cite to Figure 7 of Kakuta in support (Ex-1003, ¶120), however, Figure 7 depicts the creation of a new thread and the display of past messages that existed before thread creation. Figure 7 does not depict any “additional communication messages” and thus is not relevant to claim 3 (all messages in Figure 7 were present before the thread was created, as shown in Figure 5).

3. Claim 4

130. Claim 4 requires:

The apparatus of claim 3, wherein providing said third user-selectable control comprises displaying said third user-selectable control proximate to said additional messaging communication.

131. An exemplary third user-selectable control is shown in Figure 10G of the '731 patent:

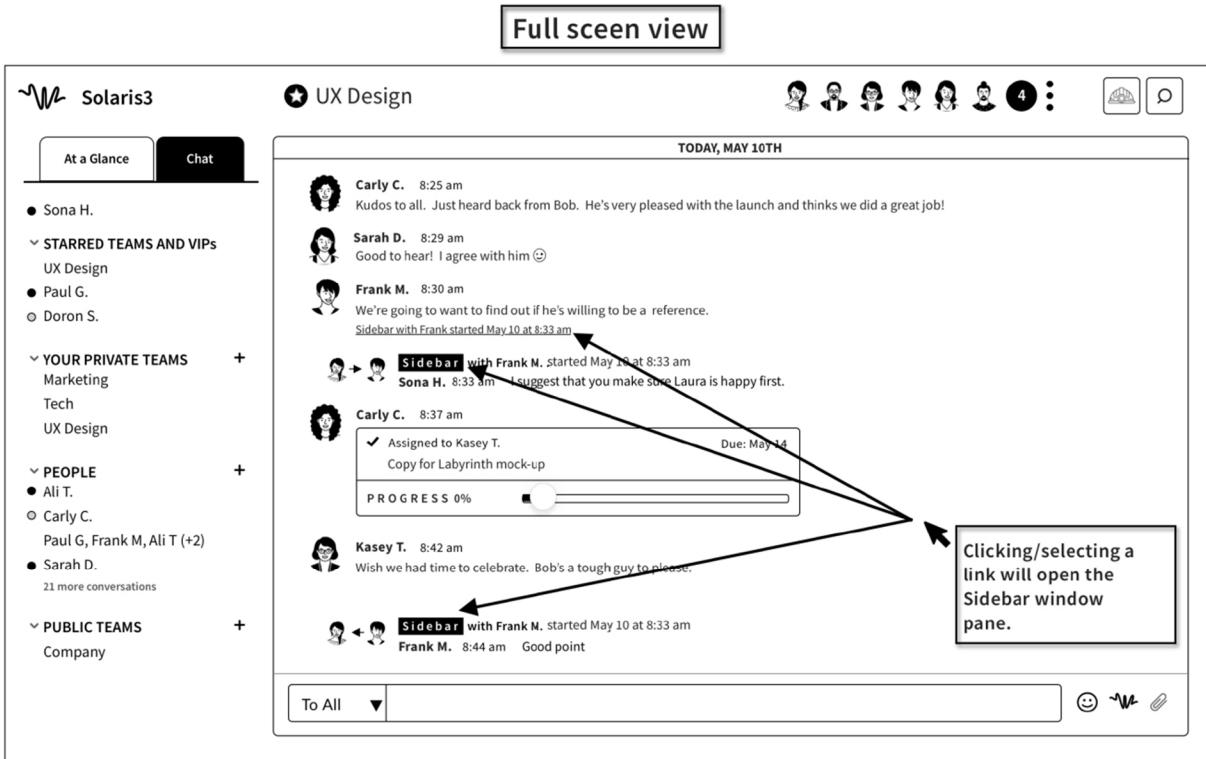
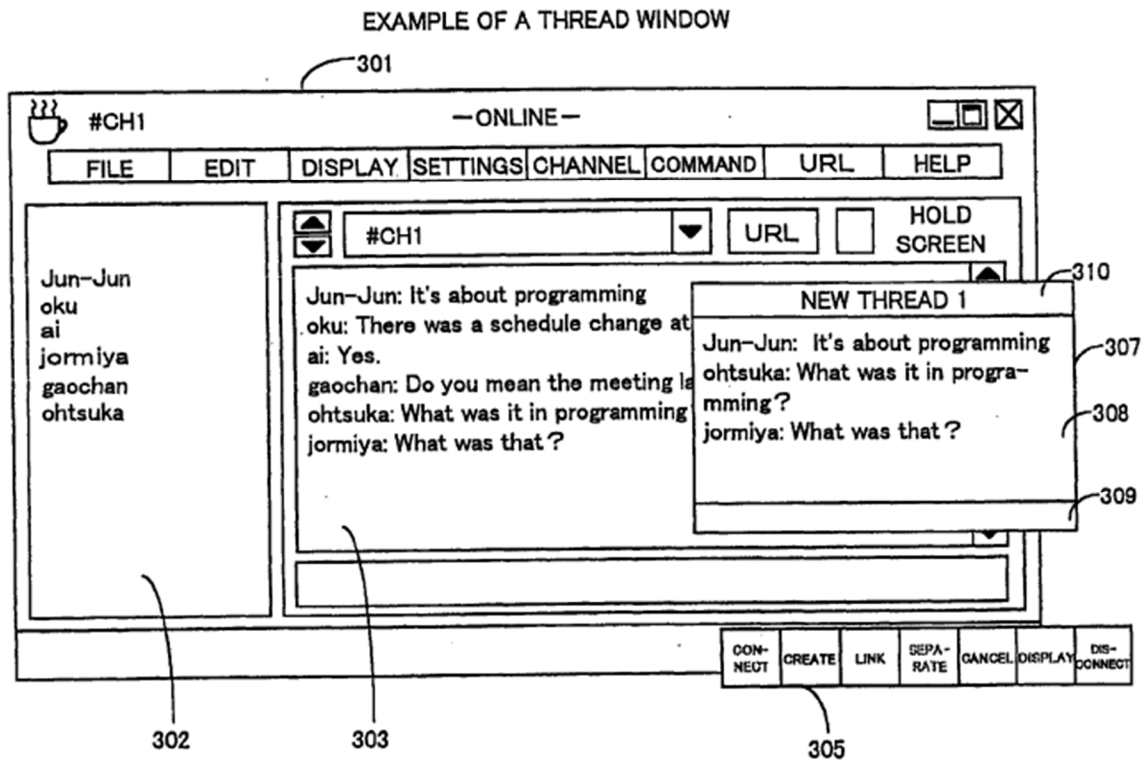


FIG. 10G

Petitioner relies again on Figure 7 of Kakuta (Ex-1003, ¶¶121-123), but Kakuta's Fig. 7 does not show such additional messaging communications in proximity to such control.

Fig. 7

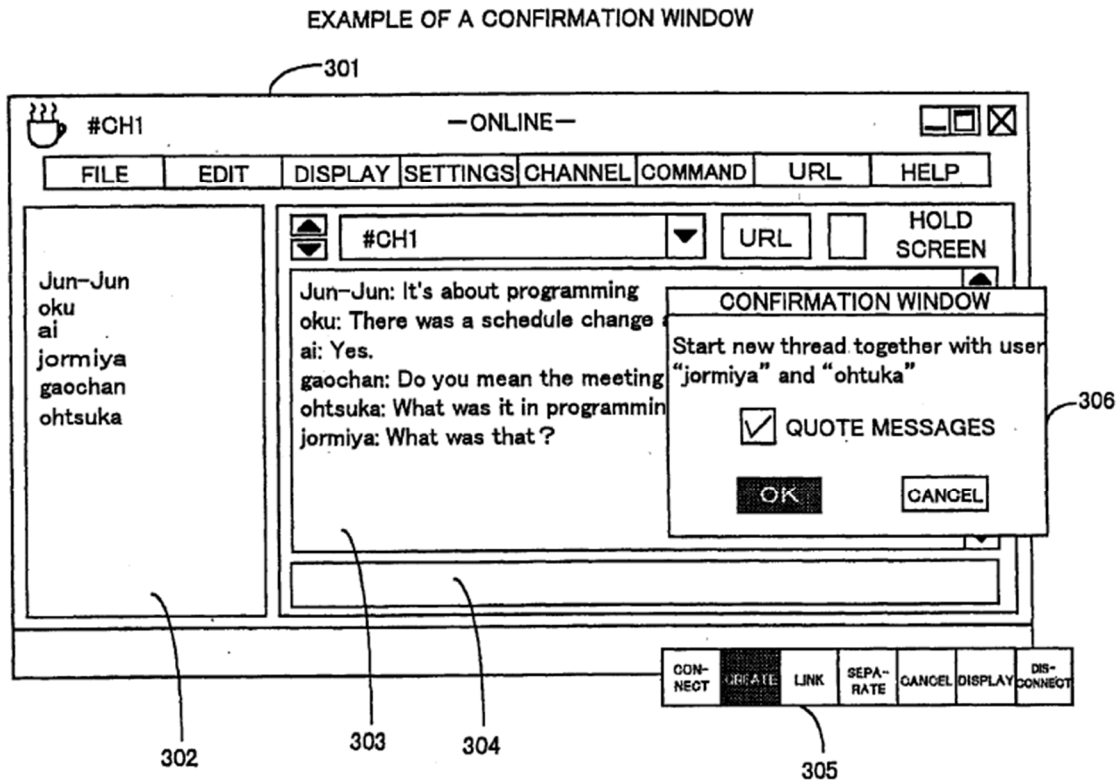


Ex-1005, FIG. 7.

132. None of the messages shown in Fig. 7 were entered into Kakuta's thread message input field 309. They are pre-existing messages as clearly shown in Fig. 6. Kakuta's Figures. 5-7 illustrate a progression in time. Figure. 5 depicts Jun-Jun selecting users and pressing "Create." Fig. 6 shows the resulting "confirmation window." Fig. 7 shows the thread window after Jun-Jun pressed "OK." The six messages in the main screen and the three messages in the thread window of Fig. 7 were already present before the thread was created, and can be seen in both Figs. 5 and 6. They therefore cannot be "additional messaging

communications.”

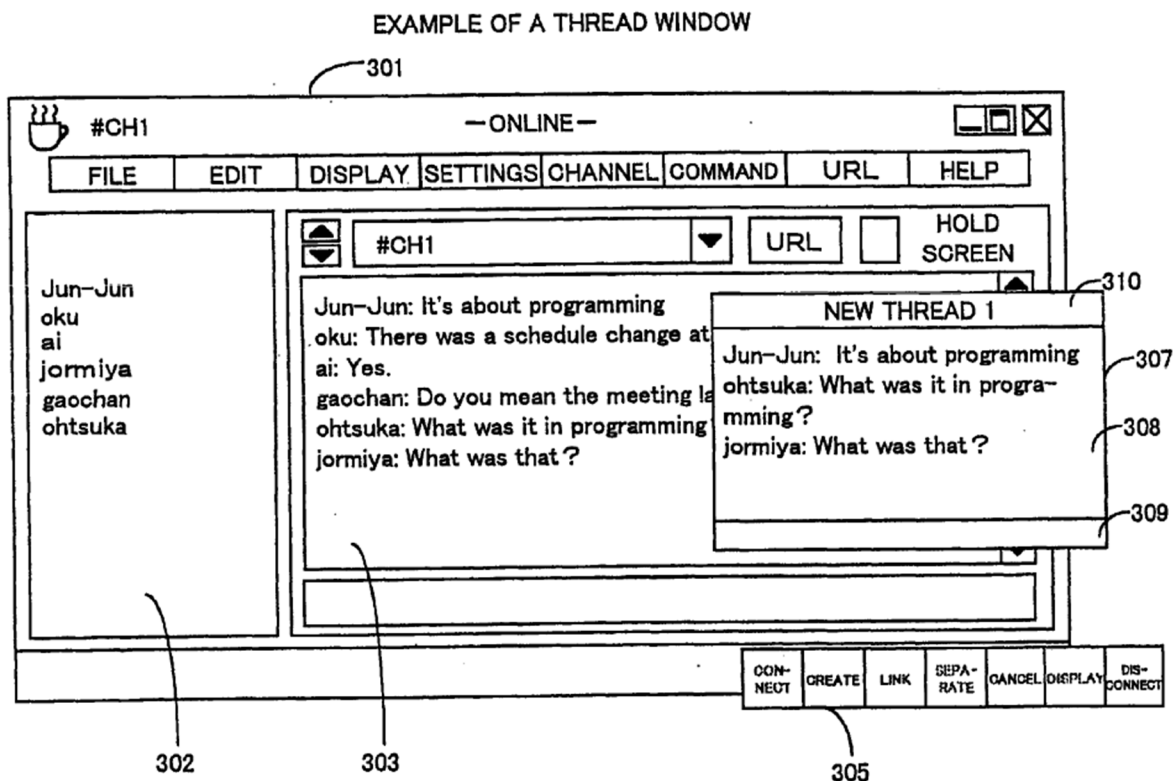
Fig. 6



Ex-1005, FIG. 6. Because there are no “additional messaging communications” displayed in Figure 7, there can be no control “proximate to” the non-existent “additional messaging communications.”

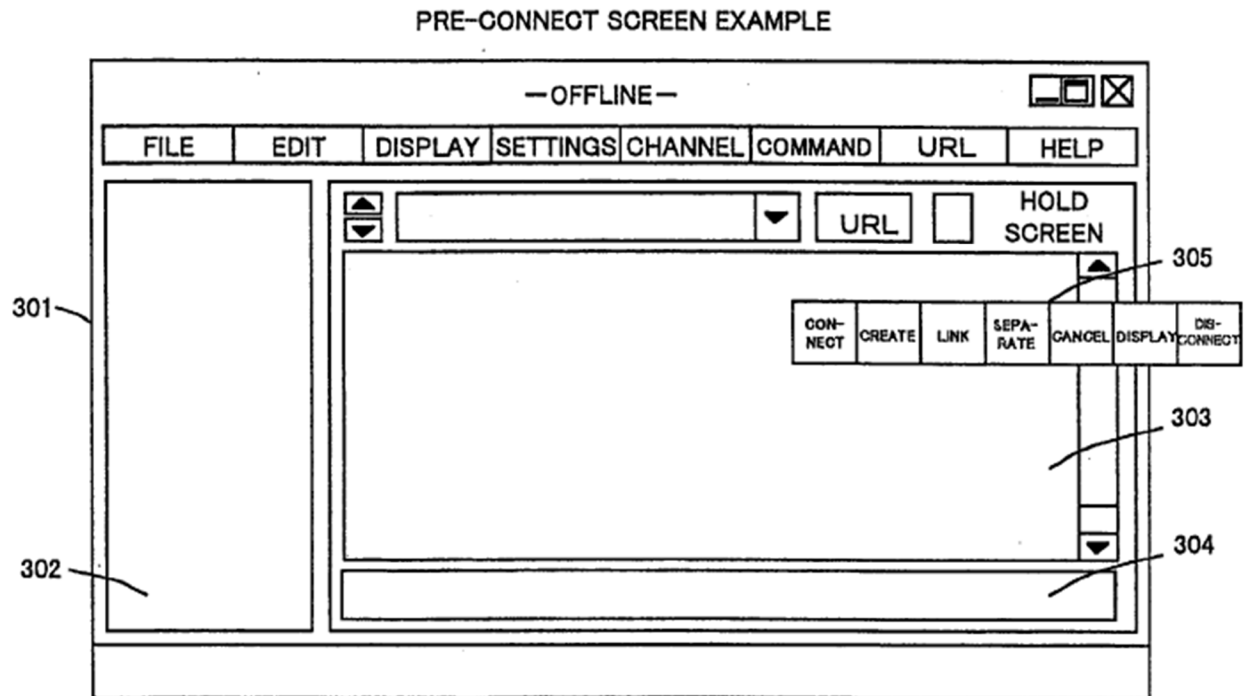
133. Further, Figure 7 does not show the toolbar 305 with the “Link” button (the alleged control) proximate to any message in the second pane. As shown in Figure 7 of Kakuta, the floating toolbar is positioned on its own at the bottom right of the display screen, entirely outside of the chat window (the alleged second pane).

Fig. 7



134. The proximity between a control and a message, as described in the '731 patent creates a visual relationship between such that a user would readily understand that the control relates to that message. An example of this is depicted in Fig. 10G of the '731 patent, depicted above. Kakuta provides no such visual relationship. Rather, the toolbar of Kakuta exists independent of any particular message and moves around the screen in a way unrelated to any message. A Skilled Artisan would understand that a floating toolbar is typically used so that the location of the toolbar does not matter. It can be moved so that it is not covering

areas the user wants to see. Thus, nothing about the location of the toolbar provides an indication of an association between the “Link” button and a particular message. In fact, the toolbar with the Link button is present before any messages are displayed, as shown in Kakuta’s Figure 4:



4. Claim 5

135. Claim 5 requires:

The apparatus of claim 2, wherein further messaging communications other than said first messaging communication and said additional messaging communications are also displayed in said second pane, and wherein displaying said additional messaging communication comprises displaying said additional messaging communication in the second pane in chronological order relative to said further messaging communications that are displayed in the second pane.

136. Again, the Petition relies on Figure 7 of Kakuta as allegedly depicting “additional messaging communications.” However, Figure 7 does not disclose any such communications, which must have been added through the thread message input field 309. In Figure 7, all messages existed prior to any thread creation, as is clear by comparing Figure 7 to Figure 5, which has the same messages.

5. Claim 6

137. Claim 6 requires:

The apparatus of claim 3, wherein causing at least a portion of said first subsidiary communication thread containing said additional messaging communication to be displayed in said third pane comprises displaying said sidebar text-entry field associated with said third pane that is operable to receive said additional messaging communication.

138. The Petition fails to establish anticipation of claim 6 for the same reasons as provided for claim 3.

6. Claim 7

139. Claim 7 requires:

The apparatus of claim 3, wherein displaying said additional messaging communication in said second pane comprises, while said additional messaging communication is displayed in said second pane, displaying identifying information associated with the anchor message in association with said additional messaging communication in said second pane.

140. Exemplary “identifying information” is depicted in Fig. 10H of the ’731 patent, which shows an identifier of the author of the anchor message (e.g., “Frank M.”) displayed in association with an additional messaging communication (e.g., the message from Sona H. at 8:33am):

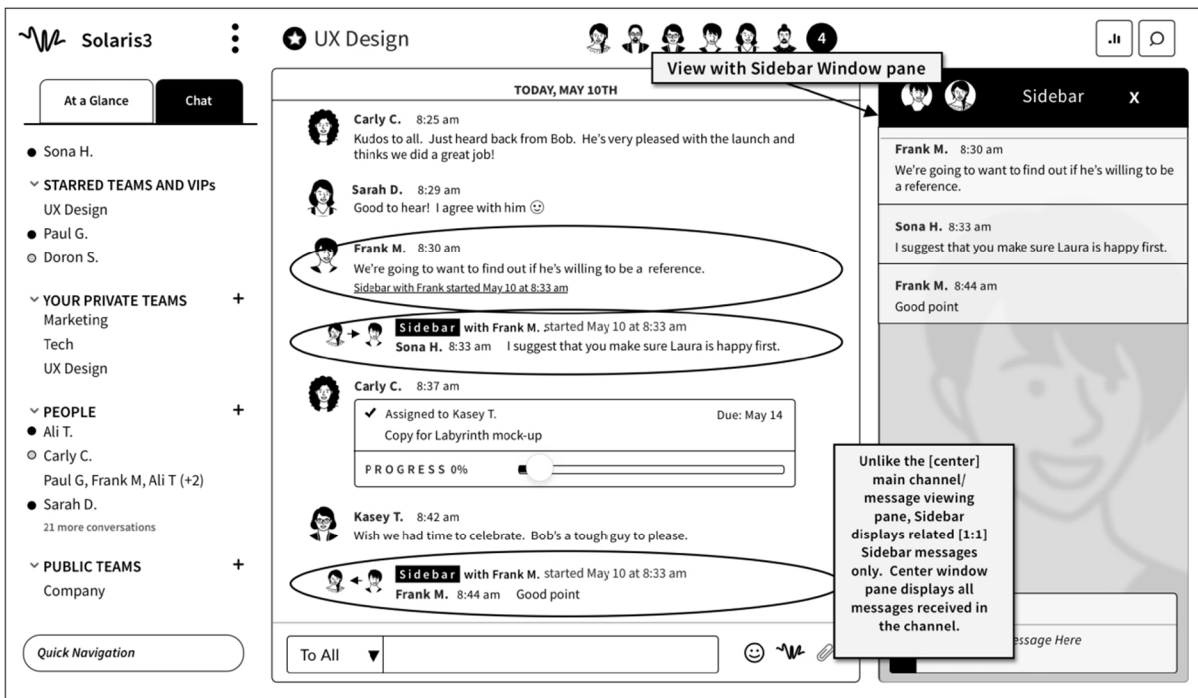


FIG. 10H

141. Kakuta has no such identifying information. The Petition relies on the author of the first message in the main chat window of Figure 7 (displayed in association with that message) as the “identifying information.” Ex-1003, ¶130. However, this is the wrong message. The “identifying information” must be “display[ed] . . . in association with said additional messaging communication.” The additional messaging communication must, in turn, have been entered via

thread text input field 309, which is not the case for the first messaging communication shown in the main chat window of Figure 7.. Kakuta thus lacks this disclosure.

7. Claim 9

142. Claim 9 requires:

The apparatus of claim 7, wherein at least some of said identifying information associated with said anchor message comprises a fourth user-selectable control operable to cause the display of at least a portion of said subsidiary communication thread in said third pane.

143. Kakuta lacks such a control. Petitioner identifies the selection of users or messages prior to pressing the “Create” button. Ex-1003, ¶133. However, the selection of users or messages is not a control whose selection causes the display of those messages in the third pane as claimed. Selecting such messages does not cause anything to be displayed in a third pane. A number of additional steps (that may or may not occur) must take place before messages will be displayed in the thread window of Kakuta.

144. Alternatively, Petitioner maps the “fourth user-selectable control” onto “the selectable user ID or author that is associated with an anchor message.” Ex-1003, ¶134. However, Petitioner does not identify what “selectable user ID or author” it is referring to. The term “user ID” is not found in Kakuta, and the term “author” is not used in connection with threads. Ex-1005, 6:45-46.

145. Petitioner relies on the following for the alleged identifying information: (a) Fig. 4, which has no selectable user IDs, authors, or messages, (b) Fig. 8, which discloses the selection of messages **before** thread creation and thus before the existence of an “anchor message,” (c) Fig. 16, step S251, which merely asks whether a selection has been made, and if so causes display of a confirmation window, (d) Fig. 5, which describes the “Create” button and (e) Fig. 7 which shows a thread created by selection of users in the user window 302 as shown in Fig. 6. Furthermore, none of these aspects of Kakuta meet the requirements of claim 7, from which claim 9 depends, that the control be comprised of identifying information that is (a) associated with an “anchor message” and (b) displayed “in association with” an “additional messaging communication.” The Petition ignores these details.

8. Claims 10, 11, & 12

146. Petitioner fails to establish anticipation of claims 10, 11, and 12 for the same reasons it fails to establish anticipation of parent claim 1. In addition, Kakuta fails to disclose the limitations found in claim 12. Claim 12 requires “a request to permit a non-member user” to receive messages from the “subsidiary communication thread.” Petitioner identifies no such request in Kakuta. The cited passage merely states that messages are sent over the network to indicate when users join or depart from a channel, as was common in IRC systems. Ex-1005,

11:6-10. This information allows each user's computer to update the administrations tables, such as those shown in Figure 3 of Kakuta. There is no description of enabling non-members to receive communications in a particular subsidiary communications thread.

9. Claims 15-19, 21-22, and 25-27

147. Petitioner provides no new arguments for these claims, but rather incorporates its arguments from claims 1-12. Accordingly, I incorporate my previous responses concerning those arguments to conclude that these claims are not disclosed in Kakuta

C. Claim 30 Is Not Anticipated By Kakuta

1. 30[c]

148. Limitation 30[c] requires:

receiving a thread request to create a subsidiary thread communication.

149. Petitioner incorporates arguments from claim 1, and I therefore incorporates my responses to those arguments to conclude that this limitation is not disclosed in Kakuta. In addition, Petitioner relies on the "Display" button on Kakuta's toolbar as a button that can "create a subsidiary thread communication." However, Kakuta's "Display" button does not create threads, it merely allows a user to select an existing thread for display. Ex-1005, 9:23-31.

2. 30[d]

150. Limitation 30[d] requires:

in response to receiving the thread request, generating and transmitting for display a thread summary indicator renderable within the messaging pane proximate the first messaging communication, wherein the thread summary indicator is configured to be selectable by a user with access to the first messaging communication;

151. Petitioner incorporates arguments from claim 1, and I therefore incorporate my responses to those arguments to conclude that this limitation is not disclosed in Kakuta. In addition, Petitioner and Dr. Cockburn rely on the highlighted “PROGRAM” thread title in Fig. 12 as a “thread summary indicator renderable within the messaging pane proximate the first messaging communication.” Ex-1003, ¶193. However, none of the thread titles from Figure 12 are “proximate” to any particular message. This would be so regardless of the location of the thread selection window on the screen. In Kakuta, there is no correlation between the location of the thread selection window and any particular message. For example, a user could not tell from the placement of the thread selection window which messages belong to which threads. In addition, there are multiple threads listed in the thread selection window. Thus, although Petitioner relies on the “PROGRAM” thread, Petitioner ignores the other threads listed there. Finally, thread selection window is not “within the messaging pane,” which Petitioner maps to “chat window 303.” See Petition, 74 (referring to VII.A.5 and A.8, mapping the “messaging pane” to the “second pane”).

3. 30[e]

152. Limitation 30[e] requires:

receiving an indication of an input indicating a selection of the thread summary indicator

153. Petitioner incorporates arguments from limitation 30[d], and I therefore incorporate my response to those arguments to conclude that this limitation is not disclosed in Kakuta.

4. 30[f]

154. Limitation 30[f] requires:

causing, based at least in part on receiving the indication of the input, rendering of the subsidiary thread communication to a thread communication pane within the communication interface, wherein a list pane, the messaging pane, and the thread communication pane are configured to be rendered within a defined display window as viewable simultaneously.

155. Petitioner incorporates arguments from claim 1, and I therefore incorporate my response to those arguments to conclude that this limitation is not disclosed in Kakuta. Petitioner further relies on Figure 7 and identifies the channel identifier element as a “list pane.” However, the area in Figure 7 (which a Skilled Artisan would not consider to be a “pane,” for the reasons I provide for claim 1) contains no list. Kakuta does not describe this area as providing a list of anything. A Skilled Artisan would have understood that the channel identifier identifies the

channel that the user has selected and to which the messages displayed in the chat window belong. The Skilled Artisan would have recognized that Kakuta, like traditional IRC, provides messages for one channel at a time. A user would select a channel from a channel selection screen (by pressing the “Channel” button near the top middle of the screen) and then the user would be presented with the user window 302 and chat window 303 for the selected channel. There would be no reason to provide a “list” of channels as the purpose of the area is to identify the channel that the user is currently viewing. Providing the names of other non-selected channels would be needlessly confusing and take up valuable real estate on the screen.

IX. GROUND 2: THE PETITION FAILS TO ESTABLISH OBVIOUSNESS OVER KAKUTA

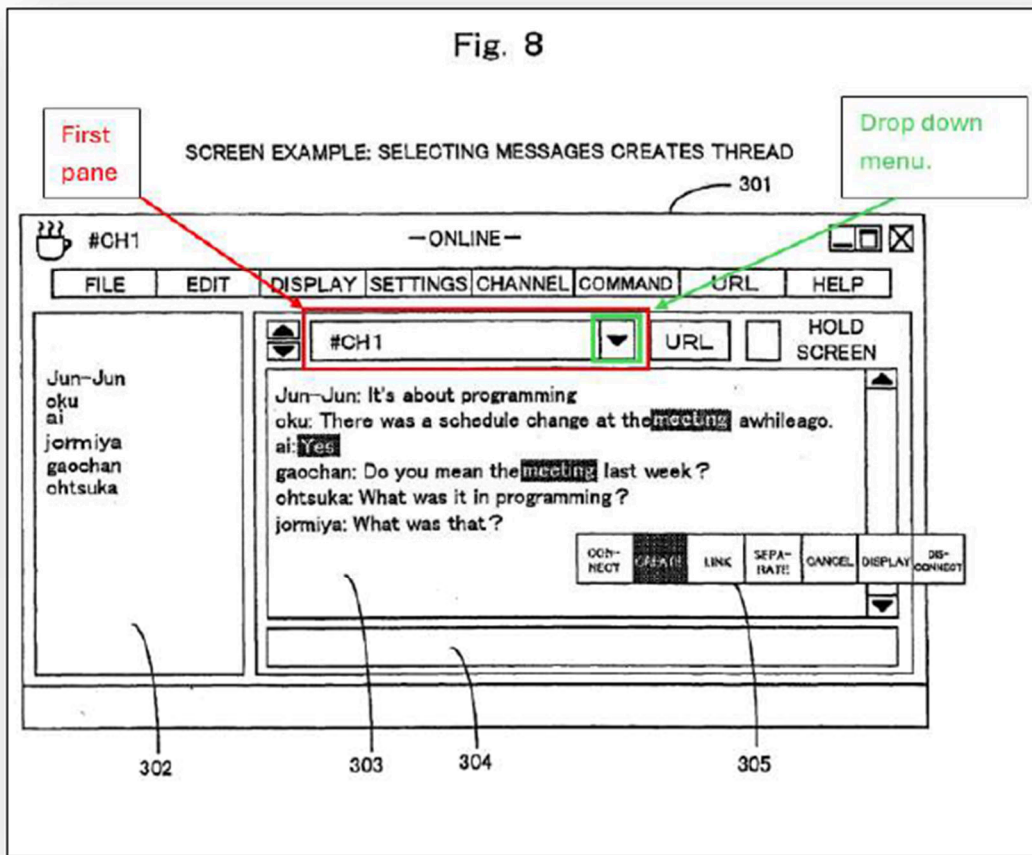
156. Petitioner largely relies on its anticipation arguments to which I have already responded. I therefore incorporate my discussion for Ground 1 here. Kakuta is missing numerous limitations from the claims, as I describe in connection with Ground 1. In my analysis for Ground 1, I explain the substantial differences between Kakuta and the claims, and for the reasons I provide in connection with Ground 1, nothing in Kakuta would suggest including the subject matter that Kakuta fails to disclose. I therefore conclude for these reasons, as well as the reasons stated below, that none of the challenged claims would have been obvious to a Skill Artisan in view of Kakuta.

157. I respond here to specific technical arguments in the Petition that may not be addressed in my responses to Ground 1.

A. 1[c], 15[b], and 25[b]

158. Petitioner asserts that notwithstanding Kakuta’s teaching of a display with only a single channel, “Kakuta inherently teaches displaying a plurality of channel identifiers.” Petition, 83. Dr. Cockburn makes this same statement. Ex-1003, ¶206. I am informed that inherency is an exacting standard and requires that “the missing descriptive material is necessarily present, not merely probably or possibly present, in the prior art” and that a claim limitation is inherent “only when the limitation at issue is the natural result of the combination of prior art elements.” A Skilled Artisan would not read Kakuta necessarily “displaying a plurality of channel identifiers” and therefore such a feature is not “inherent” in Kakuta.

159. Petitioner and Dr. Cockburn assert that Kakuta “inherently teaches” the plurality of channel identifiers “by providing a drop-down function, highlighted in green below.” Ex-1003, ¶206. I repeat Petitioner’s annotated figure here:

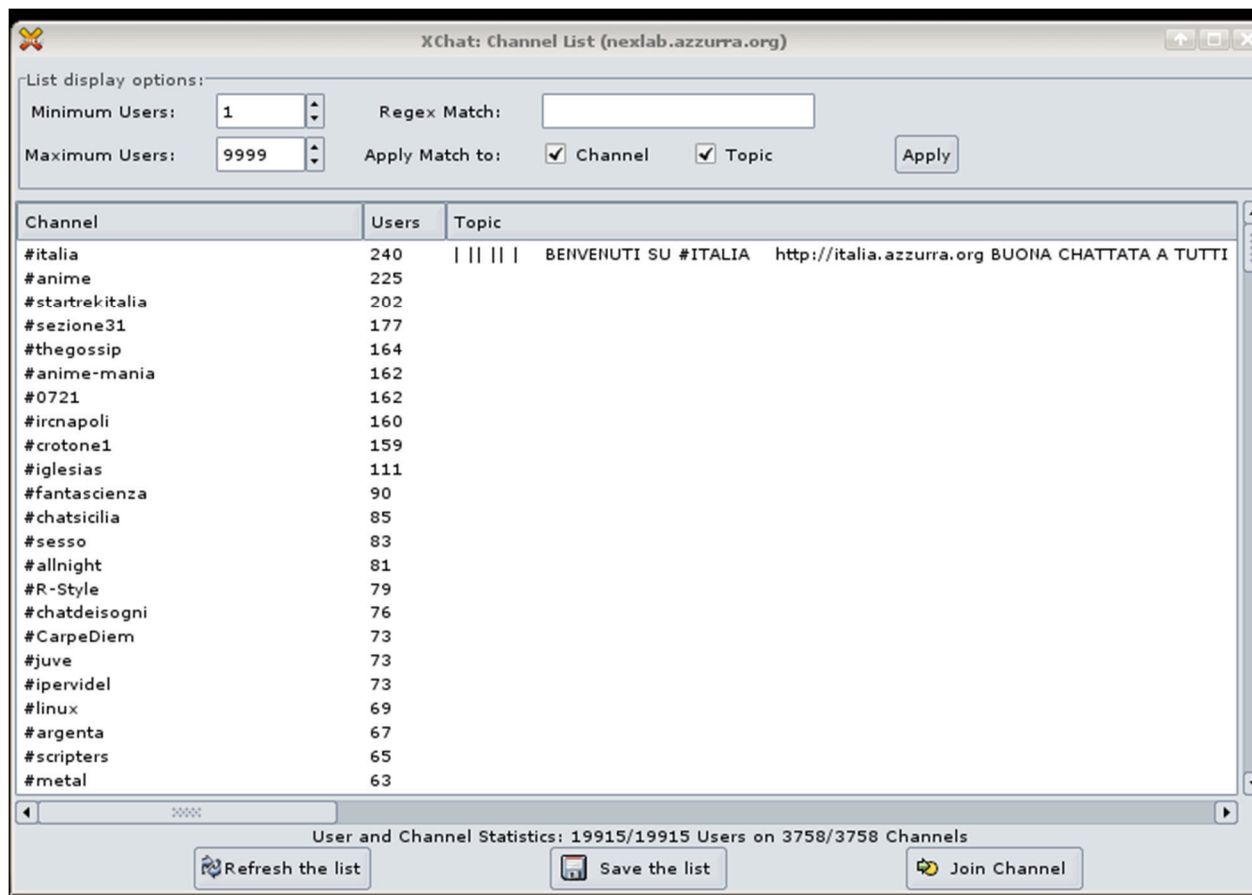


Id. Petitioner’s assertion that Kakuta provides a “drop-down function” is incorrect.

160. Kakuta merely discloses a down arrow (which Petitioner places within a green box) without any explanation as to what that down arrow does. It is not mentioned in the specification. Thus, Kakuta does not disclose a “drop down function.” The presence of the arrow does not inherently require a drop-down menu in the depicted system. Petitioner draws on the perceived similarity of the Kakuta figure and the “ComboBox widget” of Exhibit 1012. However, they are different. Exhibit 1012 provides a “Combos widget” that has a prompt (“Choose a play”) and a subject (“Play:”) followed by options (a listing of plays).

Ex-1012, 17. In Kakuta, there is no prompt, no subject, and no disclosed listing of options. In addition, the left side of the channel identifier has up and down arrows which are missing from the ComboBox. Thus, a Skilled Artisan would not conclude that Kakuta discloses a ComboBox.

161. In addition, even assuming that Kakuta's down arrow was part of a drop-down menu, a Skilled Artisan would not know the contents of that menu, and would not have been motivated to make that menu a channel listing because Kakuta discloses a "Channel" button in the collection of buttons along the top of the screen and consistent with traditional IRC systems, a Skilled Artisan would understand that the Channel Button would be used to select the channel for the chat window. An exemplary channel selection screen that would result from pressing the "Channel" button is shown here:



Ex-2008

(https://commons.wikimedia.org/wiki/Category:IRC_screenshots#/media/File:Azzurra_lista_canali.png).

162. In addition, a Skilled Artisan would not have used a drop-down menu in an IRC system like Kakuta because the number of channels is unbounded and typically quite large. The screenshot above, for example, has 3,758 channels. Thus, even though Exhibit 1012 states that a ComboBox “may be scrolled to accommodate large numbers of selections,” such “large number” of selections typical of a ComboBox are not in my experience unbounded or in the multiples of

thousands. The ComboBoxes of Exhibit 1012 contain a hardcoded list of finite options. However, the channel list would be continually changing as new channels are added to the list. A Skilled Artisan would thus not conclude that Kakuta intended to use the ComboBoxes disclosed in Exhibit 1012.

163. Finally, even assuming that Kakuta actually disclosed the use of a drop-down menu, such a menu would not be used for channel selection in Kakuta. Kakuta's system is focused on a single channel at a time, and Kakuta seeks to alleviate clutter in a single channel by creating threads. There would thus be little purpose of having a channel selector drop-down menu with potentially thousands of channels in the middle of the Kakuta chat interface given that Kakuta already provides a channel button (which a Skilled Artisan would expect to open a separate channel selection screen) for channel selection. Rather, given the focus of Kakuta on threads, a drop-down menu would provide access to certain threads within the designated channel. Such a drop-down menu, while not disclosed, would be more consistent with the Kakuta disclosure.

164. Dr. Cockburn cites to Exhibit 1012 and asserts in paragraph 207 that it was known that "when a drop-down option is selected, additional options, similar to the one chosen, will be displayed" and that "the drop-down would list different channels." However, Exhibit 1012 does not provide examples of IRC drop-down menus listing thousands of channels. Rather, Exhibit 1012 provides examples

comboboxes with a small number of hardcoded options. While comboboxes were well-known, the use of comboboxes for channel selection in IRC systems was not.

165. In paragraph 208, Dr. Cockburn states that Kakuta teaches that channel identifiers correspond to groups of users and that when users are selected to form a thread (message group), nicknames of the users will be used. Dr. Cockburn then asserts that “thus, Kakuta renders these claim elements obvious.” However, Dr. Cockburn’s observations concerning Kakuta are not relevant to the claim limitations being discussed. Limitation 1[c], for example, requires a “first pane configured to display a plurality of channel identifiers.” Kakuta’s system would have a plurality of channels, but Kakuta displays only a single channel in the chat screen. Dr. Cockburn’s paragraph 208 does not provide reasons why a skilled artisan would have modified Kakuta’s chat screen to list additional channels, and thus his analysis is deficient.

B. 1[d], 15[c], and 25[c]

166. For these limitations, Dr. Cockburn again relies on its unfound assumption that Kakuta inherently discloses “a drop-down option, allowing a user to select a channel identifier.” Ex-1003, ¶109. This assumption is wrong, for the reasons I just provided.

167. In paragraph 211, Dr. Cockburn states that “it would be obvious” that the channel identifier element “in Figure 8 is a combo box (i.e., drop-down menu),

and therefore, it is configured to display a plurality of items (i.e., more than one channel).” However, as I have previously stated, Dr. Cockburn fails to provide reasons for his conclusion, and his conclusion is based on two incorrect assumptions. First, he asserts that the channel identifier element “is” a ComboBox, and I have discussed previously why that assumption is incorrect. Second, even assuming there was a reason to modify Kakuta to include a ComboBox (and there is not), a Skilled Artisan would not use it for channel selection for the reasons I have discussed (most notably that channel selection is already provided by the “Channel” button.

C. 1[o], 15[n], and 25[n]

168. Petitioner asserts that “a PHOSITA would recognize that [the Display button] inherently indicates the anchor message of the subsidiary communication thread, making this feature obvious.” Petition, 86. I disagree. As I’ve stated, inherency requires that the inherent property be necessarily present. The “Display” button does not necessarily indicate an anchor message. In fact, the “Display” button provides no information about particular messages. No explanation is given this alleged inherency and no elaboration is provided by Dr. Cockburn.

169. In paragraph 213, Dr. Cockburn states that “a PHOSITA would have understood that the second user-selectable control, which generates the subsidiary communication thread, would indicate which message within the broader

communication channel serves as the anchor message for the thread.” Dr.

Cockburn is either paraphrasing the claim or applying hindsight, as he is describing functionality disclosed (and claimed) in the ’731 patent rather than the prior art.

Dr. Cockburn then asserts that Kakuta’s “Display” tool “acts as the second user-selectable control associated with the first messaging communication,” but this is incorrect for all reasons I discuss in Ground 1 for these limitations. Dr. Cockburn further states that the threads appearing in the thread selection window would have a message “designated as the anchor message” and cites Kakuta, 8:30-58. Nothing there discusses anchor messages and as I explained in Ground 1, Kakuta does not have anchor messages. Moreover, these claim limitations require an “indication” and Dr. Cockburn’s analysis ignores that aspect of the limitation, rendering the analysis flawed and deficient.

170. I have discussed the operation of the “Display” button when discussing Kakuta and when discussing Ground 1. The “Display” button triggers the thread selection window shown in Kakuta, Figure 12, which lists all available threads. Ex-1005, Fig. 12, 9:23-31. Thus, the “Display” button (which merely triggers a thread selection screen) cannot possibly indicate which message of a specific thread is an anchor message, and moreover, cannot possibly indicate which thread is contemplated (as the user has not made the thread selection yet). Petitioner does not attempt to provide a reason why a Skilled Artisan would

modify the “Display” button to provide the functionality of these limitations, and does not describe what these modifications would look like. On the contrary, a Skilled Artisan would have no reason to modify the functionality of the “Display” button.

D. 1[i], 1[p], 3[a], 13[a], 15[h], 15[o], 17[a], 23[a], 25[h], 25[o], 27[a], and 28[a]

171. For its obviousness argument, Petitioner and Dr. Cockburn assert that “it is inherent that the final step of selecting ‘Create’ triggers the display of the third pane with the anchor message.” Ex-1003, ¶216. This is incorrect. There is no “final step of selecting ‘Create’.” The act of clicking the Create button causes the display of the dialog box shown in Figure 6, asking the user to press “OK” or “cancel.” It does not “trigger” the displaying of a third pane.

172. Dr. Cockburn also argues that it would have been obvious to modify Kakuta’s “Create” button to provide an “immediate response” to any thread request. Ex-1003, ¶217. However, a Skilled Artisan would not reach this conclusion. Kakuta provides three ways to use the “Create” button. The first two ways allow a user to select either users or messages, in which case, the user must go through a confirmation window before generating the thread and displaying the thread window. A user is given various options, such as accepting or modifying the default state of “Quote Messages” and clicking either “Cancel” or “OK.” Kakuta also provides a third option, where an empty thread is created immediately

without any user selection. A Skilled Artisan would have understood that the immediate display of the thread window is only possible if no selections are made and that one could not modify the “Create” button, consistent with Kakuta’s disclosure, to provide an “immediate response” when a user makes a selection prior to pressing “Create.” The two possibilities are mutually exclusive. Kakuta teaches that if you want to select users or members, you have to take the path with intermediate steps and if you want an immediate thread window, then you cannot make any selections. By presenting the choices in this way, Kakuta teaches away from providing the feature that Dr. Cockburn asserts is an obvious variant on Kakuta—providing message selection with an “immediate response.”

173. Kakuta teaches the Skilled Artisan that interim steps are desirable and needed when a user wishes to select messages that will be used to create a thread. Indeed, Kakuta provides a detailed flowchart detailing the operation of the “Creation process” in Figure 16, and in that chart an immediate response is only available if no selection is made, in which case step S352 is skipped, but no messages are displayed in the thread window as a result.

174. Finally, Dr. Cockburn states that it is “inherent” that interim windows, such as the “confirmation window” are part of the process “that leads to the final display of the third pane.” Ex-1003, ¶218. But the final display is not inevitable as the user may press cancel (which is also part of the process). Thus, the interim

window does not necessarily lead to the display of a thread window (the alleged third pane). From this improper premise, Petitioner concludes that the challenged limitations are “obvious.” No analysis is provided. Petitioner fails to provide a reason why a Skilled Artisan would modify Kakuta to arrive at the claimed invention upon appreciating that Kakuta has interim windows. Kakuta’s “confirmation window” provides functionality important to Kakuta (such as the “Quote Messages” checkbox) and a Skilled Artisan would be motivated to keep the interim window rather than jettison it.

E. Claim 4

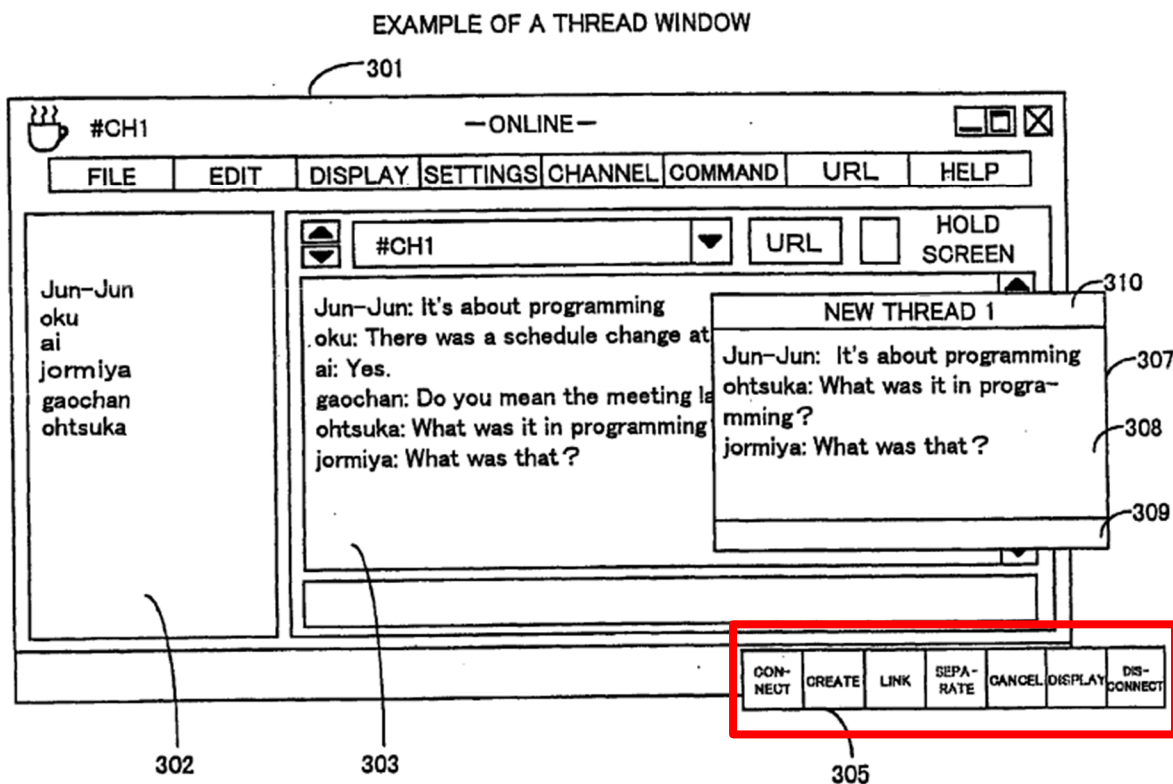
175. Claim 4 requires:

The apparatus of claim 3, wherein providing said third user-selectable control comprises displaying said third user-selectable control proximate to said additional messaging communication

176. Dr. Cockburn asserts that “messages and user selections are naturally proximate to the additional messaging communications when the user-selectable control is invoked.” Ex-1003, ¶220. However, the claim states that “said third user-selectable control” must be “proximate” to “said additional messaging communication.” This third user-selectable control is not identified in the Petition. Regardless, there is no support in Kakuta for the conclusion that “messages and user selections” are “naturally proximate” to the “additional messaging communication.” On the contrary, Kakuta discloses that additional messaging

communications will be displayed either in the thread window or, if the thread window is closed, in the main chat window. The controls are found on Kakuta's floating toolbar, which is placed on the screen at various locations that are unrelated to any particular message. For example, in Figure 7, the toolbar is shown in an area that would not be considered "naturally proximate" to any message:

Fig. 7



Ex-1005, Fig. 7.

177. In Kakuta, the toolbar is displayed prior to the selection of users or messages, and its location changes based at least in part on what windows are open (and not what messages are being displayed). The modification that Petitioner is

suggesting is unclear. To the extent that Petitioner envisions the toolbar jumping around the screen every time a user selects a message, I disagree that a Skilled Artisan would make such a modification (which would not meet the claims, in any event). Such a modification would be a change to the fundamental operation of Kakuta, and moreover would be highly distracting and disconcerting to the user. No motivation for making this change is found in Kakuta. Petitioner is applying hindsight based on the teachings of the '731 patent.

178. Dr. Cockburn further asserts that “proximity” of controls “enhances usability, making the process straightforward and logical for the user.” Ex-1003, ¶221. This is hindsight reasoning, using the teachings of the '731 patent as a guide. Nothing in Kakuta suggests using proximity as an indication of association. Dr. Cockburn further asserts that “the design and functionality of the system inherently suggest that the user-selectable control would be positioned near the additional messaging communications,” but the opposite is true—Kakuta is designed and functions in such a way that proximity is irrelevant. The Kakuta controls are on a floating toolbar that is placed at various locations unrelated to any specific additional messaging communication.

F. Claims 9, 12, and 21

179. Petitioner concludes that Kakuta renders these claims obvious, but Petitioner’s (and Dr. Cockburn’s virtually identical) discussion ignores numerous

claim limitations. Because the analysis does not track the claim language, the analysis is not relevant to the challenged claims. The Petition (and Dr. Cockburn) fails to provide the reasons why a Skilled Artisan would allegedly modify Kakuta to arrive at the claimed invention, and fails to explain how such a modification would function. Ex-1003, ¶¶222-226.

180. Dr. Cockburn states that the selection of a “user ID” in Kakuta “is operable to cause” the display of a portion of a thread “because the selection influences the behavior of the information that is included in the selection window.” Ex-1003, ¶¶223-224. But, Dr. Cockburn does not identify the “selection of the user ID” in Kakuta. And the conclusion does not logically follow from the premise. A control that “influences the behavior of the information” does not disclose a control “operable to cause” a display.

181. Regarding claim 12, Petitioner asserts that Kakuta “teaches the ability to manage user access and participation within a channel” and also teaches a “thread creation module” and that “the same mechanisms used to control thread visibility and user participation could be employed to grant access to non-member users.” Ex-1003, ¶¶225-226. I understand that the question is not what a Skilled Artisan “could” do, but what a Skilled Artisan “would be motivated to” do. Petitioner does not explain why a Skilled Artisan “would be motivated to” modify Kakuta to use its mechanisms to control “thread visibility and user participation” to

“grant access to non-member users.” A Skilled Artisan would not make such a change. Kakuta already provides public and private disclosure levels for threads for users within a channel, and moreover, each channel has its own user base. A Skilled Artisan would not have modified Kakuta to provide access beyond the channel, and the Petition fails to explain how this would be accomplished even if a Skilled Artisan were so motivated. In addition, Petitioner does not address the requirement that there be a “request to permit a non-member user” to receive thread messages. There is no reason to build such a request into Kakuta, which is focused on a single channel at a time. If a user wished to view public threads in Kakuta, the user would join the channel and then the user could use the “Display” function to view those threads.

G. Claim 13

182. Claim 13 (which is not challenged in Ground 1) requires:

The apparatus of claim 1, further comprising instructions for displaying a fifth user-selectable control, said control being operable to:

in response to user selection of said fifth user-selectable control, selectively display messaging communications having a specific type associated with said selected fifth user-selectable control, said messaging communications being from at least two separate group-based communication channels.

183. Figure 11 is exemplary:

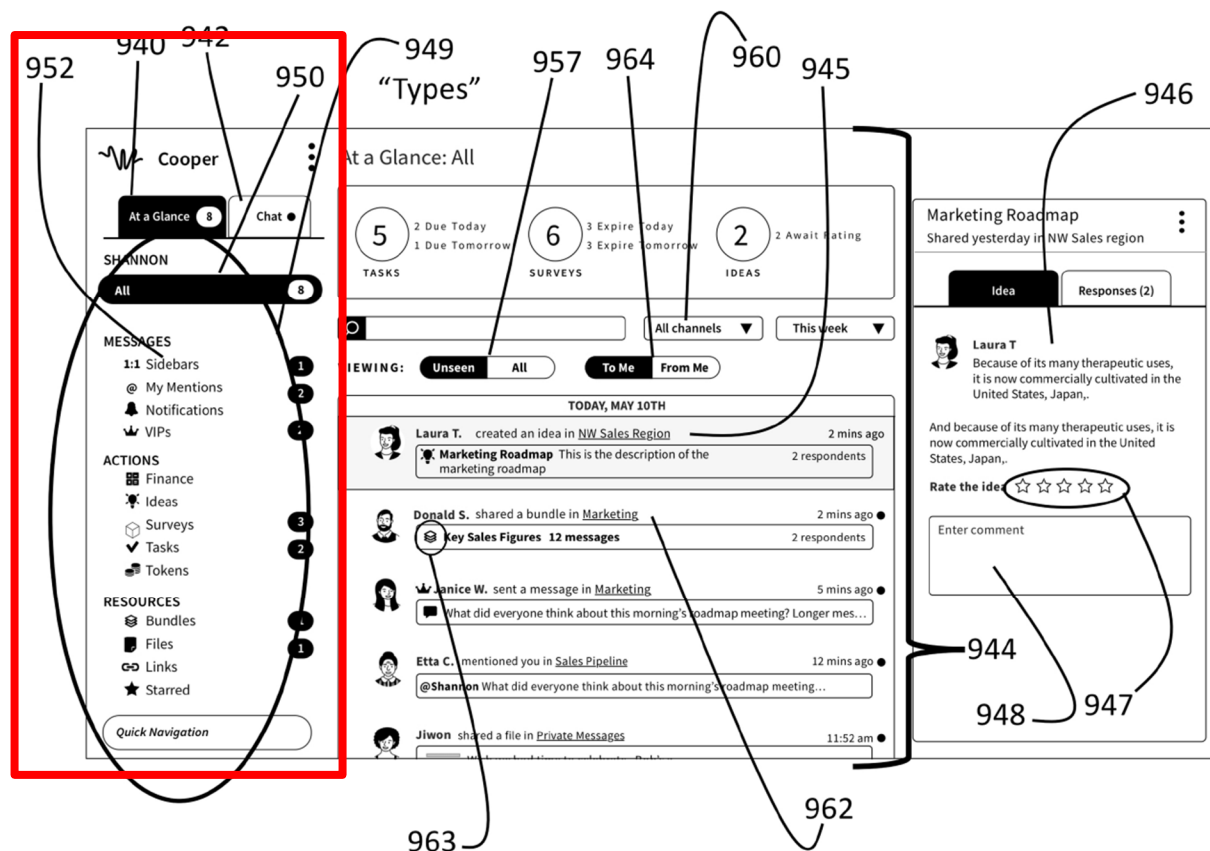


FIG. 11

Ex-1001, Fig. 11.

184. Dr. Cockburn asserts, without a proper analysis, that it would have been obvious to modify Kakuta to display messages from all available channels in the second pane and select multiple threads from those channels (using the “Link” button) for display in a third pane. Ex-1003, ¶¶231-232. But even assuming there was a motivation to do this (and there is not), the modification does not result in the control of claim 13. There is no fifth user-selectable control in the Petition’s hypothetical scenario that is associated with any “specific type” of messaging communication as the claim requires (certainly not the “Link” button), nor does

using the “Link” button cause messages “of a specific type” associated with it to be displayed. Rather, as described in Kakuta, the “Link” button creates a new thread visible to the user using the “Link” button that is the combination of two or more other threads selected from a subsequently appearing thread selection window. Ex-1005, 8:58-9:22. The “Link” functionality is unrelated to selectively displaying messaging communications of a specific type.

185. In addition, a Skilled Artisan would not modify Kakuta such that all channels are displayed in the chat window. Such a modification would be unwieldly and defeat the purpose of Kakuta. Kakuta and other IRC systems use channels to separate conversations on different topics. There is no reason why a user would want to intermingle all topics and display all the messages jumbled together. This is particularly so when one considers that IRC chat systems had thousands of channels covering disparate topics, such as shown in Exhibit 2008.

H. Claim 14

186. Claim 14 (which is not challenged in Ground 1) requires:

The apparatus of claim 13, wherein said specific type is a class of messaging communications in which the messaging communications are all parts of subsidiary communication threads that incorporate messaging communications generated by a specific user of said system.

187. Petitioner states that Kakuta’s system “inherently supports the organization of messages by specific users within subsidiary threads” and “[it]

would have been obvious to extend this functionality to create a class of messaging communications that are all parts of subsidiary threads generated by a particular user.” Ex-1003, ¶235. This statement does not support the obviousness of claim 14 because claim 14 does not require merely the existence of a “class of messaging communications,” but instead requires that the specific user-selectable control of claim 13 be “associated with” that specific class and cause the class to be displayed when it is selected. Regardless, Petitioner does not provide a reason why a Skilled Artisan would make the proposed modification.

188. Additionally, Dr. Cockburn asserts that Kakuta “teaches the ability to select messages from a specific user, creating a thread consisting entirely of that user’s messages.” Ex-1003, ¶234. Petitioner misreads Kakuta. The passage cited by Petitioner at column 8, lines 38-42 discusses the creation of a thread shown in Figure 7 of Kakuta, in which users are selected to form a thread. Figure 7 depicts a thread with **three** people based on a selection of **two** people (the thread creator is included in the thread as well). Ex-1005, FIGs. 5 & 7, 8:12-19. Thus, if a user were to select a single specific user, the resulting thread would consist of messages from both the selected user and the thread creator, and thus Petitioner is incorrect when it asserts the resulting thread would have only one user’s messages.

I. Claims 23, 24, 28, and 29

189. The Petition incorporates prior arguments and thus I incorporate my

prior responses to those arguments to conclude that it would not have been obvious to modify Katuta to make these claims.

J. Claim 30

190. Petitioner argues that it would have been obvious to modify Kakuta to use a “thread summary indicator” by “integrating the ‘Display’ tool with the selection of the desired thread into a single step.” Petitioner again applies hindsight—Petitioner starts with the claim and then asks what modifications to Kakuta would meet the claim language. Petition, 98. The alleged reasons for making the modifications (e.g., efficiency, simplicity, and usability) are conclusory and generic, and do not explain why efficiency, simplicity, and usability would produce the **claimed** functionality rather than something else. Neither the petition nor Dr. Cockburn explain why a desire to “streamline the user experience” would drive a Skilled Artisan to “integrate the ‘Display’ tool with the selection of the desired thread,” as opposed to other possible “streamlining” modifications. Ex-1003, ¶243. Moreover, Dr. Cockburn does not explain why a Skilled Artisan would have considered Kakuta’s display process (Ex-1005, Fig. 19) inadequate and in need of streamlining in the first place. Nor does Dr. Cockburn explain how a Skilled Artisan would “integrate” a thread listing with a button or how this alleged “integration” would enhance simplicity and usability rather than reduce it. For example, the “Display” button in the modification would operate differently

than other buttons on the toolbar, would make the “Display” button more complicated than before, and would leave the user with fewer display options than provided by Kakuta. Petitioner’s expert further asserts that the “integration” would allow the user to “invoke” the “display” function “directly within the message pane,” but provides no explanation as to why this is so. Kakuta’s “Display” button is on a floating toolbar outside the main chat window and would remain there after the “integration.” Nothing in Kakuta or the other cited prior art would suggest any modifications similar to what is claimed in the ’731 patent.

191. Kakuta teaches the Skilled Artisan that it offers a thread display that can “suit each person’s liking” or users may “designate that one another’s displays be made in common, enabling them to carry out efficient communication.” Ex-1005, 4:42-47. Nothing in Kakuta would suggest any modifications similar to what is claimed in the ’731 patent. Kakuta provides a “Display” button that generates a window with a listing of threads by thread title. Ex-1005, Fig. 12. Petitioner does not explain why a Skilled Artisan would have been motivated to “integrate” the “Display” button with a thread listing or how this would be done. To the extent that Petitioner asserts it would be obvious to abandon the “Display” button on the floating toolbar, such a change would be a significant change to the fundamental operation of Kakuta, and not one that a Skilled Artisan would have been motivated to make, as the floating toolbar is fundamental to the Kakuta

system.

X. GROUND 3: KAKUTA/MATSUMOTO

A. Matsumoto Fails to Fill the Gaps in Kakuta

192. In Ground 3, Dr. Cockburn argues that it would have been obvious to incorporate Matsumoto's "channel window" into Kakuta in order to satisfy the "first pane" requirements of limitations 1[c], 15[b] and 25[b] because it would provide a convenient way to "select among various channels." Ex-1003, ¶248. However, Matsumoto's "channel window 36" (*e.g.*, in Fig. 3 of Matsumoto) is not a channel selection window for "main chat window 32" (which would correspond to "chat window 303" in Kakuta). Rather, in Matsumoto, there are two chat windows. A "main chat window 32" containing messages from a single channel and a "sub-chat window 34" that contains intermingled messages from additional channels. Channel window 36 exists because of the "sub-chat window" and Matsumoto's ability to display messages from multiple channels at once and further to provide channel correlations between the multiple displayed channels. Because Kakuta does not have a "sub-chat window 34" and, like a typical IRC system, displays a single channel at once, there is no reason a Skilled Artisan would consider incorporating the "channel window 36" of Matsumoto into Kakuta. That window is unique to the Matsumoto design, lists the multiple channels being displayed on the screen, and is not relevant to Kakuta's system. Ex-1006, 9:26-48.

193. Dr. Cockburn states that “[i]t would be obvious to combine Matsumoto’s plurality of channel identifiers displayed in the channel window 36 with Kakuta’s user interface” and “[i]nstead of using a drop-down menu, Kakuta’s device could be enhanced by directly listing multiple channels in the first pane, as Matsumoto demonstrates.” Ex-1003, ¶248. This argument is flawed because it identifies no motivation to modify Kakuta to include a listing of multiple channels in the channel identifier element. As discussed above, the function of Matsumoto’s channel window 36 is to list the multiple channels being displayed on the screen (which includes both the main chat window and sub-chat window 34). Channel window 36 is not for selecting channels for display in the main chat window or the sub-chat window. Since Kakuta’s chat window 303 only displays a single channel, there would be no motivation for a Skilled Artisan to add Matsumoto’s channel window 36 to Kakuta, since it would have no utility in Kakuta’s design. Conversely, Kakuta already provides a mechanism for choosing among channels – the “Channel” button, and so a Skilled Artisan would not have been motivated to search for a different solution.

194. Matsumoto does not disclose how the main channel was selected, and presumably it was selected using a typical IRC channel selection screen rather than a “pane” that would be visible simultaneously with the main chat screen. Matsumoto contains no teachings concerning how a channel is selected for the

main chat channel. Thus, Matsumoto has no teachings that would support the proposed modification.

195. For claims 13 and 14, Dr. Cockburn asserts that Matsumoto teaches the forwarding of messages based on “specific conditions” that can be “tailored towards specific types of messages” and that “users can create filters that automatically transfer/forward messages across channels” to create a new channel with messages from multiple channels. Ex-1003, ¶¶251-253. From this disclosure, Dr. Cockburn asserts that in a combined system, when a user selects a channel for display “and the user selects the destination channel of Matsumoto’s filters, the user will see messages that original from a multitude of different channels” and that “the fifth user-selectable control is the Matsumoto filter-destination channel represented in Kakuta’s channel selector drop-down box.” Ex-1003, ¶254. There are numerous problems here. First, there is no “drop-down box” in Kakuta and I have explained why drop-down channel selector would not be used in Kakuta. Second, a channel is not a control, and thus Matsumoto’s filter-destination channel cannot be the claimed “control.” Third, in any combination, a user would not see messages from “a multitude of different channels.” A Skilled Artisan would not deviate from the fundamental structure of Kakuta, which displays messages from a single channel at a time, and seeks to simplify that display by providing a thread window. Moreover, Dr. Cockburn does not explain

what the combination he envisions would look like or how it would meet each limitation of the claims. For example, the “Link” button is not associated with any “specific type” of messaging communication, nor does its selection cause display of messaging communications of that “specific type” as claimed.

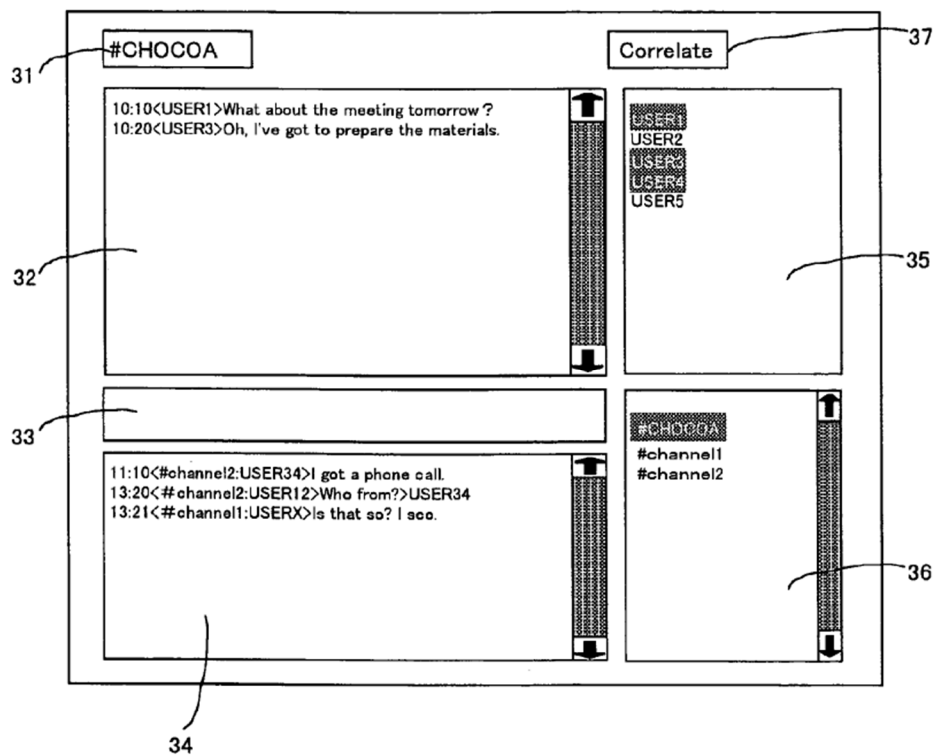
196. In paragraph 256, Dr. Cockburn relies on Matsumoto’s “Correlate” button. However, the “Correlate” button is used to activate “sending conditions” previously stored in a “table” to correlate certain messages for forwarding to a destination channel, and fails to meet the claim language.

B. No Motivation to Combine

197. A Skilled Artisan would not have combined Kakuta and Matsumoto in the manner claimed because Kakuta is concerned with simplifying the display of conversations in a single channel, and provides a solution by which individual conversations may be separately displayed in their own thread window, whereas Matsumoto is concerned with filtering messages across multiple channels in a messaging system. A Skilled Artisan would not find the teachings of Matsumoto relevant to the system of Kakuta and would not look to Matsumoto for possible improvements to Kakuta.

198. Matsumoto builds upon a traditional IRC system to allow users to participate in multiple channels at once and to “correlate” channels such that postings to one channel may be duplicated to other channels based on settable

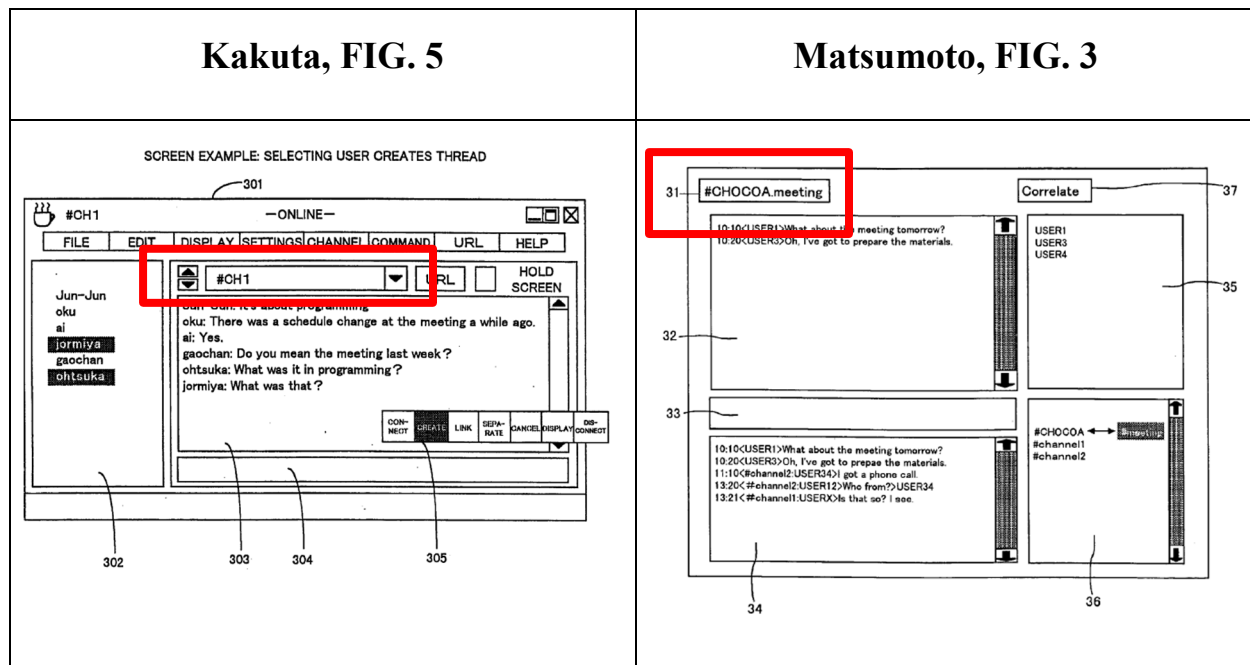
criteria. Matsumoto provides a “main chat window 32” that contains messages posted to the channel selected as the main channel. Also provided is a “sub-chat window 34” that will show postings to all additional channels a user chooses to participate in. Ex-1006, 9:26-48, 10:40-53. Matsumoto is clear that the term “sub-channels” does not refer to threads within the main channel, but instead to other channels that are being displayed in the “sub-chat window”—a window that intermingles messages from multiple channels at once. Ex-1006, 9:40-42. Figure 3 of Matsumoto, on which Petitioner relies, depicts a main channel window displaying messages posted to the “#CHOCOA” channel, and a sub-channel window displaying messages posted to either #channel1 or #channel2. Ex-1006, 9:27-10:15.



199. The title of the main channel is provided in the “main channel window 31” above the “main chat window 32”. The names of all the channels displayed on the screen (in either main chat window 32 and sub-chat window 34) are provided in the “channel window 36.” Ex-1006, 9:35-48. The provision of “channel window 36” is unique to—and necessitated by—Matsumoto’s unique “sub-chat window 34” which intermingles messages from multiple channels.

200. If a Skilled Artisan were to look at Matsumoto, he or she would readily have seen that Matsumoto provides a single channel identifier for its main chat window 32—just like Kakuta. Thus, a Skilled Artisan would not have been motivated to make any change to Kakuta’s channel field based on a review of

Matsumoto.



201. Moreover, a Skilled Artisan would not have modified the channel identifier of Kakuta to list all (or a large subset) available channels because doing so would take up valuable screen real estate without any identified benefit. And nothing in Matsumoto suggests doing so given that “channel window 36” does not list all available channels, but only those channels being viewed on the screen. Moreover, the dominant focus of Kakuta’s user interface is on a single channel, and threads within a channel. Thus, the conventional IRC channel selection screen already met Kakuta’s needs.

XI. GROUND 4: CHANG DOES NOT ADDRESS THE DEFICIENCIES OF KAKUTA OR MATSUMOTO

A. Claims 1, 15, and 25

1. Chang Does Not Disclose a “Second User-Selectable Control”

202. The claimed second user-selectable control must have four characteristics: (a) it must be associated with a first messaging communication (1[o], 15[n], 25[n]); (b) it must be in the second pane (1[o], 15[n], 25[n]); (c) it must be associated with an indication that said first messaging communication is an anchor message of a subsidiary communication thread (1[o], 15[n], 25[n]); and (d) selecting it must cause at least a portion of the first subsidiary communication thread to be displayed in the third pane (1[p], 15[o], 25[o]). An exemplary control is shown here:



Ali T. 8:42 am

You should see if Kasey is available to help with some copy. She's awesome.

Sidebar with Ali started May 10 at 8:43 am

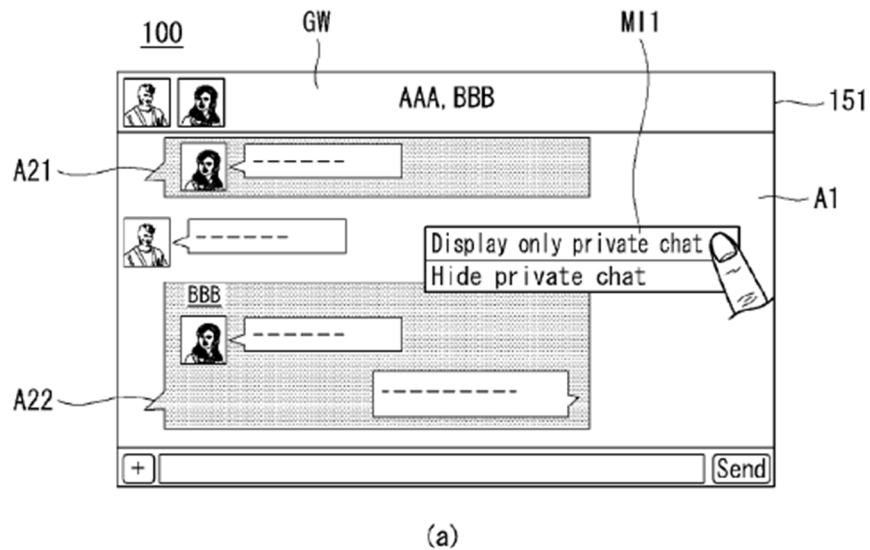
203. Petitioner relies on Chang's description of a mechanism to allow a user to “display only private chat” messages or to “hide private chat” messages, which is accomplished by touching private chat areas for a predetermined time period, which causes display of a pop-up menu with the aforementioned options. Ex-1003, ¶¶267-268; Chang, [0270]. However, this mechanism is not associated with any particular messages. The private chat area may be touched anywhere to

trigger the pop-up menu. In addition, the Chang menu bears little resemblance to the claimed “second user-selectable control” as it controls whether certain areas are visible or hidden and it provides no indication that a first messaging communication is an anchor message.

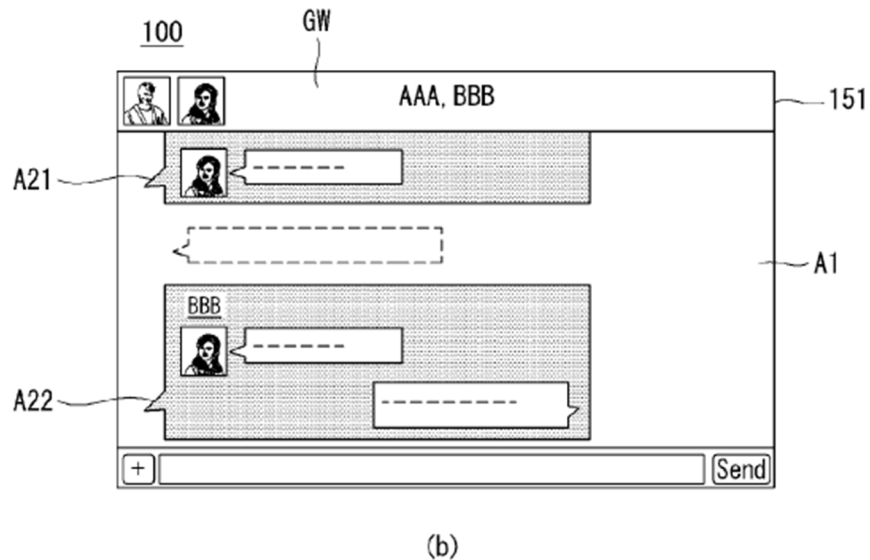
204. Petitioner asserts that the visible message in private chat area A21 indicates that it is an “anchor message,” but the Petition does not explain why or how this is so. Dr. Cockburn suggests the alleged “anchor message” status is shown merely by the message “being enclosed within the private chat area A21.” Ex-1003, ¶268. That is not correct. Being enclosed by a “private chat area” just signifies that the message is private—and not that it is an anchor message for a thread. In addition, Petitioner assumes that there is only one message in the top chat area A21, but there may, and likely would be, multiple such messages in that private chat area (even though only one message is visible). In Chang, private messages are shown in shaded areas and Chang discloses that more than one message may be in a single shaded area. Figure 29 of Chang, for example, discloses a top message that may be part of a private conversation, where the prior private messages scrolled up and off the screen. Thus, while the message at the top of Figure 29 of Chang is the last message in the illustrated private area, but it may not be the first. There is no indication in Chang as to how many messages came before the message visible at the top of the screen and whether the immediately

preceding message or any preceding message was also a private message between AAA and BBB. Thus, Chang provides no indication that the top message is or could be an anchor message.

FIG. 29



(a)



(b)

205. The control on which Petitioner relies is a touch of the private area.

But the private area corresponds to the thread window of Kakuta. Thus, a Skilled Artisan would not have been led by Chang's touch control in Chang's private area to modify Kakuta's controls. One reason is that Chang's control is in the wrong place. Kakuta's "Display" button is used when a user wants to display a thread and as depicted in Figure 12, the thread selection window is present at a time when the window for the desired thread is closed. Thus, the "Display" control cannot, logically, be placed in the thread window, since it must exist before the thread window is displayed. By contrast, Chang's private area control is part of the private area. The private area must be displayed before the control can be selected. Thus, a Skilled Artisan considering Kakuta and its "Display" button would not be motivated to modify the operation of the "Display" button based on Chang's private area control, because Chang's private area control could not perform the functions required of Kakuta's Display button

206. Also, like Kakuta, Chang teaches a multi-step process in which a user must first touch the private chat area, and then perform a further step (selecting from the pop-up menu) to accomplish the desired function (hiding the private chat or hiding the non-private chat messages). Therefore, a combination of Chang and Kakuta would not transform Kakuta's mechanism for displaying a chat window into the claimed second user-selectable control, which provides for a single step to select and display in the third pane a thread associated with an associated

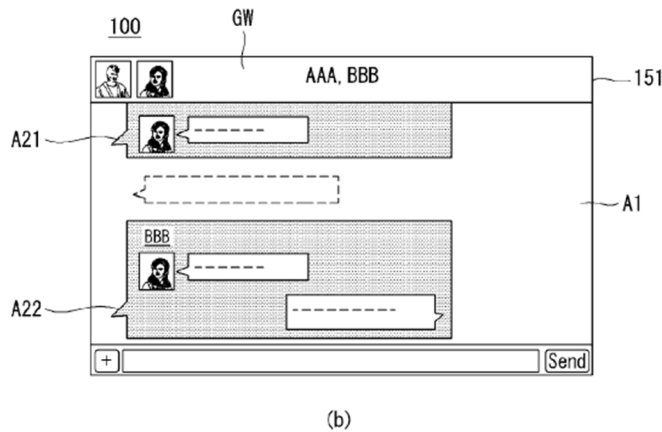
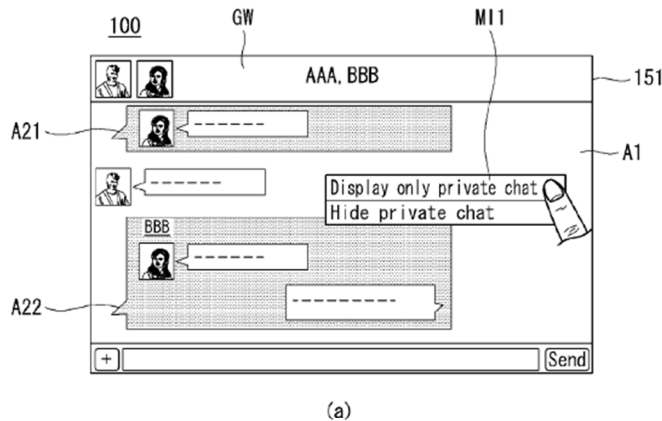
additional messaging communication.

207. Finally, a Skilled Artisan would have recognized that Chang and Kakuta are substantially different in that Chang lacks a separate window or anything corresponding to a “third pane.” Regardless of the functionality of Chang’s control, the private chat messages are not displayed in a third pane, but rather are displayed in the same pane as other messages. Chang and Kakuta are sufficiently different that a Skilled Artisan would not borrow ideas from Chang to fill gaps in Kakuta. It is apparent that the combination is rooted in hindsight—starting with the claims, identifying (some of the) gaps in Kakuta, and then seeking out distant references in an attempt to fill the gap. A Skilled Artisan starting with Kakuta would not seek out Chang and would not see value in the touch control in Chang’s private chat area. Notably, the proposed modification—adding thread names to Chang’s selection window to hide various areas—is not suggested by the prior art. A Skilled Artisan would not have combined such disparate functions into a single selection menu, as confirmed by reviewing the selection menus in both Kakuta and Chang, each of which provide selection menus with like options.

2. The Proposed Combination Would Not Result in The Claimed Invention

208. Chang teaches a private chat area where one-on-one private message may be displayed. Chang teaches that a press of a certain duration in the private chat area causes a selection window to appear:

FIG. 29



Ex-1007, FIG. 29. Even assuming a Skilled Artisan would have been motivated to combine Chang, FIG. 29 with Kakuta (and the Petition fails to establish that), the Skilled Artisan would follow the teachings of Chang and trigger the selection window off a press of predetermined length in Kakuta's **private chat area**—namely, Kakuta's thread window.

209. Moreover, Petitioner overlooks **which** subsidiary communication thread must be displayed. The displayed thread in the claim must be the thread with the anchor message, but under the Petition's mapping the thread with the

anchor message is already displayed and the hypothetical selection window would be allegedly used to select a **different** thread. Petition, 108 (allowing selection of other threads unrelated to the alleged anchor message).

210. Moreover, the proposed combination based on the teachings of Chang and Kakuta would not result in the hypothetical system depicted on page 108 of the Petition. The pop-up window in Chang provides the user with two options: Display only the private chat areas (hiding the public chat area) or hide the private chat areas (and only display the public chat area). Petitioner's hypothetical of allowing a user to select one thread for display in the private chat area is nowhere suggested in Chang, is significantly different from the operation of Chang, and is incompatible with Chang. For example, the pop-up menu in Chang is directly related to the area shown on the screen, whereas a thread selection menu would not be. I understand that when references are combined, they are not bodily incorporated, but rather the teachings of the references are considered. The teaching on which Dr. Cockburn relies is the touch control in Chang's private chat area that triggers a menu allowing either the private chat area or the group chat area to be hidden. Ex-1003, ¶268. Such a control does not translate to the Kakuta system, and thus the Chang teaching is incompatible with Kakuta.

211. Dr. Cockburn asserts that the teaching from Chang would allow for placement of a control "into the second pane of Kakuta [which] would enable the

invocation of a subsidiary communication thread in the third pane.” Ex-1003, ¶269. However, Chang’s control does not allow for such functionality. Chang’s control operates in a pane to show or hide portions of that pane. Chang provides no teaching of a control in one pane influencing the behavior of another pane. And Kakuta’s system would not use controls in a pane that show or hide that pane. In Kakuta, the controls are specifically placed in a floating toolbar independent of any window, which Petitioner maps onto the various panes.

212. Dr. Cockburn further asserts in paragraph 269 that the combination would have a “selectable menu” that would “include options for selecting each of the threads associated with each message.” Dr. Cockburn applies hindsight to reach this conclusion. Nowhere in Kakuta or Chang is there a selectable menu associated with individual messages—that is a concept in the ’731 patent and not any of the three prior art references. Nor do the prior art references suggest the ability to select threads associated with specific messages.

B. Claim 4 – The Combination Fails Because Petitioner Misstates the Teachings of Chang

213. Dr. Cockburn asserts that Chang’s Figure 29 discloses that a user may “open a private subsidiary chat window from within a group chat interface.” Ex-1003, ¶272. That is incorrect. In Chang, the control permits only two options: “display only private chat” which is already displayed, or “hide private chat.” Neither option entails opening a private subsidiary chat window. Further, the

control in Figure 29 of Chang is not associated with any particular message. The selectable menu is not triggered by, or associated with, any particular message. The position of the selectable menu is not associated with any particular message. And, the options in the selectable menu (“Display only private chat” and “Hide private chat”) are not associated with any particular message. Accordingly, Chang does not provide a reason or motivation to place controls proximate to a specific message in the main chat window (the alleged second pane) of Kakuta. Chang specifically has a group chat area (not window, but area) and yet Chang does not teach placing the control on which Petitioner relies in the group chat area. Moreover, Chang teaches the touching of anywhere in the private chat area, and thus the location of the messages are irrelevant. Therefore, a Skilled Artisan would not have a reason to place controls proximate any particular message based on the teachings of Chang. Dr. Cockburn asserts (in paragraph 274) that Chang has “proximity-based user controls,” but as I explained, that is incorrect. Chang’s control in Figure 29 is an area control unrelated to any particular message and is thus not proximate any particular message. Moreover, the resulting menu shown in Figure 29 has two menu options, neither of which are associated with any particular message.

C. Claims 8 & 20 – The Combination Fails Because Petitioner Misstates the Teachings of Chang

214. As with claim 1, Petitioner again asserts that “Chang teaches that

when each message, such as a message among private chat areas A21” is touched, the pop-up menu appears. Petitioner is incorrect. Chang teaches that when any portion of the shaded private chat areas A21 or A22 is touched, the pop-up menu appears. Ex-1007, [0270]-[0271]. Moreover, Petitioner does not actually map the proposed combination onto claims 8 or 20. Claims 8 and 20 require the display of “said additional messaging communication” in the second pane and displaying an “indicator identifying said additional messaging communication as associated with a subsidiary communication thread.” Nowhere does Petitioner or Dr. Cockburn explain how the combination achieves the claimed subject matter. Petition, 111-113, Ex-1003, ¶¶276-280. From Dr. Cockburn’s incorrect assumption about Chang, Dr. Cockburn asserts that a combination of Chang and Kakuta would result in a selectable menu specific to a message, which would display multiple threads that are associated with that message. However, Chang does not teach a selectable menu that is associated with any specific message, but rather a selectable menu triggered by, and associated generally with, areas independent of any message, and the menu items provide options that are likewise unrelated to any specific message, but rather act on the whole screen.

D. Claim 12 – The Combination Would Not Operate as Claimed

215. Petitioner asserts the “Kakuta and Matsumoto device” would be modified to add “Chang’s feature to add a non-member to a communication

thread” to “enable said non-member user to receive such communications.”

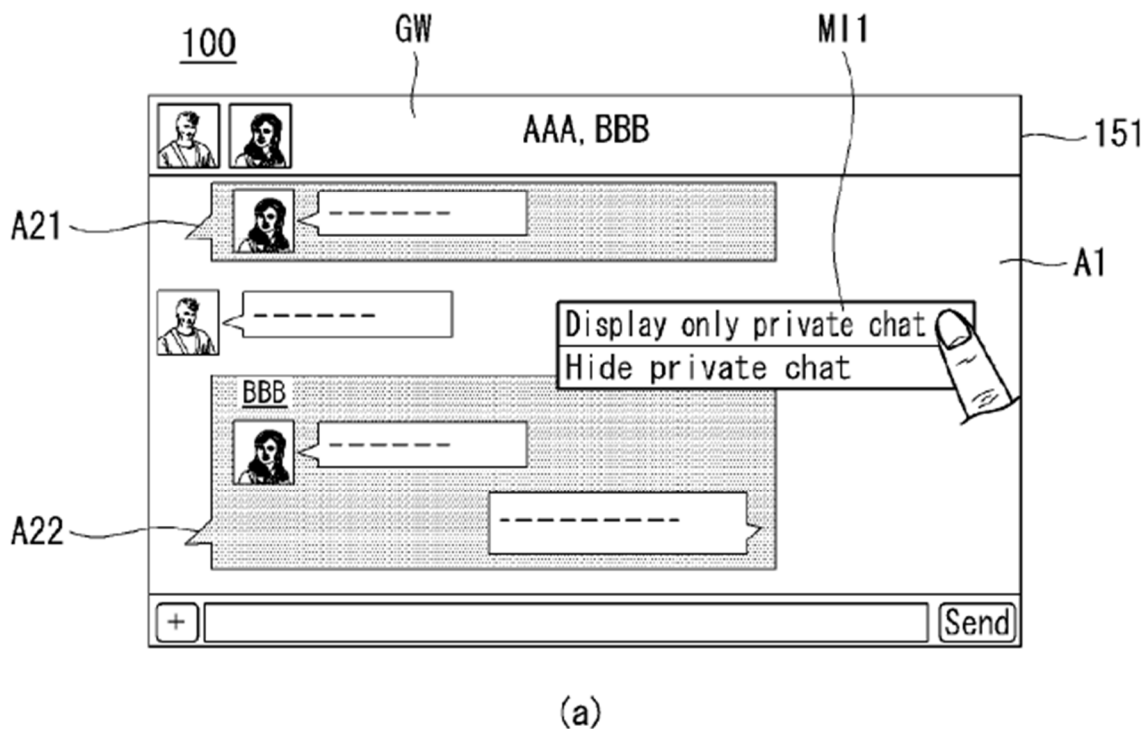
However, the Dr. Cockburn does not explain what the combination looks like or how it would operate, and does not explain how the combination ensures “controls are logically and spatially related to the communications they affect.” Ex-1003, ¶284. A Skilled Artisan starting with Kakuta would have no reason to combine teachings of Matsumoto and Chang to result in the claimed invention. Chang does not disclose inviting a “non-member” of an “associated group of members of [a] group-based communication channel” to receive messaging communications comprising a “subsidiary communications thread.”

216. Unlike Kakuta and Matsumoto, Chang has no concept of a communication channel having a group of associated members. Chang is not a channel-based communication system, but instead is a conventional text messaging application for a mobile device without channels or channel members. Thus, Chang does not teach adding “non-members” because Chang does not have a concept of channel membership. Petitioner relies on a disclosure of one party inviting “another party” to a chat. Ex-1007, [0276]-[0285]. Nothing in such teaching would motivate a Skilled Artisan to modify Kakuta or Matsumoto to include a concept of inviting non-members of a channel to receive messaging communications comprising a “subsidiary communications thread.”

XII. GROUND 5: CHANG DOES NOT FILL THE GAPS IN KAKUTA FOR CLAIM 30

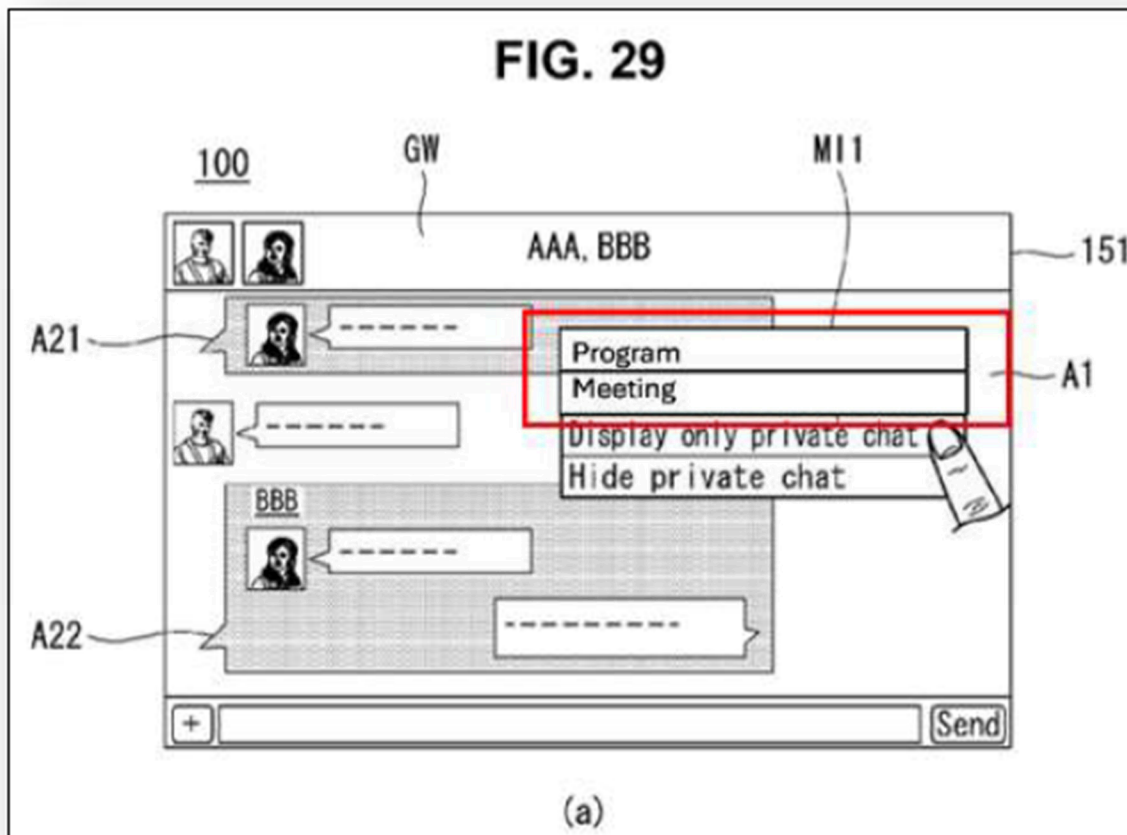
217. Petitioner and Dr. Cockburn assert that it would have been obvious to combine Kakuta with Chang's selectable menu, "thereby incorporating an indicator associated with a message." Petition, 115, Ex-1003, ¶289. But, Chang does not teach that the pop-up menu is associated with any "message." Rather, Chang teaches that the same pop-up menu appears when any of private chat areas A21 or A22 (the entire shaded regions) is touched:

FIG. 29



218. Dr. Cockburn is also incorrect to assert that Chang's modified pop-up menu would indicate that a message "is associated with a subsidiary

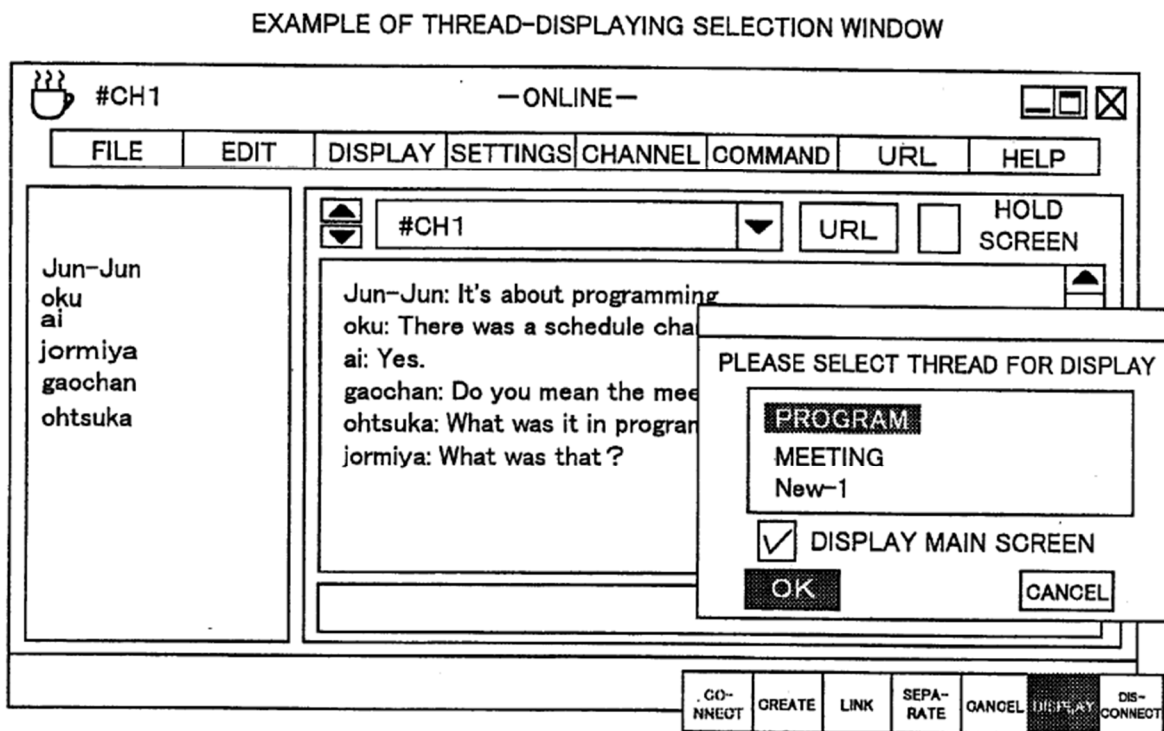
communication thread,” such as “Program” or “Meeting.” Ex-1003, ¶279. To the contrary, nothing in Chang’s menu associates the menu options to any particular message, and thus in the combination, nothing in the hypothetical menu (shown below) would be associated with any particular message.



Ex-1003, ¶279. Based on the teachings of Kakuta and Chang, the hypothetical menu would list all available threads, as taught by Kakuta. Petitioner’s contrary argument that only those threads associated with the nearest message would be listed deviate from the teachings of the references, Kakuta’s “Display” button

displays all threads accessible to a user and Chang's pop-up menu is likewise independent of any particular message).

Fig. 12



Ex-1005, Fig. 12.

XIII. PETITIONER'S MOTIVATION SECTION (SECTION VII) APPLIES IMPROPER HINDSIGHT

219. Kakuta specifically seeks to make a “conversation display easy to view” within an IRC or similar environment. Ex-1005, 1:6-23. Kakuta displays a single channel at a time (as was conventional) and makes conversations within that channel easier to view by creating threads. “The object” of Kakuta is to “display topics respectively segregated from other topics.” Ex-1005, 1:60-65. Matsumoto

addresses a fundamentally different problem and heads in a different direction.

Although Matsumoto recognizes that multiple conversations may be present in a single channel, Matsumoto is concerned with conversations that span multiple channels, which it views as problematic because “in IRC at present, messaging is always limited to a single (current) channel.” Ex-1006, 1:47-2:15.

220. Rather than carve channels into threads, Matsumoto copies messages from one channel into another based on text filtering, thus intermingling messages from multiple channels. Ex-1006, 2:44-58. Matsumoto proposes displaying multiple channels at once and provides a filter to merge conversations on multiple channels into a single “sub-channel” window. Ex-1006, 9:25-48. There is no reason a Skilled Artisan seeking to improve Kakuta’s thread display would look to Matsumoto’s multi-channel display for guidance. Kakuta provides an improvement to traditional IRC in the form of a new thread window. Kakuta remains faithful to the traditional IRC’s single channel approach. Matsumoto is silent as to how the channel is selected for the main chat screen and thus a Skilled Artisan would assume it likewise operates in the traditional way.

221. Chang’s focus is even more distant from Kakuta. Chang is concerned with incorporating private chats into a mobile text messaging app with a single window to avoid the inconvenience of moving between windows on a mobile app. Ex-1007, [0002]-[0008]. This is not a concern of Kakuta, which does

not mention “mobile” and depicts ordinary computer screens.

222. Dr. Cockburn asserts that “[a] PHOSITA would have looked to Matsumoto for its specific teachings relating to the display of multiple channels.” Ex-1003, ¶295. This is hindsight based on the ’731 patent. A Skilled Artisan presented with Kakuta would implement Kakuta in the traditional manner, in which, as discussed, a user selects a channel from a channel selection screen and then a chat window is presented for the selected channel. Each channel would have a different topic and therefore a Skilled Artisan would not consider combining channels. Thus, a Skilled Artisan would not turn to Matsumoto for possible improvements, and would not adopt Matsumoto’s complex system, which heads in a direction opposite that of Kakuta. Kakuta seeks to simplify the display of a single channel and thus a Skilled Artisan would not seek to complicate matters by displaying multiple channels.

223. Dr. Cockburn asserts that “integrating Matsumoto’s filter-destination channel into the functionality provided by Kakuta would be a simple and routine matter.” Ex-1003, ¶295. However, Kakuta and Matsumoto are substantially different systems with different functionality. Thus, integration of the two systems may pose a challenge to a Skilled Artisan. One of the challenges would be to determine exactly how the integration should occur—considering which features of Matsumoto should be brought into Kakuta, and considering how those features

would be implemented in view of the teachings of both references. Petitioner does not explain how the integration would proceed or what an integrated product would look like and it is not clear that such an integration is possible without “breaking” the functionality of either Kakuta or Matsumoto. For example, Matsumoto allows users to forward messages from one channel to another. Such functionality has no connection to Kakuta, which operates on a single channel at a time. Thus, a Skilled Artisan would not seek to combine Kakuta with channel forwarding, and therefore a Skilled Artisan would not understand how such “integration” should proceed. And, even if one were to add “channel forwarding” to Kakuta’s system, the result would not be the claimed invention. Channel forwarding functionality, for example, could be provided in a system through a separate screen—one that would not be integrated into the screens disclosed by Kakuta. However, such a system would still not provide the claimed first pane. For example, the system would have (1) the chat screens shown in Kakuta, (2) a separate channel selection screen, as was known in the IRC art, and (3) a separate channel forwarding screen providing channel forwarding in the background.

224. Petitioner and Dr. Cockburn rely on hindsight for its motivation to combine Chang (a motivation that appears relevant to claim 12). Dr. Cockburn states: “A PHOSITA would also have been motivated to look to Chang for its specific teachings regarding the adding of parties to group chats.” Ex-1003, ¶297.

I disagree. A Skilled Artisan working with Kakuta would have no desire to add non-channel members to Kakuta's threads as there was no perceived problem in this regard. Nor does Kakuta suggest that this is desirable. The statement in Kakuta on which Petitioner relies merely states that information regarding the "joining/departing of channel participating users" is provided to each client using "ordinary IRC routines." Ex-1005, 11:4-12.

225. In ordinary IRC, a user selects a channel to join. Chang provides no insight into this ordinary and common function of IRC systems. Figures 31-33 of Chang concern different functionality, and specifically a user interface that allows one user to add other users to a channel or private chat. Kakuta does not contemplate one user adding another user to a channel and Kakuta already discloses how one user adds another user to a thread (such as by selecting users as in Fig. 5 of Kakuta). Thus, no Skilled Artisan would seek out Chang to fill gaps in the teachings of Kakuta.

XIV. CONCLUSION

238. For the reasons stated in this Declaration, it is my expert opinion that the challenged claims would not have been anticipated or obvious to a Skilled Artisan.

239. This declaration and my opinions are made to the best of my knowledge and understanding and based on the material available to me at the time

of signing this declaration. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001. I declare under penalty of perjury that the foregoing is true and correct.

Executed this 21st day of August, 2025 in Santa Barbara, CA.

Kevin C Ameroth