

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner

v.

X1 DISCOVERY, INC.,
Patent Owner.

IPR2025-00253 (Patent 7,370,035 B2)
IPR2025-00254 (Patent 10,552,490 B2)
IPR2025-00255 (Patent 9,633,139 B2)

Record of Oral Hearing
Held: April 27, 2026

Before WILLIAM V. SAINDON, SCOTT C. MOORE, and LISA A.
MURRAY, *Administrative Patent Judges*.

IPR2025-00253 (Patent 7,370,035 B2)
IPR2025-00254 (Patent 10,552,490 B2)
IPR2025-00255 (Patent 9,633,139 B2)

APPEARANCES:

ON BEHALF OF THE PETITIONER:

SAMUEL THACKER, ESQ.
TODD SIEGEL, ESQ.
Klarquist Sparkman, LLP
One World Trade Center
121 SW Salmon Street
Suite 1600
Portland, Oregon 97204
(503) 595-5300 (Thacker)
(503) 473-0918 (Siegel)
samuel.thacker@klarquist.com
todd.siegel@klarquist.com

ON BEHALF OF THE PATENT OWNER:

CHRISTIE LAROCHELLE, ESQ.
EMMA MURRAY, ESQ.
LORI GORDON, ESQ.
NATASHA DAUGHTREY, ESQ.
Goodwin Procter LLP
100 Northern Avenue
Boston, Massachusetts 02210
(617) 570-1683 (Larochelle)
clarochelle@goodwinlaw.com

The above-entitled matter came on for hearing Monday, April 27, 2026, commencing at 1:00 p.m. EDT, at the U.S. Patent and Trademark Office, 600 Dulany Street, Alexandria, Virginia, and via videoconference.

P-R-O-C-E-E-D-I-N-G-S

1:00 p.m.

JUDGE SAINDON: Please be seated. Good afternoon.

I am Judge Saindon. With me today are Judges Murray and Moore. Judge Moore will be joining us remotely. So as a reminder, please refer to slide numbers whenever you are discussing slide numbers and to citations to the record as appropriate. We have all the exhibits and papers in front of us.

And, let's see. Okay. My computer has loaded. So, welcome. This is a hearing for IPR 2025-00253 and 254 and 255. We have here 90 minutes for Petitioner. And because we have a LEAP practitioner, 105 minutes for the Patent Owner.

Before we get started, let's do introductions for the Counsel. Patent Owner, why don't you go first?

MS. LAROCHELLE: Sure, good afternoon. Christie Larochelle from Goodwin Procter on behalf of Patent Owner X1. And with me is our LEAP practitioner, Emma Murray, also from Goodwin Procter, and Lead Counsel Lori Gordon, Natasha Daughtrey also from Goodwin Procter, and then we have a client representative, John Patzakis from X1.

JUDGE SAINDON: Okay, welcome. And Petitioner?

MR. THACKER: For Petitioner, Sam Thacker from Klarquist Sparkman. And with me is my colleague Todd Siegel, also from Klarquist.

JUDGE SAINDON: All right. Welcome, and you may take the podium.

1 So we have 90 minutes on the clock. You can divide that
2 as you see fit between your times, with a few caveats. But if
3 you want to reserve a certain amount of time, I can take care of
4 that now, or you can just work with your pool of time.

5 MR. THACKER: Yes. Can I reserve 40 minutes for
6 rebuttal? Thank you.

7 JUDGE SAINDON: And you may begin when ready.

8 MR. THACKER: All right. May it please the Board,
9 Sam Thacker on behalf of Petitioner Microsoft. The challenged
10 claims in these three IPRs recite a combination of well-known
11 software concepts that had established benefits before the
12 claims' alleged priority date.

13 I invite the Board to Slide 3, which shows the
14 obviousness grounds raised by Petitioner. Petitioner raises two
15 sets of obviousness grounds, each of which combines teachings
16 from references describing commercial systems, Lotus Notes or
17 Entourage, with other references describing well-known
18 features.

19 Now, we turn to Slide 4. There's no dispute as to the last
20 feature here, the intersection of results. So the features in
21 dispute are performing an incremental search across multiple
22 files for multiple search strings entered into a single search
23 field, and for only the '490 Patent, the 254 IPR, using an index.

24 Now, if we turn to Slide 6, Patent Owner does not
25 dispute, and this is the Patent Owner Response shown on Slide

1 6 here. Patent Owner does not dispute that these features were
2 known.

3 And as shown on Slide 7, it's undisputed that these
4 features had known benefits. The primary dispute in these IPRs
5 is whether a skilled artisan would have been motivated with a
6 reasonable expectation of success to combine these prior art
7 teachings. Unless the Board would prefer otherwise, I'd like to
8 start with Ground 2, since there appears to be fewer issues with
9 that ground. I'll, of course, turn to Ground 1 after, but I want to
10 make sure that we cover Ground 2. Okay.

11 So I'd like to invite the Board to Slide 43, which shows
12 the Ground 2 combination. And this combination includes
13 references describing the Entourage commercial system, True,
14 which is an issued patent, and Baeza-Yates, which is an
15 undergraduate-level textbook. Specifically, Ground 2 combines
16 Entourage's teachings to incrementally search across files with
17 True's teachings to incrementally search for multiple search
18 strings entered in a single search field and Baeza-Yates'
19 teachings to use an index to speed up the search process.

20 Turning to Slide 45, Entourage teaches incrementally
21 searching across files, including, as shown here on Slide 45,
22 contacts in an address book. And then it's shown on Slide 46,
23 emails and an inbox. It also discloses incrementally searching
24 across tasks in a calendar. And as shown on Slide 46,
25 Entourage confirms that this search is conducted as you type.

1 So this is an incremental search, across what's shown here in
2 46, is emails.

3 Now turning to Slide 47, the motivation to combine
4 Entourage with True and Baeza-Yates on a first start with the
5 motivation to combine Entourage with True, starting on Slide
6 48. And a POSITA or skilled artisan would have been
7 motivated to combine Entourage's teachings to incrementally
8 search across files with True's teachings to incrementally search
9 for multiple search strings entered in a single search field,
10 because multiple search strings provides more targeted results
11 to the user. This is explained on pages 66 to 69 of the petition.
12 True teaches incrementally searching multiple search strings
13 and teaches that the transcript windows that contain both search
14 strings are given a higher score and thus are highlighted to the
15 user because they're more likely to be relevant.

16 Now, this concept of providing more targeted search
17 results to a user via a multi-string query was known in the art,
18 as shown on Slide 49. Patent Owner does not dispute that there
19 is this known benefit of allowing a user to enter multiple search
20 strings.

21 JUDGE SAINDON: Counsel --

22 JUDGE MOORE: Counsel.

23 JUDGE SAINDON: Go ahead.

24 JUDGE MOORE: Just a question about True.

25 MR. THACKER: Yes.

1 JUDGE MOORE: I think the example you cited in True
2 was the search for the term "Queen Mary." How do I know that
3 True is treating that as two independent search strings and not a
4 single search string with a space as one of the characters?

5 MR. THACKER: Yes, so, this is shown on the petition
6 at page 69, which is citing True. And this is True on column 5,
7 line 41 to 50, cited by the petition at page 69. And it describes
8 how each word, "Queen" or "Mary", or "Queen" or "Mother,"
9 excuse me, has its own score. And so, the total relevancy score
10 for a given transcript window is computed as the sum of a score
11 for each individual word. So it's treating these words
12 separately.

13 JUDGE MOORE: Thank you.

14 JUDGE SAINDON: Counsel, while we have you, can
15 we jump back a little bit and talk about the Entourage
16 references themselves?

17 You showed Slide 45, which is the address book filter,
18 and Slide 46, which depicts the email search box. So the claims
19 are a method, and they're performing various tasks in a search
20 field. But these are two different search fields. So how are we
21 putting this together?

22 MR. THACKER: Can you repeat the question? I missed
23 what the separate search fields are in Entourage.

24 JUDGE SAINDON: Right. So in Entourage, Figure 46
25 has the filter on the inbox.

26 MR. THACKER: Yep.

1 JUDGE SAINDON: And it says, I think, chapter is what
2 is being searched for there. And then Slide 45, we have the
3 address book filter. And it says, name contains Schwartz. And
4 so we're looking there. So those are different boxes, right?
5 We're not typing in the same box. Is there something being
6 combined from the address book feature of Entourage and the
7 Lotus, or not the Lotus, the email portion? What are we doing
8 with these different figures?

9 MR. THACKER: Yeah, so, what these figures are
10 showing is the same concept, which is incrementally
11 performing a search using a single search field. And what these
12 slides and figures are showing is that Entourage has separate
13 windows for searching a specific file type, right? So if
14 it's -- you can search address or contact information
15 specifically. And then, much like the patent's different tabs for
16 different file types, there is a separate window for performing
17 incremental search across emails.

18 JUDGE SAINDON: Okay. So, if we wanted to say
19 we're taking one of these views, figures, and we just pick one,
20 are we going to have everything that you're pointing out as
21 necessary from the claims' standpoint in that one form or view
22 or whatever we want to call these particular figures in
23 Entourage, or do we need to mix and match features from
24 multiple figures in Entourage?

1 MR. THACKER: It's not necessary to mix and match.
2 One of them alone is sufficient to satisfy the independent claim
3 elements.

4 JUDGE SAINDON: Okay, thank you.

5 MR. THACKER: If there are no further questions on the
6 motivation to combine Entourage with True, which I think is
7 what I was covering last, I would like to turn to Slide 52,
8 which --

9 JUDGE MOORE: Counselor, sorry. If you're going to
10 move on, I do have another question about that subject before
11 you move on to the next one.

12 MR. THACKER: Yeah, for sure.

13 JUDGE MOORE: Okay, so your basic argument is that a
14 person would have looked at, you know, the incremental search
15 and the ability to type two search terms into the window from
16 True and combine that with the Entourage database to create
17 something like that there. So just stepping back a minute and
18 thinking about what a POSITA would have realized just
19 looking at the references themselves, you know, what are the
20 benefits of an incremental search?

21 You know, I can see it in the Entourage references, but
22 we're talking about entering two terms in a single field. What
23 are the benefits of a search in that context?

24 MR. THACKER: So, sorry, I missed the last part. Are
25 you saying what's the benefits of an incremental search in the
26 context of multiple search strings or just generally?

1 JUDGE MOORE: In the context of two terms entered in
2 a single search field, what we're talking about in True, is it the
3 ability to display, or you know, to display hits to the user so you
4 can kind of get a visual representation of the hits? Is that the
5 advantage that makes it desirable?

6 MR. THACKER: Yeah, so, on -- there's slides in our
7 appendix on this. But in the Ground 1 slides, I would invite the
8 Board to Slides 12 to 14, which those disclose some of the
9 well-known benefits of incremental search, which is, it's more
10 responsive, the user gets results immediately, they're able to tell
11 whether they had a typo or their search strategy isn't working.
12 So, like, they don't have to wait until they enter their full query
13 to start getting feedback. So it wastes less of their time; it's
14 easier to use.

15 Now, for entering multiple search strings specifically, it's
16 also beneficial, and this is shown on Slide 50. And this is
17 Raskin explaining, which is Exhibit 1021, this is Raskin at 127,
18 and it's saying that even building a Boolean search pattern, so a
19 search pattern that has multiple search strings, is made more
20 effective when the early results of the pattern appear. So
21 essentially, there were these known benefits of incremental
22 searching, which is providing early feedback to the user, and
23 those benefits apply in the multi-string context as well.

24 JUDGE MOORE: Okay. I understand that. Raskin is
25 obviously not a reference here. I guess you're arguing this is
26 kind of background knowledge of a POSITA. I'm just trying

1 to -- so look to True for a moment. And True, as I understand
2 it, you're searching a transcript of an audio file.

3 MR. THACKER: Correct.

4 JUDGE MOORE: For reference, breaks -- I forget what
5 it calls them. It calls them windows or whatever. But the True
6 reference breaks that transcript down into blocks, and then it
7 runs the search in blocks. And if there is a, you know, a hit, it
8 determines these confidence intervals, and it will display kind
9 of the matching blocks on the display. So the user gets, you
10 know, feedback. They can see, oh, is this what I was looking
11 for or not. They get the display of the hits, which it can do
12 because it has broken this transcript down into little bits. Is that
13 generally correct or not correct?

14 MR. THACKER: No, that's correct.

15 JUDGE MOORE: What I'm trying to understand is how
16 would a POSITA, staying inside the claims because we don't
17 want to rely on hindsight, how would a POSITA look at that
18 type of functionality that's performing searches within little
19 blocks, you know, blocks of text, a paragraph or so maybe or a
20 couple of sentences within a file, but it's displaying a result,
21 why would a POSITA look at that and think that I could take
22 that same type of query and, you know, feedback mechanism
23 that displays the results and apply it to a database of the type in
24 the Entourage reference, which, you know, contained the
25 thousands, potentially more, emails and other types of contacts?
26 That seems like a little bit of a leap, and I'm trying to

1 understand why you would do that and how it would work.

2 You know, what would the resulting system display?

3 MR. THACKER: Yeah. Well, the skilled artisan would
4 understand that the general teaching to incrementally search for
5 multiple search strings is applicable to Entourage, because for
6 several reasons.

7 First, True is having to have some way of uniquely
8 identifying each of those blocks, much in the same way that
9 when you're searching across files or emails, you have to be
10 able to uniquely identify each email. And Dr. Terveen
11 explained that the general approach is the same. And this was
12 in paragraph 11 of his supplemental declaration, Exhibit 1100,
13 where he explained that this -- even if True were searching
14 within a single file, that same concept of being able to uniquely
15 identify a file or a block and determine whether or not it
16 contains the search strings, is not meaningfully different than
17 searching across files, which where you also have to uniquely
18 identify the file and determine whether or not it contains the
19 search strings.

20 And then as far as the large volume of files in Entourage,
21 a skilled artisan would have understood that on the kinds of
22 database sizes that were typical at the time, there would have
23 been no problem performing an incremental search across those
24 database sizes.

25 JUDGE MURRAY: Counsel, what is your evidence for
26 that?

1 MR. THACKER: Well, first of all, Entourage itself
2 already performs an incremental search across files. So merely
3 extending it to be able to search another search string is not
4 suddenly going to make Entourage not work, right.

5 So if you think about it, Entourage is incrementally
6 searching a single search string. So it's performing that search
7 for a single search string. Let's say you enter the word "cat."
8 As you're typing "cat," it's updating the results. Now, let's say
9 you type space and then you start entering, "dog." All that the
10 incremental search tool has to do is look up those search results
11 for "dog," which is the same process it used to search for "cat."
12 And then as Dr. Terveen explained in his supplemental
13 declaration and initial declaration, then those results are
14 intersected as explained by Baeza-Yates.

15 JUDGE MURRAY: But can you speak to the issue of
16 database size? You said, at the time, the size would not have
17 been an issue. What is your evidence for that?

18 MR. THACKER: So the evidence for that is on Slides
19 58 to 61. As shown on Slide 58, the median email size in the
20 relevant time period was only 2.2 kilobytes. And then, as
21 shown on Slide 59, Dr. Terveen explained that an index for
22 even a 4-megabyte database, which would hold a lot of those
23 2.2-kilobyte emails, was well within the capacity of then-
24 existing computers.

25 JUDGE MOORE: Counselor, I used computers in 2002.
26 I was working in a law firm in Los Angeles, and I'm pretty sure

1 my email box was a lot bigger than 4 megabytes even back
2 then. I mean, am I missing something? Am I wrong, or am I
3 missing something? You're talking about RAM capacity, 120
4 megabytes of RAM. You know, this database is stored on a
5 hard drive, not in RAM.

6 MR. THACKER: Right, and so the index for that
7 database would be pulled from disk or hard drive into memory
8 to then --

9 JUDGE MOORE: Oh, you're talking about the index
10 being in RAM, not the database itself, right. But I think my
11 colleague is highlighting one of the issues that's important to us,
12 and that is, you know, just to not to put too fine a point on it,
13 you've given us references that show incremental searching
14 within a single file or a very small, very small subset of
15 information. And you're saying that a person of ordinary skill
16 in the art would have applied those teachings and that they were
17 well-known, you know, to be very useful and to just apply those
18 to large databases like the Entourage database or the
19 enterprise-level database and the Lotus Notes reference. And,
20 you know, that does seem to be quite a jump. And I think that's
21 what my colleague is struggling with.

22 And, you know, to the extent you can, explain why a
23 person would have made that jump and thought it would have
24 been successful despite the fact, for example, in the Entourage
25 references, you have some types of incremental search, but it's
26 very, very limited to, for example, the filtering context, and

1 they never tried to apply it in that way, right. So that's
2 something that we're struggling with. So to the extent you can
3 spend some time on that, that would probably be helpful.

4 MR. THACKER: I just want to make sure I'm clear on
5 the issues. So you agree, then, that Entourage expressly
6 discloses performing an incremental search across emails. So is
7 the issue whether that is truly a search, or is the issue whether
8 or not you would have allowed the user to enter multiple search
9 strings?

10 JUDGE MOORE: I'm just trying to highlight a larger
11 issue for you to search in. You know, I would say -- you know,
12 I'm not going to tell you what I have concluded or not. But,
13 you know, for example, Entourage does seem to talk about an
14 incremental search in the context of filtering. It doesn't seem to
15 talk about a search across all the fields or of an entire database.
16 For example, I don't see a teaching that you could search your
17 email inbox or email containing a certain search item
18 incrementally. Am I wrong about that?

19 MR. THACKER: Can you say that part again, sorry?

20 JUDGE MOORE: Sure. Let me try and rephrase it. The
21 type of incremental search described in Entourage seems to be
22 very limited. For example, on your Slide 45, you've got a filter,
23 right. You're searching across just the names associated with
24 the emails, you know, recipients or senders. I don't see
25 anything in Entourage that would allow you to perform an
26 incremental search across the entire emails. For example, I

1 don't see functionality in there that would allow you to
2 incrementally search for all emails that contain a certain word,
3 you know, the word "patent" or something like that. I don't -- a
4 functionality where you could type the letter P, and then all of
5 the emails that have the letter P anywhere in the body would
6 pop up. And then you could type A, and then they would go.
7 And then T. I don't see anything like that. It seems to be very,
8 very constrained.

9 MR. THACKER: Well, Entourage --

10 JUDGE MOORE: So --

11 MR. THACKER: Sorry.

12 JUDGE MOORE: Right. And I'm just trying to explain
13 the issue that I am seeing so you can address it. And then in
14 True, it seems to be once again you've got very limited context
15 for the incremental searching. It's performed within small
16 subsets of a single file. So the issue that, I think, it would be
17 very heavily to focus on is why would a person look at these
18 references, and understanding that these searches are
19 processing-intensive, but they take time, and think that you
20 could apply that incremental search, which is only disclosed in
21 Entourage and True as being done in a very limited fashion, and
22 apply it to the entire database, and with a reasonable likelihood
23 of success.

24 MR. THACKER: Yeah. So, looking at Slide 46, for
25 example, Entourage does disclose searching across an entire

1 inbox, the incrementally searching across an entire inbox. So
2 you see in the top --

3 JUDGE MOORE: Slide 46 shows the subject field only.
4 That's a very limited topic.

5 MR. THACKER: Right. But you're still searching
6 across all of the emails, you're just searching within a specific
7 field of those emails, which is in line with some of the
8 dependent claims of these patents.

9 So, one, Entourage does search across the entire inbox.
10 But I want to return to the actual claim language itself, because,
11 which is shown on Slide 5, there is nothing in the claims about
12 searching across a giant database, the whole computer system,
13 the whole inbox, none of that. It is --

14 JUDGE MOORE: I understand that, Counselor, but
15 that's also -- you're getting dangerously close to hindsight when
16 you're looking at the claims to justify your combination, right?
17 And I'm sure you recognize that. So when you say there's
18 nothing in the claims that requires a large database, that's true.
19 However, you've posited it that a person of ordinary skill in the
20 art would have applied these teachings to an Entourage
21 database and a Lotus database. Those are large databases. So
22 your motivation to combine has to take into account the nature
23 of those databases, not whether the claims require databases of
24 those sizes.

25 MR. THACKER: Right. And as we explained in the
26 petition and in the reply, and as shown on Slides 45 and 46,

1 Entourage itself already teaches incrementally searching across
2 all the files in an inbox. And I agree, it's within a certain field,
3 but it is searching across all these emails. And all we're doing
4 is adding True's teachings to allow the user to enter a second
5 search string in Entourage to be able to provide more targeted
6 results, which was, as I discussed earlier, that concept of
7 providing more targeted search results via a multi-string query
8 was well-known in the art.

9 JUDGE MOORE: Okay. So when you were describing
10 that, you weren't talking about, for example, a search within the
11 subject line that would have multiple terms, then provide the
12 results? Because, you know, to be honest, I didn't get that detail
13 clearly from your petition.

14 MR. THACKER: Right. So you would search within a
15 specific header field for multiple search strings. That's correct.

16 JUDGE MOORE: And did your expert or your petition
17 ever clearly spell that wrinkle out?

18 MR. THACKER: I'm not sure if it was expressly --

19 JUDGE MOORE: You don't have to answer that on the
20 spot, if you want to come back in your rebuttal time. But that is
21 something. That's a wrinkle I had not understood until just
22 now. And I'm wondering whether or not that is supported by
23 your petition and your expert. You know, feel free to answer
24 that question in your rebuttal time.

25 MR. THACKER: Will do, thank you.

1 Okay. I would like to turn now to Slide 54, 54 to 56,
2 which is testimony from our expert, Dr. Terveen, explaining the
3 reasonable expectation of success in combining Entourage,
4 True, and Baeza-Yates. And specifically as shown on Slide 55,
5 Dr. Terveen explained that a skilled artisan would have
6 reasonably expected success in separately searching each string,
7 using an inverted index or one of the other well-known data
8 structures that's described by Baeza-Yates, such as tries.

9 And as shown on Slide 56, Dr. Terveen explained that
10 the search tool would use the -- simply look up the set of
11 matches for the first search string, do the same thing for the
12 second search string, and then intersect the results as taught by
13 Baeza-Yates.

14 JUDGE MOORE: Counselor, is there any teaching
15 anywhere that you could cite of this type of search being
16 performed, this type of method being used for an incremental
17 search?

18 MR. THACKER: Yeah. So Dr. Terveen in his
19 supplemental declaration cited, for example, Baeza-Yates
20 Exhibit 1023, Green Exhibit 1018, and Londergan Exhibit 1061
21 as supporting an indexed search for partial search strings. And
22 so what that teaches to a skilled artisan is that they would
23 reasonably expect success in implementing an indexed
24 incremental search, because the capability to search for prefixes
25 was already known. And specifically here for this ground

1 which relies on Baeza-Yates, Baeza-Yates expressly discloses
2 performing prefix searches.

3 JUDGE MOORE: You know, maybe this is -- maybe
4 your index searching is partially the answer to this, but you
5 know, a layperson may wonder, if you're searching for two
6 search terms and you have already entered one of the search
7 terms, and then you start entering the second one, why would I
8 continue searching on the entire database, when I am entering
9 the second term that is searching on the subset that you have
10 already identified in response to the first search term?

11 MR. THACKER: Yeah. Well, in Exhibit 1100,
12 paragraph 47, for example, Dr. Terveen explained that the
13 skilled artisan would understand that if you're typing the word
14 "dog," for example, and the user types the letter G, or D, sorry,
15 types the letter D, the search tool would look for results that
16 have words starting with D. And then, if the user types the
17 letter O, the search tool wouldn't just, like, omit all the results
18 that it just found. It would continue to narrow down the results
19 within those initial batch of results that it had obtained for D.

20 JUDGE MOORE: Right. That's what I am struggling
21 with a bit. Because, please correct me if I am wrong, my
22 understanding, but it seems like your combination. You're
23 saying that then there's two search terms. The system is going
24 to incrementally search as each letter in the first search term is
25 entered. And then it's going to start again from the entire
26 universe when you start entering the second search term and

1 incrementally search. And then each time it's going to compare
2 the results and find the overlap. And I'm wondering why a
3 system would do that? It seems like if you're going to have two
4 search terms and you can do so incrementally, you just narrow
5 the universe each time a separate letter is done, is entered. And
6 if that's what you are doing, then you're never -- you would
7 never be performing the
8 take-two-universes-and-find-the-overlap step.

9 MR. THACKER: Well, as explained by Dr. Terveen in
10 his initial declaration, and I can get you a pin cite for this in my
11 rebuttal, what he explained is that this intersection approach is
12 particularly well-suited for an incremental search because the
13 user can modify either search string. So they're not necessarily
14 just going to enter "cat" and then "dog" sequentially. They
15 might enter C, D, and then add to the first or add to the second.

16 And so what you want to be able to do is dynamically
17 update the results for only the string that's being modified. And
18 so what Dr. Terveen explained is that using this intersection
19 approach, you can maintain the search results for each string
20 that has been entered, and then as the user modifies or adds to
21 one of those strings, the search results for only that search
22 string are narrowed. And then the intersection is reperformed.
23 But the search results for the other search string that hasn't been
24 modified, those stay. The tool wouldn't just disregard those.
25 And so Dr. Terveen explained that when a user modifies one of

1 the two search strings, only that search on that particular search
2 string is being narrowed.

3 JUDGE MOORE: Okay. Thank you.

4 MR. THACKER: If there are no further questions on
5 Ground 2, I would like to turn to Ground 1.

6 Okay. So turning to Slide 10, the Ground 1 combination
7 includes references describing the Lotus Notes system and
8 Raskin, which is a textbook on user interface design.
9 Specifically, Ground 1 combines Lotus Notes' teachings to
10 search across multiple indexed files for multiple search strings
11 entered in a single search field. And the Ground 1 combines
12 these teachings with Raskin's teachings to perform a search
13 incrementally. So here, Lotus Notes discloses every disputed
14 claim feature, except for this incremental search feature, which
15 we're relying on Raskin for.

16 So turning to Slide 12 and the motivation to combine
17 these teachings from Lotus Notes and Raskin, a skilled artisan
18 would have been motivated to incorporate Raskin's incremental
19 search teachings for the reasons stated by Raskin. For example,
20 as shown on Slide 12, Raskin explains the problems with
21 traditional delimited searches like Lotus Notes. Specifically,
22 these traditional delimited searches are punishing to the user
23 because, as we discussed earlier, the user has to wait until they
24 enter their whole query and then hit the enter key before they
25 receive results. So they're not going to know immediately
26 whether they made a typo, whether their search strategy is

1 weak. And so the delimited search wastes a lot of the user's
2 time.

3 And that's why, as shown on Slides 13 and 14, Raskin
4 explains that incremental search is much better, because it
5 provides constant feedback to the user as the user types. The
6 user doesn't have to wait until they enter their whole search
7 string to get results. So the user can tell immediately if they
8 have made a typo; and inversely, if they've already reached
9 their desired target, they can stop typing. They don't have to
10 type in the entire search string.

11 And so, as shown on Slide 14, Raskin explains that the
12 advantages of incremental search are so numerous that Raskin
13 himself can see no occasion in which a delimited search would
14 be preferred.

15 JUDGE MURRAY: Counsel, could we go back to Slide
16 10 for a moment. It refers to the Lotus Notes, but that's actually
17 two different references, Wilcox and Londergan. Are there any,
18 of the check marks under Lotus Notes in Slide 10, are there any
19 of those that are taught only by Londergan, or can you rely just
20 on Wilcox for these aspects of Claim 1?

21 MR. THACKER: These aspects of Claim 1, only Wilcox
22 is needed.

23 JUDGE MURRAY: Thank you.

24 MR. THACKER: I'd like to turn now to the reasonable
25 expectation of success. There are at least four reasons why a
26 skilled artisan would have reasonably expected success in

1 combining Lotus Notes' interfile multi-string indexed search
2 teachings with Raskin's incremental search teachings.

3 One, the Lotus Notes' indexes were of an appropriate size
4 to support an incremental search. Two, incrementally searching
5 across files was known in the art. Three, Dr. Terveen explained
6 at length exactly how a skilled artisan would implement an
7 incremental search with the teachings of Lotus Notes. And
8 four, the patent itself provides little to no explanation of how to
9 solve this alleged large-volume argument that Patent Owner
10 raises, indicating to a POSITA that this would not have been an
11 issue.

12 So, turning first to Dr. Terveen's explanation of the
13 reasonable expectation of success, I'd like to direct the Board to
14 Slides 16 through 24 which show Dr. Terveen's testimony
15 explaining the reasonable expectation of success in combining
16 Lotus Notes' teachings and Raskin's teachings.

17 JUDGE SAINDON: Counsel, I have a question on this
18 topic. So, okay, somebody could code it if they knew what they
19 wanted to code. My question is more, I guess, practical.
20 Looking at, I'm looking at Exhibit 1059, page 163. This is the
21 Query Builder form. And there are a couple of other forms or
22 dialogue boxes, whatever you want to call them, from
23 Exhibit 1059. And I'm wondering, so if I am using this Query
24 Builder and I am getting real-time results, I'm just wondering
25 how that could work.

1 With the Entourage, it was already doing that. I could
2 see how just having the string is not all that much. But on this
3 one, I'm having a hard time seeing what's being displayed in
4 real time in a beneficial way. You have this dialogue box in a
5 way, presumably, or am I wrong on that?

6 MR. THACKER: No, that's correct. And so to answer
7 your question, I mean, Lotus Notes, later, I believe it's on
8 page 165, shows the results pane. And so a skilled artisan
9 would understand that you could have either a single window or
10 a single user interface that presents both the dialogue box and
11 the search results as Lotus Notes does with its just
12 slimmed-down search bar.

13 JUDGE SAINDON: That's the one that's in Londergan?

14 MR. THACKER: The search bar is in both references.

15 JUDGE SAINDON: Okay.

16 MR. THACKER: The simple search bar.

17 JUDGE MOORE: And did you say that -- what you just
18 said, was that in the petition or the declaration of Mr. Treveen?

19 MR. THACKER: On this specific issue, no, because it
20 was not in the briefing.

21 JUDGE SAINDON: So, I guess, just to make sure I
22 understand it. So Exhibit 1059, page 164, there is a search field
23 that you have identified. I'm looking at the 255 Petition,
24 page 16. Anyway, there is a query, "full-time." And is that the
25 bar that you are talking about?

26 MR. THACKER: Yes.

1 JUDGE SAINDON: Okay.

2 MR. THACKER: Yeah. And the other thing I will say
3 on this is that what we're combining is the teachings that a
4 skilled artisan would understand from these references. And as
5 the Board noted in its institution decision, the question is not
6 whether a single embodiment or a single tool in a single
7 reference, you know, discloses every element of a claim. It's
8 what inferences and creative steps a skilled artisan would take
9 from the collective teachings of the references. And here, the
10 collective teachings of Lotus Notes are suggesting to a skilled
11 artisan that you can perform this multi-string query and get
12 results from that query, all in this one tool. And like you said,
13 maybe in that specific implementation of Lotus Notes, there
14 was a separate dialogue box for entering the query, but a skilled
15 artisan wouldn't read that so myopically to, when they're going
16 to implement their own search tool, think, oh, these have to be
17 separate.

18 JUDGE SAINDON: Just a note of the time.

19 MR. THACKER: So returning to the reasonable
20 expectation of success, Dr. Terveen explained in his
21 supplemental declaration, paragraphs 41 to 49, how Raskin's
22 teachings would have required simple and straightforward
23 software. For example, a skilled artisan would have reasonably
24 expected success in using C or C++ languages that skilled
25 artisans were familiar with to implement both the front-end user
26 interface and the back-end logic needed to support incremental

1 results. Patent Owner in its sur-reply does not even address
2 these paragraphs in Dr. Terveen's testimony and chose not to
3 question Dr. Terveen about this in deposition.

4 Further, Patent Owner's expert, Mr. Haeberli
5 acknowledged during his deposition, and that's shown on Slide
6 79, that a skilled artisan would have been familiar with these
7 programming languages and would have been able to use them
8 to implement a dynamic text box which is needed to provide
9 incremental search results. And he said this was evidenced by
10 the fact that incremental search user interfaces were known and
11 existed in commercial products.

12 So now, I want to talk about the incremental searches that
13 were known. And if we turn to Slides 26 to 32, they show that
14 incrementally searching across files was known in the art, thus
15 confirming the reasonable expectation of success. So here on
16 Slide 26, this is showing disclosure from Raskin itself,
17 page 132, discussing how, by the use of indexes and other
18 techniques, the search tool can ensure that the results are
19 responsive enough to maintain an incremental search that's
20 satisfying to the user, that doesn't lag too much.

21 And, as shown on Slides 28 to 32, numerous other prior
22 art references, including Entourage, Eudora, Spaey, Nguyen,
23 and Siitonen, also teach incrementally searching across files.
24 The Patent Owner's expert, Mr. Haeberli, does not dispute that
25 Spaey and Nguyen, at least, teach incrementally searching
26 across files.

1 And so now I want to turn to Slide 34. A skilled artisan
2 further would have reasonably expected success in
3 implementing Raskin's incremental search across files, because
4 Lotus Notes' database, its index, excuse me, its index was small
5 enough to be able to support an incremental search. Here in
6 Londergan, page 136, it expressly discloses that the index for
7 an email database would be 5 megabytes. And we're 15
8 megabytes total minus 10 megabytes for the database. And this
9 was --

10 JUDGE MOORE: Counselor, you're quoting Londergan
11 for the size of the Lotus Notes database?

12 MR. THACKER: This, yes, this express disclosure of
13 the index size.

14 JUDGE SAINDON: Okay.

15 MR. THACKER: Yep. And Patent Owner, one more
16 thing on Lotus Notes. Patent Owner does not dispute that an
17 index of 5 megabytes, that computers at the time would have
18 been able to perform an incremental search across a database of
19 this size.

20 So now I want to turn to what the patents themselves
21 actually say. This is shown on Slides 37 to 39. And the patents
22 do not provide any detail on how to solve Patent Owner's
23 alleged large-volume problem, thus confirming that a skilled
24 artisan would have reasonably expected success.

25 So, for example, on Slide 37, the patent describes using
26 standard inverted indexes to perform the search. Patent Owner

1 does not dispute that inverted indexes were known, and the
2 patent doesn't describe any new or novel way to solve this
3 scalability problem that it identifies. And further, on Slide 38,
4 what the patent is describing here is just a conventional way of
5 using a conventional index to identify matches. It's just looking
6 up in the index to find character strings that match in the index.

7 And then shown up on Slide 39, the patent describes the
8 very problem that Patent Owner alleges. So it's saying here in
9 this section that every single prefix is indexed. Well, Patent
10 Owner alleges in its Patent Owner Response at 58 that if you
11 index every prefix, the index is going to get too large and you're
12 not going to be able to perform the incremental search. So
13 there's nothing here in this patent about how to solve this
14 problem. If anything, the patent itself creates the problem that
15 Patent Owner alleges that there is.

16 JUDGE MURRAY: Counsel, is it true that if you have
17 an index where every prefix is indexed, it will be larger and
18 searches will run more slowly than an index that does not have
19 every prefix indexed?

20 MR. THACKER: I want to be clear about what you
21 mean by every prefix indexed. Are you saying that there is a
22 separate, like, row or record for every single prefix?

23 JUDGE MURRAY: You referred to Slide 37 when you
24 were discussing that. So I guess I would mean the index that's
25 shown on Slide 37.

1 MR. THACKER: Okay. Yes. So, it depends on exactly
2 what data structure you're using. But generally speaking, yes.
3 If you're indexing every single prefix, the size of the index is
4 going to increase.

5 Now, whether or not that it slows down the search
6 obviously depends on the size of the index itself, the original
7 index which -- or the original size of the database, for example,
8 you know, like if you index just whole words on a 2-gigabyte
9 database, that's probably going to have a bigger impact on
10 speed than indexing every prefix in a database that's only 5
11 megabytes.

12 JUDGE MOORE: Right. But if you indexed every
13 prefix in a 2-gigabyte database, that would be a much larger
14 index then if you just indexed for words in a 2-gigabyte
15 database.

16 MR. THACKER: Correct.

17 JUDGE MOORE: You would need to index every prefix
18 in order to perform a search using an index, right?

19 MR. THACKER: No. We disagree with that. Dr.
20 Terveen explained that in his supplemental declaration at
21 paragraph 47, for example, that there are two approaches. One
22 is that you index every single prefix, in others that you index
23 whole words, and then you simply join or merge the rows that
24 contain words that start with that letter, similar to the approach
25 that Green, for example, describes at column 5.

1 JUDGE MOORE: And this is not in the references that
2 you cited. This is more information that would have just been
3 background information that a person of ordinary skill in the art
4 would have known?

5 MR. THACKER: Give me one second. I can get back to
6 you on that in rebuttal, but Baeza-Yates does disclose prefix
7 searches using an inverted index.

8 JUDGE MOORE: Okay. Please get back to me in the
9 rebuttal time, and if the relevant pins are cited in the briefs,
10 please point me to where they're cited as well, thank you.

11 JUDGE SAINDON: Just a reminder, you're a little under
12 40 minutes' total time left.

13 MR. THACKER: All right. Just wrapping up here, I
14 would like to direct the Board to Slide 43. The Ground 2
15 combination, or I'm sorry, I want to go back to the Slide 40. So
16 in sum, on Ground 1, a skilled artisan would have been
17 motivated with a reasonable expectation of success to combine
18 Lotus Notes' teachings to search across multiple index files for
19 multiple search strings entered in a single search field with
20 Raskin's teachings to perform a search incrementally.

21 As explained by Raskin and was well-known in the art,
22 incremental search provided several benefits, including speed,
23 responsiveness, ease of use. And implementing this
24 well-known predictable software in Lotus Notes' 5-megabyte
25 email database, for example, would have required
26 straightforward software modifications, as explained by Dr.

1 Terveen and evidenced by the numerous references that teach
2 incrementally searching across files.

3 If there are no further questions, I'd like to reserve my
4 remaining time for rebuttal.

5 JUDGE SAINDON: Nope. Thank you.

6 JUDGE MOORE: Thank you.

7 JUDGE SAINDON: Before we get started, just two
8 points of order. This clock only goes to 99. So we're just going
9 to start with 90, and then we'll add 15 minutes. Also, I
10 anticipate taking a break after this session if that's all right with
11 everyone.

12 MS. LAROCHELLE: Absolutely, thank you.

13 Apologies, technical difficulties. Good afternoon, Your
14 Honors, Christie Larochelle from Goodwin Procter on behalf of
15 Patent Owner X1. So, as I mentioned earlier, with me is Emma
16 Murray, our LEAP practitioner, and she'll be presenting a
17 portion of this argument. So I'll be covering the background
18 and Ground 1, and Ms. Murray will address Ground 2.

19 So, turning to Slide 3, I want to make sure that we
20 ground ourselves in the patent and the claims at issue here, and
21 I think that that was maybe glossed over a bit in Petitioner's
22 presentation. So I'd like to start with Slide 3. What we have on
23 Slide 3 is an excerpt from the patent on the left. And this is -- if
24 you look at the '035 Patent, this is from column 1, giving a
25 description of the related art, so what was known in the art.

1 So the inventors of the challenged patent identified
2 problems in the existing computer search systems. And
3 specifically at the time of invention, search systems for locating
4 documents or files on your computer were slow and
5 cumbersome. And some of us are old enough to remember,
6 back in the early 2000s, searching for files on your computer
7 could be very slow and very cumbersome, much more
8 cumbersome than today. You'd need to bring out the search
9 interface. You'd type in your search query, hit the search
10 button, and wait for results to populate. Could take a long time
11 sometimes. And then, if your results weren't quite what you
12 were looking for, you'd have to go back and do that all over
13 again. And maybe you had to do it in different programs, too,
14 so you were searching email in a different program from
15 searching for files on your computer.

16 So, turning on Slide 3 -- or Slide 4, I'm sorry -- so the
17 inventors recognized that as computer use became more and
18 more common, more and more part of our daily lives, digital
19 data was accumulating. And the more the digital data
20 accumulated, the harder it became to locate that one file or
21 email that you were looking for. So traditional searching like I
22 just described is sometimes referred to as delimited in that you
23 have to hit enter or the search button to execute the search. So
24 traditional search tools incorporated, you know, search boxes.
25 Maybe there were multiple search boxes. Maybe you could use
26 a Boolean "and" to build your searches. But the bottom line is

1 that search tools for searching for files and documents on your
2 computer were generally delimited, were all delimited, and they
3 required the user to be very proficient in constructing search
4 queries.

5 So, turning to Slide 5, shown on this Slide are some
6 examples of typical search tools at the time for locating groups
7 of documents. Each of these search tools is an example of
8 delimited searching. In most cases, the user would call up a
9 search interface, type in a query, hit enter or the search button,
10 and wait for results to populate in a different window.

11 Turning to Slide 6, and this is where the invention of the
12 challenged patent comes in. The inventors recognize this
13 problem in the art, determined that a shift in mindset was
14 necessary to address the problems. So they conceived of and
15 developed the inventions claimed in these patents. These ideas
16 resulted not only in the challenged patents but also in the
17 formation of the company X1, a small U.S.-based company.
18 They're still based in California. They're still in business today,
19 developing and selling search software. These three challenged
20 patents are from the same family, share a common
21 specification, and each generally relates to incrementally
22 searching for and locating groups of documents using multiple
23 search strings.

24 So the invention overcomes these problems in the art that
25 I identified, and they did this by providing methods and systems
26 for incrementally locating documents or files on your computer.

1 So I think the parties agree that incrementally locating
2 documents means that as you type the search string into the
3 search field, results populate as you type. And so this type of
4 searching allows the user to search dynamically. And in
5 response to seeing the search results as they populate, the user
6 can modify the search string.

7 So, turning to Slide 8, I want to ground us in the actual
8 disclosures of the patent. And so we'll talk about the claims in
9 a minute, but here is how the patent describes the invention on
10 Slide 8. So this is Figure 3A from the patent. And what the
11 common specification describes is a unified user interface that
12 has a search field that you can type multiple strings into. It has
13 a search results pane that populates as you type characters into
14 that search field. And there's a preview pane in that same
15 window that will display a portion of the document, the selected
16 document that pops up in the list.

17 And so this is just the user interface. The patent also
18 provides significant structural and operational detail about how
19 the search is performed, how the indexes are created, the
20 architecture of the Word files, and the manner in which these
21 are searched. And the specification reveals a far more
22 complicated intersection of different techniques that are
23 leveraged to accomplish the claimed incremental searching than
24 the Petitioner would have you believe. There is a far more
25 extensive disclosure of what's going on in the background, if
26 you look -- you can turn to columns 5 through 8, at least, in the

1 patents. But Petitioner has greatly oversimplified what's in the
2 patent for disclosure, to the extent disclosure is even at issue
3 here.

4 So, turning to Slide 9, incrementally locating groups of
5 documents was known at the time of invention. We admit that
6 the phrase incremental search appears in some of the prior art
7 references. But in each case, the references were referring to
8 either searching within the text of a single open document or
9 list filtering, a lightweight list filtering, such as what we saw in
10 the Entourage reference that Petitioner talked about earlier.
11 And my colleague Ms. Murray will talk a bit more about that
12 list filtering tool in Entourage when we get to Ground 2.

13 But regardless, these functionalities are fundamentally
14 different than what's claimed in the challenged patents, which is
15 incrementally searching for and locating groups of documents.

16 JUDGE MOORE: I think you might have anticipated
17 one of my questions, Counselor, but I am interested in -- you
18 know, you labeled it as filtering. I understand that, but I am
19 interested in whether your colleague discusses it.

20 MS. LAROCHELLE: She will.

21 JUDGE MOORE: And how is that different from
22 searching?

23 MS. LAROCHELLE: She'll discuss it in much more
24 detail. And as a preview, I can tell you that all of the list
25 filtering functionality that Petitioner points to, it's -- I would say
26 it's akin to like searching an Excel spreadsheet, where you have

1 got rows with columns of fields. So it's almost like searching a
2 single document rather than searching across documents. And,
3 Your Honor, you noted that this is a much more lightweight
4 functionality than searching, you know, the contents of all of
5 your documents on your computer, but much more limited.

6 JUDGE MOORE: Thank you.

7 MS. LAROCHELLE: Okay. Turning to Slide 23,
8 Petitioner's Ground 1 combination relies on picking and
9 choosing various ideas from different references without regard
10 for how these disclosures would be combined or why a
11 POSITA would be motivated with a reasonable expectation of
12 success to do so.

13 So, for example, Petitioner relies on the Wilcox and
14 Londergan references, specifically I would say primarily
15 Wilcox's Query Builder for the receiving of first string in a first
16 search field; Raskin for the idea of incrementally locating;
17 Petitioner needs to then go back to Wilcox for the
18 groups-of-documents portion of that limitation; and then Wu
19 for the intersection approach.

20 And then for dependent claims, Your Honors ask the
21 question, is Petitioner combining any different functionalities
22 within Lotus Notes, and the answer is, yes, they are in the
23 dependent claims. There are dependent claims where Query
24 Builder does not meet all of their limitations and they do need
25 to go to a different search functionality. So, I'll get to that when
26 we get to the dependent claims. But the answer there is, yes,

1 they are combining multiple different search tools within those
2 references without expressly disclosing how they're combined.

3 Okay. So, turning to Slide 17, so shown on this Slide is
4 Wilcox's Query Builder. And so with this framework in mind, I
5 want to look at what this actually discloses. So, Wilcox
6 describes the Lotus Notes system. And as I just previewed, you
7 know, Petitioner relies heavily on Query Builder but not solely
8 on Query Builder. Petitioner will -- in some of the dependent
9 claims relies on the search-by-forms technology without
10 disclosing anything about how that would be combined with
11 this Query Builder functionality. But this is a typical,
12 traditional, delimited search-for-documents functionality where
13 you bring up a search interface, you type in your query, and you
14 hit the search button, and then you wait and you get results in a
15 different panel. And Petitioner does admit that neither of the
16 Lotus Notes references describes incrementally searching for or
17 locating documents.

18 So, turning to Slide 18, Wilcox and Londergan describe
19 the Lotus Notes system, which is sometimes referred to as
20 groupware, where the software for communication and
21 collaboration included email, file and database organization and
22 management, and as Your Honor -- I think it was Judge Moore
23 -- noted, that was an enterprise system that was designed to
24 handle a large number of documents.

25 Turning to Slide 19, so in contrast, the portion of Raskin
26 that Petitioner relies on describes the search functionality of the

1 EMACS text editor. And I don't know how many of you -- I
2 used the EMACS text editor back in the 90s when I was in
3 college. And this is just a simple UNIX-based text editor.
4 Admittedly, Raskin uses the phrase incremental search when it
5 talks about EMACS and the search functionality, but the way
6 that this so-called incremental search works in Raskin in the
7 EMACS system is, you have a single open document, you enter
8 a search command, and you start typing a search result. And
9 then Raskin scans through all of the characters of the text of
10 that document. And when it finds a match, it moves the cursor
11 to that first match. So, this is just a sequential scanning,
12 brute-force-type method of searching within a single open
13 document. It doesn't return a list of search results. It simply
14 moves the cursor to the first occurrence of a matching string.

15 So moving to Slide 20, in the petition, Petitioner relied
16 on Raskin solely for the idea of incremental search. There is no
17 reference in the petition to Raskin disclosing anything related to
18 searching across documents. And this is clear from the portions
19 of Raskin that Petitioner cites. A couple of examples here.
20 Page 129 in Exhibit 1021, which is Raskin, describes, for
21 example, when you type the first character of the search string,
22 the computer instantly proceeds to find the first instance of the
23 character in the text. And that's what I was just talking about,
24 the sequential scanning and moving the cursor to the first
25 instance of the search string.

1 JUDGE SAINDON: Counsel. Could I jump in here for
2 a second?

3 MS. LAROCHELLE: Absolutely.

4 JUDGE SAINDON: Is it fair to say that an incremental
5 search of the Raskin variety is different from the incremental
6 search over documents? Is that a difference in degree, a
7 difference in kind? And could you explain that?

8 MS. LAROCHELLE: Yeah, I would say it's both, right.
9 You know, sequentially scanning through a single open
10 document, you've got a document. It's open. It's loaded into
11 your RAM. And it's literally scanning through all of the
12 characters. And so that is not something that you would or a
13 POSITA -- and I believe Mr. Haeberli discussed this -- a
14 POSITA -- and I think actually Dr. Terveen agreed with
15 this -- a POSITA wouldn't implement that in a system with lots
16 of documents. Because if you think about it even, if you have a
17 hundred, a thousand, a hundred thousand documents to
18 sequentially scan through all of those characters, that would
19 take an exceptionally long time. So, I would say that's a
20 difference in kind, right. That -- I can't see any way that a
21 POSITA would consider applying that method to a Lotus Notes
22 enterprise-level database. So, that's definitely a difference in
23 kind.

24 But I think it also is a difference in scale, right. You're
25 talking about the characters of a single open document, versus

1 all of the words in thousands of documents. So, I would say it's
2 both kind and scale.

3 JUDGE SAINDON: So, when you're searching multiple
4 documents, you're using an index. When you're searching a
5 single document, you're just proceeding through the text string.

6 MS. LAROCHELLE: Right, and I'm not saying you
7 couldn't use an index in a single document, but that's not what
8 Raskin describes. Raskin is describing the sequential scanning.

9 JUDGE SAINDON: Okay. Thank you.

10 MS. LAROCHELLE: Okay. So I do want to point out
11 today that today for the first time in all of the papers -- this
12 appears in none of the papers -- Petitioner put one or two slides
13 up. It was their Slides 26 and 27 where they said, I believe,
14 quote, Raskin already combined an incremental search with
15 multi-document search or something to that effect. And I just
16 want to point out that this was never argued before. It wasn't in
17 the petition. It wasn't in their reply. There's no disclosure of
18 this in the record. And it is my understanding they are not
19 relying on Raskin for multi-document incremental search. So I
20 just wanted to point out that those two slides appear to make a
21 new argument today.

22 So turning to Slide 22, Petitioner's Ground 1,
23 Wilcox-Londergan-Raskin-Wu combination, fails for a number
24 of reasons. And first and most importantly, the combination
25 doesn't disclose or render obvious all limitations of all of the
26 challenged claims. Second, Petitioner has not articulated a

1 reasoning with rational underpinnings for motivation to
2 combine with a reasonable expectation of success for the Lotus
3 Notes and Raskin references. And then there's also some
4 improper gap filling for dependent claims. And there's also for
5 some of the dependent claims no motivation to combine the
6 distinct search tools in Wilcox and Londergan.

7
8 So turning to Slide 25, if we look just at the
9 Lotus-Notes-Raskin-Wu combination, and really what we're
10 going to focus on is the Lotus-Notes-Raskin portion of that,
11 where Petitioner is oversimplifying and this -- the claims
12 essentially. So the claims require incrementally locating a first
13 group of documents. And this is an example, Claim 1 from the
14 '035 Patent. And what Petitioner is doing is kind of divorcing
15 the incrementally locating from the context of the locating,
16 which is over groups of documents, right. So, they're saying,
17 well, I can take just the idea of incremental search from another
18 reference and combine it with this enterprise-level-file-system
19 search-type functionality of Lotus Notes, and it's just going to
20 work, right.

21 So if we move to Slide 27, Petitioner admits that neither
22 Wilcox nor Londergan disclose incremental search across
23 documents and nor does Raskin. As I just described, Raskin is
24 a brute-force sequential scanning method to find a character
25 string in a single document that doesn't even return a list of
26 search results.

1 And turning to Slide 28, so we have here an excerpt from
2 the petition. Petitioner has provided no explanation for how,
3 how a Petitioner would combine Raskin with Wilcox or Raskin
4 with the Lotus Notes references to achieve incremental search
5 across documents. Petitioner simply says a POSITA would be
6 motivated for reasons of speed, effectiveness and ease of use,
7 and it would have required only simple and straightforward
8 software modifications. This is all the petition says about a
9 motivation to combine these two, well, the Lotus Notes and
10 Raskin references, and reasonable expectation of success. So,
11 Petitioner's argument taken to its logical conclusion would
12 be -- appear to be that any software invention would be obvious
13 with a reasonable expectation of success, simply because
14 anything can be coded.

15 Turning to Slide 29, Petitioner's expert, Dr. Terveen,
16 admitted at his deposition that, quote, Raskin is not explicitly
17 talking about searching across a set of documents and that he
18 was only relying on, quote, the idea of incremental search. So,
19 in other words, because the words "incremental search" appear
20 in Raskin, Dr. Terveen seems to have decided that it could be
21 combined with Wilcox and Londergan. And he insisted that he
22 didn't need to look any further than that. And so for this
23 fundamental reason, Dr. Terveen's opinions about Raskin
24 should not be given any weight. He failed to consider whether
25 Raskin's teachings, the brute-force sequential scanning method
26 of a single open document, would be applicable in an indexed

1 multi-document search system like that described in the Lotus
2 Notes references.

3 Turning to Slide 30, at deposition, Dr. Terveen also
4 admitted that he didn't consider the entirety of Raskin's
5 disclosures. And he couldn't even recall if he had reviewed the
6 entire reference. He admitted that he hadn't considered whether
7 any portion of the reference might teach away. And this is just
8 another reason why Dr. Terveen's opinions on Raskin should be
9 given little weight.

10 So, I apologize, I'm going to skip ahead in the slides to
11 Slide 63 because what I would like to talk about are missing
12 limitations first. I think I want to make sure that we hit the
13 limitations that are not disclosed even if you accept their
14 combination.

15 So, the Wilcox-Londergan-Raskin-Wu combination
16 doesn't render obvious an index that facilitates incremental
17 search or partial strings. So, what we're talking about here is
18 the '490 Patent. And all claims of the '490 Patent require an
19 indexed data structure. So, if we go to Slide 64, I have Claim 1
20 of the '490 Patent up here. And we can see in the highlighted
21 portions, the claim requires an indexed data structure
22 comprising associations between each of a plurality of character
23 strings to associated files. And then later in the claim, it
24 requires searching the indexed data structure for one string that
25 corresponds to the first partial search string and then searching

1 the indexed data structure for one string that corresponds to the
2 second partial search string.

3 So, turning to Slide 65, in the petition, Petitioner relies on
4 the fact that Wilcox and Londergan disclose a full-text index,
5 but they don't point to any disclosures in the Lotus Notes
6 references that would disclose how a full-text index would
7 identify associations to partial search strings as required by the
8 claims. And what we have on this slide is an excerpt from the
9 petition, Paper 1 in the '490 IPR on the left at page 21.

10 Now moving to Slide 66, perhaps recognizing this gap in
11 the disclosure, Petitioner resorts to knowledge of a POSITA,
12 stating in the petition -- and this is highlighted and boxed in red
13 on the left. This is the petition, Paper 1 at 21. Quote, a
14 POSITA knew that this is how indexes work. And Petitioner
15 cites to Dr. Terveen's declaration. But the declaration merely
16 parrots that statement back that a POSITA knew how indexes
17 work, and then as a fallback says it was well-known in the art to
18 use indexes to expedite the search process. But neither of these
19 conclusory statements explain how a full-text index would
20 identify associations to partial search strings as required by the
21 claims.

22 Turning to Slide 68, in reply, Petitioner provided a new
23 theory on this. And specifically, Petitioner argued that, quote, a
24 system that indexes only whole words inherently indexes
25 prefixes, because common words that would be indexed in any
26 system, e.g., the word "a," are prefixes for other words, e.g.,

1 "an," "and," and "Andrew." This argument did not appear in
2 the petition. It appears for the first time in reply, which is
3 improper. Patent Owner did not have a full and fair opportunity
4 to respond, so we ask that this be disregarded.

5 Regardless, the argument fails because most partial
6 strings are not actually words. In fact, at deposition, we have
7 Dr. Terveen's deposition transcript here on the left -- I'm sorry,
8 on the right. That's Exhibit 2026 at page 211 through 212, 213
9 through 214. And what Dr. Terveen was asked, and he had to
10 admit, that if you're going to index the word declaration, you
11 need to index D, DE, DEC, DECL. And he agreed. And I
12 think we can all agree these are not common words that would
13 be indexed in any indexed system.

14 So, I'm going to skip around again and go to Slide 32
15 now. So, in addition to these missing elements, Petitioner's
16 motivation to combine, as stated, fails. So, we have on this
17 slide an excerpt from the petition in the '035, Paper 1. This is at
18 page 25. And this presents Petitioner's motivation to combine
19 Raskin with Wilcox and Londergan. And Petitioner says, a
20 POSITA would have been motivated for the reasons
21 stated -- allegedly stated by Raskin, speed, effectiveness, and
22 ease of use. But Raskin's alleged benefits are limited to, as I
23 explained, the sequential scanning of a single open document.
24 And neither Petitioner nor Dr. Terveen explained how
25 implementing Raskin's technique would improve speed,

1 effectiveness, or ease of use in the search tools disclosed in the
2 Lotus Notes references.

3 Moving to Slide 33, Petitioner relies on Dr. Terveen's
4 conclusory statement that a POSITA would have reasonably
5 expected to succeed because the combination would require
6 only simple and straightforward software modifications.
7 Nowhere does Petitioner grapple with the challenges associated
8 with taking a system designed to handle a large number of
9 documents, the Lotus Notes system, and using -- modifying that
10 system using ideas gleaned from Raskin's brute-force sequential
11 scanning method, which is designed to search for a string
12 within a single open document. And this conclusory opinion is
13 rebutted by Patent Owner's expert Mr. Haeberli's testimony.

14 So, turning to Slide 34, the petition is completely silent
15 on what the proposed combination would look like or how it
16 would work. So, in the POR, Patent Owner was left guessing
17 as to what this combination would actually look like. So what
18 we do know from the petition is that the Petitioner never
19 suggested that Raskin should be modified, so we have to take
20 Raskin as it is, and that is brute-force sequential scanning of a
21 single document starting from the current cursor position and
22 moving that cursor to the first instance of a search string in the
23 text. All the Petitioner says is, take that, put it together with the
24 delimited enterprise-level database searching and Lotus Notes,
25 and it will work.

1 So, left with no details from the Petitioner, Mr. Haeberli,
2 Patent Owner's expert, looked at whether Petitioner's stated
3 motivation would hold up under two possible approaches a
4 POSITA might have considered for combining Raskin with
5 Wilcox and Londergan. And Patent Owner established that
6 even if these references could be combined in one of these
7 ways, the combination would not meet the claim limitations,
8 and Patent Owner's motivation to combine would be defeated.
9 And in brief, the reason for this is because -- well, let's look at
10 Slide 35.

11 This is an excerpt from Mr. Haeberli's supplemental
12 declaration, Exhibit 2020, and it's paragraph 52 in each of the
13 IPRs. And what Mr. Haeberli explains here is that, well, one
14 possibility would be, we can take Raskin's brute-force
15 sequential scanning method, scan through the full text of every
16 document in the Lotus Notes database and do it that way. But
17 as Mr. Haeberli explains, the sheer volume of material would
18 slow the search down so much, so dramatically that this would
19 not be able to be done incrementally. And so, this would not
20 only defeat the stated motivations of speed, ease of use, and
21 effectiveness, it would also not meet the claim limitations,
22 because it would no longer be an incremental search as required
23 by the claims.

24 And turning to Slide 36, Dr. Terveen agrees. This is
25 from Dr. Terveen's supplemental declaration, Exhibit 2023, at
26 paragraph 148 in each of the IPRs. And he even says, rather

1 than scanning every word of every file sequentially which a
2 POSITA knew takes more time, consumes more computing
3 resources than using an indexed system such as that in Lotus.

4 So, turning to Slide 37, to the extent the combination
5 involves implementing using the existing indexed search
6 system in the Lotus Notes references, Mr. Haeberli also
7 explained that doing that would -- trying to implement
8 incremental search there would increase the size of the index
9 dramatically to accommodate all of the potential partial strings.
10 And as we have already discussed, increasing the size of the
11 index dramatically has an impact on search speed.

12 So, skipping ahead to Slide 39, Mr. Haeberli explained
13 why a larger index leads to slower searches. He says, because
14 the larger the index, the more processing power is required, the
15 larger the document that has to be searched. And for most
16 efficient searching -- and I think we talked about this when
17 Petitioner was presenting -- an index should be loaded in RAM
18 for the most efficient searching. And so, the larger the index,
19 the more available RAM is required. And as RAM usage
20 increases, performance degrades. So, the bottom line is the
21 larger the index, the slower the search.

22 Turning to Slide 40, Dr. Terveen, Petitioner's expert,
23 agrees. So, we have Exhibit 2021 is the Terveen deposition
24 transcript at page 38. He says, if you make the index larger, it
25 could increase the amount of time it takes to search it. And he
26 says the same thing in his supplemental declaration,

1 Exhibit 1100, at paragraph 52, would require an even larger
2 index. And that index, you know, it would be difficult to
3 provide results quick enough to perform an incremental search
4 when searching across a large volume of files or documents.

5 So, turning to Slide 41, between the petition and reply,
6 Petitioner's motivation to combine shifted. On reply, Petitioner
7 pivots, argues that this motivation, well, it only applies as long
8 as the system has only two documents. This does not appear
9 anywhere in the petition. Nowhere in the petition does the
10 Petitioner say that -- make any arguments about two or more
11 documents. And so, this is improper. We did not have a full
12 and fair opportunity to respond, and we would ask the Board to
13 disregard this new argument.

14 Regardless, even if you consider it, it still fails.
15 Petitioner's lead references for Ground 1 describe the Lotus
16 Notes system which is expressly designed to handle large
17 numbers of documents. So, Petitioner fails to explain why a
18 POSITA would be motivated to combine these references and
19 limit this combination to a system that can only handle two
20 documents. And a POSITA would not be so motivated,
21 because why would a POSA take a sophisticated
22 enterprise-level system like Lotus Notes and dumb it down so
23 that it can only handle two references. A POSITA would not be
24 so motivated to do that. The whole point of the invention --

25 JUDGE MOORE: I'm --

26 MS. LAROCHELLE: Go ahead.

1 JUDGE SAINDON: Counselor, you were talking about
2 how increasing index size would slow the search. I understand
3 that. Is there any testimony that qualifies that or that points you
4 to say approximately what size of an index, or what size of the
5 corresponding database, would then become unwieldy?

6 MS. LAROCHELLE: Yeah, I don't know that we have
7 an explicit disclosure of when it would become unwieldy. But
8 Mr. Haeberli's supplemental declaration addresses this. And so,
9 I will get that when I step down, and we'll bring that back to
10 you on surrebuttal, if we could.

11 JUDGE MOORE: Thank you.

12 MS. LAROCHELLE: So, one last point here is just that
13 the whole point of this invention and the prior art references is
14 to efficiently deal with the mounds of electronic data that
15 people were accumulating. And so limiting a system to only
16 two documents simply didn't make any sense.

17 So, I'm going to skip ahead to Slide 62 now. I want to
18 talk about some of these dependent claims. So, what I have on
19 the screen here on Slide 62 is Claim 4 of the '490 Patent which
20 is a dependent claim, depends from Claim 1, and it has the
21 additional limitation wherein the indexed data structure
22 comprises indications of character strings associated with
23 respective attribute fields of the files.

24 So, Petitioner fails to point to any disclosures in the art
25 that disclose this limitation. Petitioner again falls back on
26 knowledge of a POSITA, stating on page 32 of the

1 petition -- this is the '490 Petition -- quote, a POSITA would
2 have understood or at least found it obvious that the indexed
3 data structure would need to include indications of character
4 strings associated with the respective attribute fields of the files.
5 So essentially, Petitioner is using the requirement of the claim
6 in hindsight to reconstruct it from the alleged knowledge of a
7 POSITA, saying a POSITA would have just known that you
8 had to do this, you had to do this thing that's required by the
9 claim anyway. And this is improper, and so, as we noted in our
10 POR and our sur-reply, Petitioner has failed to specify where
11 each element of this claim is found in the prior art, patents, or
12 printed publications relied upon.

13 Turning to 43 now, so on Slide 43, I want to talk now
14 about the displaying and highlighting claims. So each of the
15 patents has dependent claims directed to displaying a portion of
16 a selected identified document and other claims directed to
17 highlighting the search terms in the portion of the document
18 displayed. So, these are Claims 2 through 4 of the '035 Patent.
19 Claim 2 comprising displaying a portion of a selected
20 document. And then Claim 3, automatically highlighting a first
21 occurrence of the first word in the selected document. Claim 4,
22 highlighting at the same time a plurality of occurrences of the
23 first word in the selected document.

24 The '139 Patent also includes highlighting and display
25 claims. Claim 3 of the '139 requires displaying in a user
26 interface an indication of the one or more documents included

1 in both the first one or more documents of the search results.
2 Claim 4, further requiring displaying via the user interface a
3 portion of a selected identified document. And then Claim 5,
4 the highlighting a plurality of occurrences of the first and
5 second partial search strings. These are dependent from method
6 Claim 1. Claims 24, 25, and 26 are dependent from
7 independent Claim 23, which is a system claim, but they have
8 similar limitations.

9 And then we have the '490 Patent, which is a computing
10 system claim. So, Claim 1 of the '490 Patent recites a
11 computer-readable storage medium which must cause display
12 of an interactive user interface on the display device, the
13 interactive user interface including at least a first search field.
14 And that's in Claim 1. And then dependent Claim 10 requires
15 display in the interactive user interface a portion of the first file.
16 And Claim 11 requires highlighting a plurality of occurrences
17 of the first and second partial search strings.

18 So, this is Slide 45. So, what we have claimed here is not
19 a generic user interface. It's a specific user interface. It's the
20 user interface of the claims. So, for example, in the '490 Patent,
21 we have a computer-readable storage medium that causes
22 display of an interactive user interface. So, this is the display
23 coded by the software that is the computer-readable storage
24 medium. So, this isn't a generic user interface. It's not any
25 window on your computer. It is the user interface of the search
26 system that's claimed.

1 So, turning to Slide 50 now, I want to refresh us on what
2 the patents describe. And so again, this is Figure 3A I showed
3 you earlier. The patents describe this unified user interface
4 which has a search box, a results pane, and a review pane. So,
5 if we turn back to Slide 47, this is what Petitioner relies on in its
6 petition for the displaying and highlighting claims. And this is
7 just one example. Obviously, there are a lot of claims. But
8 they say the same thing in the petition for each of the three, the
9 three IPRs.

10 So, what they're relying on is Wilcox and Londergan's
11 alleged displaying and highlighting capabilities. So, what I
12 have highlighted in red at the top is an excerpt from the
13 petition, Paper 1 in the '035 IPR. It's at page 34. And quote,
14 Notes, bracket, shows you the docket and highlights the text
15 that you wanted to find. I've put these cited portions of
16 Londergan, the reference that they're citing, quoting here, below
17 it. And so this is Exhibit 1061, the Londergan reference at 146.

18 And I think there was a question earlier about whether
19 Petitioner relied on Londergan versus Wilcox for different
20 functionalities. This is an example of Petitioner relying on
21 Londergan rather than Wilcox for a disclosure. So, Londergan
22 says, you open a document uncovered by a search the same way
23 you open any other document. And so, then we come to -- it
24 says double-click it or select it with your mouse and press enter.
25 So then we come to the sentence that Petitioner quotes, which
26 is, Notes then shows you the document and highlights the text

1 that you wanted to find. So Londergan makes clear that in
2 order to view the document, and not a portion of the document
3 as required by the claims but in order to review the document,
4 you have to open it. You have to double-click. You have to
5 select it and hit enter and actually open it. And only then do
6 you view the document and see the highlighting.

7 So now turning to Slide 48 --

8 JUDGE MURRAY: Counsel, before you do that, which
9 search functionality in Lotus Notes is this describing? Is this
10 Search Builder or the Search Bar, or is it Forms?

11 MS. LAROCHELLE: So, unclear to us, right. So we
12 could look this up in the reference. I believe that this is from
13 the Search Builder in Londergan, but it might also work from
14 the search bar in Londergan.

15 JUDGE MURRAY: Thank you.

16 MS. LAROCHELLE: Yep. Okay. Moving to Slide 48,
17 so acknowledging this gap in the art, Petitioner and its expert
18 again fall back on knowledge of a POSITA. I won't harp on
19 this. This is in our papers. But this is a common theme in the
20 dependent claims where Petitioner can't identify a disclosure for
21 the claims in their primary art, so they say, oh, it was
22 well-known to do so in the art.

23 And then in each case, what they do is they cite the
24 entire -- you know, an entire section of their background. And
25 if you go to that section of the background -- so, in the top
26 excerpt, this is '035, Paper 1, the petition, at page 35. They say

1 it was well-known in the art to do so. Supra Section IV.B.3.
2 And then they cite a single paragraph of Dr. Terveen's
3 declaration. Well, if you go to Section IV.B.3, that's in their
4 background. It's not in their grounds. And in that background
5 section, they just have a long string cite of references with no
6 pin cites in them. So, they're not even pointing you to what in
7 the prior art they're relying on for these claims in their petition.

8 Okay, so I quickly just want to address their new
9 argument in reply for the user interface claim construction.
10 Petitioner argues that a Petitioner would have understood a user
11 interface to be multiple windows, not just one. And they cite
12 expert testimony taken out of context for this. So, at
13 deposition, they ask Mr. Haeberli, completely abstractly
14 divorced from the context of the claims, could a user interface
15 have multiple windows. And Mr. Haeberli, his deposition
16 transcript is 1098. And you can go read for yourself pages 18,
17 line 19 to 20, line 5, which they cite here. They don't directly
18 quote it, but they cite it. He's asked in a very abstract way,
19 could it include multiple windows. And he says, yeah, it could.
20 But he wasn't asked about the user interface of these patents, of
21 these claims.

22 And I just want to point out one last thing before I turn
23 things over to my colleague. The patents distinguish between
24 displaying and opening documents. And we see this in the
25 patent specification. So, if you go to the '035 Patent,
26 Exhibit 1001, Microsoft's Exhibit 1001, column 14, lines 21 to

1 48, I'll just point to the bottom highlight I have on the left here.
2 When the user clicks on a list pane entry, the corresponding
3 contents are displayed in the view pane. If the user
4 double-clicks or presses the enter button on the list pane entry,
5 the search application causes the entry to launch in its native
6 application. So the patents are distinguishing between viewing
7 a portion of the document in the view pane and actually
8 opening the document.

9 Okay. And with that, I will turn things over to my
10 colleague Ms. Murray to talk about Ground 2.

11 JUDGE SAINDON: Thank you.

12 MS. LAROCHELLE: Thank you.

13 MS. MURRAY: Good afternoon, Your Honors. Emma
14 Murray from Goodwin Procter on behalf of Patent Owner X1.

15 So turning to Slide 70, I'll turn to Ground 2 now, which is
16 based on a combination of five references: Schwartz, Sebastian,
17 Barber, True and Baeza-Yates. And I'm going to refer to
18 Schwartz, Sebastian, and Barber as the Entourage references.

19 Turning to Slide 71, so much like the Petitioner's Ground
20 1 combination, Ground 2 is a complicated web of cherry-picked
21 features from the five different references of Ground 2 with
22 very little regard for the context of disclosures that Petitioner
23 relies on. And Slide 71 here is a visual depiction of this picking
24 and choosing. Petitioner relies on True for the first and second
25 string limitations that we see in red. Then for the blue,

1 incrementally locating, and the pink, displaying limitation,
2 Petitioner is relying on the list filtering tool of Entourage.

3 And while Mr. -- or my opposing Counsel now states that
4 Entourage's list filtering tool discloses everything that Petitioner
5 needs in Entourage, they do in their petition cite to the Find
6 tool -- if you can see it in green -- for the groups-documents
7 part of that limitation. And not only does Petitioner rely on the
8 Find tool, they also rely on an Advanced Find tool, which we
9 didn't include on this slide to save space, but they cite them
10 interchangeably. And then, finally, for the orange identifying
11 documents limitation, Petitioner relies on Baeza-Yates.

12 So, like in Ground 1, Petitioner selectively splits
13 limitations to find support in the prior art, resulting in a
14 hindsight-driven combination that does not render obvious the
15 challenged claims.

16 Moving to Slide 72, so Ground 2 fails for several
17 reasons. First, Petitioner has not articulated any reasoned
18 motivation to combine the distinct tools in Entourage. Second,
19 Petitioner has also not articulated a reasonable
20 combination -- reasonable motivation to combine True with the
21 Entourage references. Third, none of the five references alone
22 or in combination discloses or renders obvious incrementally
23 searching and locating documents. Fourth, True is not
24 analogous art. And fifth, Petitioner improperly relies on the
25 knowledge of a POSITA to gap-fill missing limitations in the
26 dependent claims.

1 Turning to Slide 73, so at the heart of Petitioner's attempt
2 to pick and -- of Ground 2 is Petitioner's attempt to pick and
3 choose across three distinct search tools in the Entourage
4 references, to reconstruct the claimed incrementally searching
5 for and locating documents. But Petitioner has not shown that a
6 POSITA would have been motivated to combine these three
7 distinct tools with a reasonable expectation of success. And for
8 this reason, Ground 2 fails.

9 Turning to Slide 74, so the petition's stated motivation for
10 combining the Entourage references is that because they
11 describe the same Microsoft email system, a POSITA would
12 combine them to obtain a more complete and comprehensive
13 understanding of the Entourage system, its features and its
14 capabilities. But this is not a motivation to combine the three
15 separate tools within these references. Petitioner's stated
16 motivation here is more like a reason that a POSITA would
17 have read the references together. But it doesn't say anything
18 about why or how a POSITA would extract a specific feature
19 from a first tool, another feature from a second tool, and yet
20 another feature from a third tool, and combine them together in
21 a specific arrangement required by the claims.

22 And the petition -- I'm sorry. I keep changing the slide
23 with my -- oh, never mind. It didn't change, apologies. The
24 petition never actually clearly identifies which features, tools,
25 or references it's relying on for each claim limitation. Instead,
26 Petitioner misleadingly portrays features drawn from three

1 separate tools as if they were all present in one unified search
2 tool. And now, today, we heard Your Honors ask my opposing
3 Counsel whether Petitioner needs to combine the different
4 aspects of Entourage, and he said no. But that is simply
5 incorrect, and let me explain why.

6 I'm going to skip to Slide 77. So, the Entourage
7 references describe three distinct tools. The first tool that we
8 see at the bottom of the slide is the list filtering tool. The
9 second tool on the top right is the find tool. And the third on
10 the top left is the advanced find tool.

11 Moving to Slide 78, so just as an initial note, Entourage
12 is not an indexed system. All searching in Entourage is via a
13 brute-force method. So, starting with list filtering, the list
14 filtering tool is only available in Entourage windows that
15 display Entourage -- lists of Entourage items, such as emails or
16 address book entries. And this tool allows the user to filter by
17 various attributes of the items in the list. For searching an
18 address book list, as we see in the figure on this slide, these
19 attribute fields include name, company, work phone, home
20 phone, et cetera. And the list filtering tool can filter one single
21 attribute at a time.

22 And as Judge Moore noted earlier, list filtering does not
23 search the full document. The way this tool works is by
24 extracting attribute data and compiling it into a table as my co-
25 counsel Christie compared to an Excel table. Then the tool will
26 scan that table and eliminate items in the table that do not

1 match the filtered text that the user has typed in. For example,
2 if you were searching the "from" attribute in a window that was
3 displaying a list of emails, and you typed in the name Christie,
4 you would see the list filtered down to email items sent by
5 Christie, but you would not see any items in that list that
6 contained the name Christie in the body of the email. So you
7 aren't actually searching the entirety of the document. And the
8 claims here require incrementally searching documents, not
9 some portion of it or extracted attributes. And so, the list --

10 JUDGE MOORE: Counselor, let's look at, for example,
11 Claim 1 of the '035 Patent. So, it's your position that Claim 1 of
12 the '035 Patent requires a search of the entire document? It
13 would not be satisfied by a search of a portion of the document?

14 MS. MURRAY: Yes, that is our position, Your Honor.

15 JUDGE MOORE: Okay. Thank you.

16 MS. MURRAY: And just to add onto the question, the
17 reason that that is our position is because if you can look at
18 some of the more narrow dependent claims, for example, the
19 highlighted claims that my co-counsel discussed, and they
20 require things like displaying a portion of that document and
21 then highlighting words in the document. And so, in order to
22 do that, you would have to search through the entirety of the
23 document.

24 JUDGE MOORE: Okay. And what I am -- specifically
25 what I am wondering about is the limitation that Petitioner's
26 Counsel discussed during their time where you would take the

1 filter functionality that was applied to the subject field. And I
2 believe Petitioner's Counsel stated that a person of ordinary
3 skill in the art would have applied the teaching of multiple
4 search strings into a field to that specific search functionality in
5 Entourage, that that would have yielded the claimed invention.

6 And I'm trying to understand why that -- putting aside the
7 issue for a moment whether or not that specific implementation
8 was actually disclosed in the petition, I'm setting that aside. I'm
9 trying to understand whether that implementation would or
10 would not have satisfied Claim 1 of the '035 Patent.

11 MS. MURRAY: Our position is that it would not satisfy
12 Claim 1 of the '035 Patent because of the same reason that
13 when you are just filtering a single attribute field of these list
14 items, you are not searching the document.

15 JUDGE MOORE: And that gets to another issue. How
16 is a filter different from a search? I understand it's not an
17 indexed search, but how is that not a search?

18 MS. MURRAY: So, you know, whether we call it
19 filtering or searching, that is maybe more of a semantics issue,
20 but it comes down to whether you are searching the content and
21 the entirety of the document. So, if you have a table of attribute
22 data and you're just running down that table, that is much more
23 like searching within a single document than it is searching
24 an -- the entirety of a document.

25 JUDGE MOORE: Right. I understand that's a -- it
26 seems like a control-F-type brute-force search. And I also

1 understand your position the Entourage references are not
2 indexed. I guess that the issue that -- maybe there's no answer
3 to it. But, you know, when I heard Petitioner make that
4 argument, I looked at Claim 1, and I'm having trouble seeing in
5 Claim 1 any limitation that would require the search to be
6 applied to an entire document. And I'm trying to understand
7 why -- you know, obviously, an email has multiple fields if it's
8 in a database. It could have a subject, sender, body. But if the
9 search was just performed in the subject field, why would that
10 not satisfy Claim 1?

11 MS. MURRAY: So, if we look at Claim 1 of the '035
12 specifically, it requires incrementally locating documents. And
13 the Entourage list filtering tool is not -- or locating groups of
14 documents. The Entourage list filtering tool is not doing that.
15 You eliminate entries in your list of attribute data that do not
16 contain a matching string that you typed in. You're not
17 searching the entirety of the document and then locating
18 documents that contain that search string.

19 JUDGE MOORE: So, it's a different type of operation.
20 Rather than identifying a group, you're just striking -- you're
21 striking, I guess, from the group the things that don't match.

22 MS. MURRAY: That's correct, Your Honor.

23 JUDGE MOORE: Okay. Thank you.

24 MS. MURRAY: So if we pick up here with list filtering,
25 we heard Petitioner argue that the list filtering tool does
26 incrementally search documents. But if this were the case,

1 Entourage's find tool, which I'll discuss this in detail later, but
2 this tool would be redundant. The find tool actually does
3 search the entirety of documents, but it's a delimited search that
4 the references describe as very -- a very slow process. And this
5 is precisely because searching the entirety of a large group of
6 documents takes a very long time.

7 And now, I'd like to quickly point out that when
8 Petitioner discussed the list filtering tool earlier, they displayed
9 a figure. I believe it's on Petitioner's Slide 46, and this figure is
10 from Exhibit 1043. Now, Exhibit 1043 is not one of the five
11 references in Petitioner's Ground 2 combination. Ground 2 has
12 no fewer than three references that describe the Entourage
13 system, yet it turns to one of its background references to
14 portray the list filtering tool it selects. And this figure never
15 appears in the petition. Petition pages cited on Petitioner's Slide
16 46 include a long string cite with a see-also at the end for
17 Exhibit 1043. And it's the same in Dr. Terveen's declaration.
18 And so, Petitioner's reliance on this figure in their slides today
19 at hearing is a new argument, and Petitioner -- or Patent Owner
20 did not have a chance to fully and fairly respond to this new
21 argument. So, we would ask that it be disregarded.

22 Moving to Slide 79, so I had mentioned the Find tool a
23 little bit, but I'll discuss it in more detail now. This is the
24 second Entourage tool that Petitioner relies on. And this tool
25 performs a delimited search across a set of documents that the
26 user specifies. And the Find tool is limited to a single string in

1 the search box. And we can see on Slide 79 that the user has
2 chosen to search messages in their inbox. And once the user
3 types a full string and presses the find button, Entourage will
4 display a list of results in a separate window.

5 Moving to Slide 80, the third tool describing the
6 Entourage references is the Advanced Find tool. To access
7 Advanced Find, user first opens the Find tool and then clicks
8 more options. And a separate or an alternate version of the
9 dialogue box will appear that will allow you to search multiple
10 criteria, search multiple items and any folders within the
11 Entourage system. And like the Find tool, Advanced Find
12 performs a delimited search on a single string.

13 I'm going to jump back to Slide 75. So, with this
14 background in mind, let's consider what the petition actually
15 does. On the left-hand side of Slide 75 here is an excerpt of the
16 petition. And this excerpt would have us believe that the
17 Entourage references describe something Petitioner calls the
18 Entourage incremental quick find search. But no such search
19 tool exists in Entourage.

20 This figure that we see at the top of this excerpt is
21 actually the list filtering tool, which, as I just discussed, is not
22 incrementally searching and locating groups of documents.

23 Then, below the figure, we have a description of this tool.
24 But this is actually describing the Find and Advanced Find
25 tools. But Petitioner portrays it as if it is describing the tool
26 shown in the figure above, which is again, the list filtering tool.

1 JUDGE SAINDON: Counsel, if I could jump in here.

2 MS. MURRAY: Yes.

3 JUDGE SAINDON: So what about this Figure 1.12 is
4 not finding a document? It's finding something, right? We
5 have these names. Are you saying that those things aren't
6 documents, like in the way that they're stored in the system?
7 I'm trying to understand what you mean by aren't finding
8 documents.

9 MS. MURRAY: It's not our position that these would
10 not be documents. It's more the method by which this
11 incremental list filtering tool retrieves them. So it's really not
12 retrieving them. It's like I had discussed before. It's looking at
13 a list of attribute data, and then it is filtering down that list itself
14 to eliminate entries that don't have the matching text. So while
15 you are seeing, I guess, documents appear on your screen,
16 you're seeing them sort of be filtered down. And they're
17 not -- whatever you type in this box is not going to give you
18 results that have that term anywhere else in the document
19 besides just if you're searching names, for example.

20 JUDGE SAINDON: I guess I'm trying to, in my head,
21 separate what finding is -- you know, maybe this in the
22 background is like a SQL query on a table or something like
23 that, that that's not searching in the way that the claims require?
24 And perhaps there's no answer to that, but to the extent you
25 have anything that could help me understand.

1 MS. MURRAY: So, I think -- I had mentioned earlier
2 that Entourage has the list filtering tool, and it has the Find tool.
3 Now, the Find tool actually does -- like, if you type something
4 into the Find tool, if I type in, what's in here is Schwartz,
5 Schwartz is actually searching the full content of the document.
6 It's giving you -- it's doing a delimited search to do that, and it
7 returns results that have that term within the document. And
8 now, it describes -- like the references describe this Find tool as
9 being pretty slow, a pretty monotonous process, and it's totally
10 separate from the list filtering tool.

11 If the list filtering tool is truly searching documents, there
12 would be no reason to have a separate Find tool. If they, you
13 know, if they were truly doing the same thing, you wouldn't
14 need to separate these functionalities. And I'll reiterate that our
15 position is that the claims require searching the entirety of the
16 document because the dependent claims then require you to
17 highlight terms that appear inside the body of the document.

18 So after this portrayal of the three tools as one, we then
19 also -- I'm turning to Slide 76 -- we then see Petitioner relabel
20 another figure as the Entourage incremental quick find search.
21 And this time, the figure that we see on the left in this excerpt
22 of the petition is the Find tool, which cannot even perform an
23 incremental search. And so again, the Petitioner collapses the
24 distinct tools into one. Petitioner never explains why a
25 POSITA would be motivated to combine the three distinct
26 tools, and a POSITA doesn't -- Petitioner doesn't explain how a

1 POSITA would actually implement them into a single search
2 tool with a reasonable expectation of success.

3 And where a combination involves multiple
4 embodiments from a single reference, Petitioner must articulate
5 a motivation to combine these embodiments in the precise way
6 required by the claims. Petitioner has not done that here, and
7 this failure of proof is fatal to Ground 2 on its own. And
8 Petitioner's failure is not surprising, because the Entourage
9 references actually explain why a POSITA would not have been
10 motivated to combine these Entourage tools.

11 Turning -- I'm going to skip ahead to Slide 81. So as
12 Patent Owner's expert Mr. Haeberli explained in his
13 supplemental declaration, Entourage's separation of its search
14 tools was driven by performance considerations. The Find and
15 Advanced Find tools are described in the references as slower
16 searches, particularly when searching across many items. And
17 the list filtering tool, on the other hand, is a separate, more
18 lightweight feature designed to quickly narrow list items
19 without triggering a more resource-intensive search across
20 multiple documents.

21 Turning to Slide 82; so Barber, which is one of the
22 Entourage references, describes the Find tool as Entourage's
23 slower, more powerful search, and warns that if you want to
24 search across all items, be prepared to rent a video during the
25 time Entourage will take to search its entire multi-megabyte

1 database. And the Advanced Find tool is similarly described as
2 a lengthy search.

3 Now, a POSITA reading Barber's description of these
4 tools would understand exactly why the designers kept them
5 separate, and that's because they knew that even a simple
6 delimited search of the entire Entourage database would be a
7 painfully slow process. And so applying the Entourage list
8 filtering tool, which is an incremental, as-you-type updating, to
9 the Find or Advanced Find tools that actively search
10 documents, would -- without modification, would result in a
11 search that's not actually incremental, because Entourage's
12 brute-force search method is simply not fast enough to update
13 as you type when you're searching across multiple documents.
14 And this is consistent with what we see in the prior art as my
15 co-counsel explained earlier.

16 For these reasons, a POSITA would not be motivated to
17 combine the three Entourage search tools. Petitioner's stated
18 rationale for combining the references generally, which is that a
19 POSITA would seek a comprehensive understanding of
20 Entourage, lacks the necessary justification for combining
21 different embodiments within the Entourage references. We
22 just saw that the Entourage references themselves provide
23 affirmative reasons why a POSITA would be motivated to keep
24 these tools separate, given the documented performance issues
25 that drove the design decision to keep them that way.

1 Moving to Slide 83; so setting aside the petition's failure
2 to show a motivation to combine the three Entourage tools,
3 Petitioner separately fails to show why a POSITA would have
4 been motivated to combine True with the Entourage references.

5 Moving to Slide 84; so I don't have it on the slide, but on
6 page 40 of the '035 Petition, and similarly in the other two
7 petitions, Petitioner concedes that the Entourage references do
8 not disclose receiving two partial search strings in a single
9 search field, and specifically relies on True to fill this gap.
10 Now, if Petitioner is indeed relying on True solely for
11 disclosure of two strings in a field, as the petition states, the
12 combination fails because the Entourage references do not
13 disclose incrementally searching for and locating documents.
14 And True's disclosure of two strings in a field does not change
15 that.

16 But in its reply, and as we just heard my opposing
17 Counsel state, Petitioner now seems to be relying on True for
18 the search mechanism that True uses to incrementally search on
19 those two strings. First, I'll say this was a new argument in
20 reply and here today, and Patent Owner did not have a full and
21 fair opportunity to respond, and so we would ask that Your
22 Honors disregard this argument. But even if Petitioner is
23 allowed to rely on True's search mechanism, the combination
24 still fails.

25 The petition -- sorry, I'm staying on Slide 84. So the
26 petition's stated motivation for combining the Entourage

1 references with True is that a POSITA would have known that a
2 multi-string search would return more relevant search results
3 than a single-string query. It also quotes one of its background
4 references, Exhibit 1057, which states that searching for two or
5 more keywords, both of which must appear in the results, is an
6 excellent strategy for doing precise searches and greatly
7 reducing the number of results you have to consider. So that
8 was what Petitioner quoted in its petition. We can see it in this
9 excerpt at the bottom in the parenthetical, but Petitioner did not
10 actually consider the context of True, and therefore
11 misunderstands True.

12 Moving to Slide 85. As Judge Moore so adeptly
13 explained earlier how True works, we know that True discloses
14 a system for visualizing the contents of an audio file using a
15 text transcript generated by a speech recognizer. And speech
16 recognizers are not a hundred percent accurate. So to overcome
17 this issue, True uses a confidence-level method for searching
18 for strings in the transcript. And for our purposes, it's important
19 to understand that True's confidence-level technique does not
20 require both strings to appear in the transcript window result for
21 it to be displayed as a search result.

22 And now, I know my opposing Counsel already
23 discussed this, but a transcript window is just a portion of the
24 transcript that corresponds to a certain length of time in the
25 transcript that the user specifies. In this slide in the figure, it's
26 five seconds. So using True's search method, as long as the

1 sum of the confidence levels in a given transcript window
2 exceeds a threshold, that transcript window will be displayed as
3 a result.

4 So in the figure, we have the strings "Queen" and
5 "Mother." And using this technique, a transcript window that
6 has, say, three occurrences of the word "Queen" with high
7 confidence levels could be returned -- and no instances of
8 "Mother" -- could be returned as a result, as long as that interest
9 level exceeds the designated threshold. So while True may
10 disclose two strings in its search field, True does not actually
11 require both of those strings to appear in the results. And True
12 can more accurately be described as searching the transcript for
13 sections that might contain at least one string.

14 And once you consider how True acts on these two
15 search strings, you see that True's method is inherently
16 imprecise. And so, if a POSITA was motivated by the idea that
17 multi-string search, where both strings must appear in the
18 results, returns more relevant search results, as the petition
19 contends, they would never look to a two-string search method
20 where the results don't even have to contain both strings.

21 And so Petitioner's motivation to combine True's
22 multi-string search with the Entourage references rests on an
23 assumption that a POSITA would look at True's multi-string
24 disclosure in a vacuum without considering True's full
25 teachings. But if Petitioner is really relying on True's search

1 mechanism, this is not the case. And looking at the True
2 without considering its context is improper.

3 And moving to Slide 86; and when Dr. Terveen was
4 asked about this at his deposition, he admitted that he didn't
5 think True's search method using confidence levels was
6 relevant. And because he didn't even consider that True's
7 search method returns results with only one string, his opinions
8 should be given little weight.

9 So turning to Slide 87; similar to Ground 1,
10 Petitioner -- the petition gives a conclusory statement of
11 reasonable expectation of success, saying that the combination
12 would require only simple and straightforward software
13 modifications.

14 Turning to Slide 88, my co-counsel walked through why
15 it would be neither simple or straightforward to implement a
16 brute-force search of a single document in a system with many
17 documents like Entourage. I'm not going to rehash it all, but
18 just as a summary, incremental search would require an index,
19 and a large index, which would take up a significant amount of
20 RAM in the user's computer and would cause a slow search and
21 degradation across the entire system. We know that brute-force
22 searching is quite slow, and so doing it across documents might
23 even be slow enough that the search would not be incremental.

24 Turning to Slide 90; so Ground 2 also fails because the
25 combination does not disclose or render obvious incrementally
26 searching and locating documents.

1 Skipping to Slide 72, the petition relies on Entourage's
2 list filtering tool and Find tools for this limitation, but has not
3 established that this combination actually discloses the
4 limitation. As I discussed earlier, list filtering is not actually
5 incrementally searching and locating documents. It is simply
6 filtering certain attribute data, but does not search documents in
7 their entirety. And the Find tool searches documents but not
8 incrementally.

9 So as I explained, Petitioner has not shown that a
10 POSITA would combine the list filtering tool and the Find tool
11 to arrive at the incremental search of documents. Instead,
12 Petitioner splits this limitation into two parts because it cannot
13 point to any disclosure in the prior art that teaches
14 incrementally searching and locating documents. And this is
15 just high school hindsight bias. Petitioner has found separate
16 disclosures, neither of which teaches the full limitation, and
17 then uses hindsight as a threat to stitch them together to arrive
18 at the full limitation.

19 JUDGE SAINDON: Counsel, if I could jump in here.
20 Two things. First, I was able to add nine more minutes to your
21 15. So I still owe you six on this clock. Just to let you know
22 what the time actually is, it's six more minutes than that.

23 MS. MURRAY: Thanks.

24 JUDGE SAINDON: And I do have a question, looking
25 at the '139 Patent. If you go to Claim 12, depends from Claim
26 1, it says wherein the search field corresponds to a single field

1 of the documents and that incremental searching is performed
2 only on content of the single field. Then we go to Claim 13,
3 which depends from 12, and it says, “wherein a single field of
4 the document comprises one of to, from, cc, subject” -- and I'll
5 stop there. But it does go to, “or body fields” of email
6 documents.

7 So, anyway, 13 has email for sure, but 12 is kind of
8 limited to fields, which is getting closer to what's shown in
9 Slide 92 that you have here, this Figure 1.12, where we have
10 maybe not searching in a body, but searching in a field. It
11 seems to me that this series of dependent claims may actually
12 allow Claim 1 to be broader to include both searching the body
13 of a document but also these kind of specific fields. So I'm
14 wondering if Figure 1.12 here isn't showing us searching one or
15 more documents. It seems to me like it is.

16 MS. MURRAY: So I think we would still maintain that
17 it's not locating documents. In the way that at least the '035 and
18 the other claims don't say locating but they say searching, we
19 would maintain that it's still not searching because it is still just
20 simply filtering this list attribute data.

21 JUDGE SAINDON: So the actual mechanics of what's
22 going on?

23 MS. MURRAY: Yes, that's correct.

24 JUDGE SAINDON: Not what it's looking at. That's
25 how you're delineating it, okay.

1 MS. MURRAY: So just to finish out this slide, I'll just
2 add that the petition is silent on how a POSITA would have
3 combined different tools. And for this reason alone, the
4 Ground 2 fails because the combination does not teach this
5 limitation.

6 I'm going to skip to Slide 101. So the Ground 2
7 combination does not render obvious the highlighting claims.
8 And you know, Ms. Larochelle walked through these already,
9 but Claims 3 and 4 of the '035 Patent require highlighting first
10 occurrence, and then 4 requires highlighting a plurality of
11 occurrences of the first string, and then Claims 5 and 26 of the
12 '139 require highlighting a plurality of occurrences of both
13 strings, as does Claim 11 of the '490 Patent.

14 So moving to Slide 105, Petitioner actually relies on a
15 fourth Entourage search tool for these claims, and this is the
16 Find tool within a single document.

17 Moving to Slide 106; as we know, the Find tool is able to
18 search across multiple documents, but it also has another mode
19 where it can be used to search the -- within the text of a single,
20 already open document. And in this mode, the user would open
21 their document, open the find dialogue box, and type in their
22 search string, and then press find, and the tool will scroll to and
23 highlight just the first occurrence of the single string that you
24 typed into the search field. And now this scrolling and
25 highlighting feature is not available when you're using Find
26 within -- or across documents. It's only available when you are

1 using it to search within a document. And the list filtering tool
2 does not do any sort of highlighting.

3 Petitioner has failed to identify any disclosure in the
4 Ground 2 references that highlights a plurality of occurrences of
5 the first string as required by at least Claim 4 of the '035 Patent
6 and Claims 5 and 26 of the '139 and Claim 11 of the '490. And
7 nor has Petitioner identified any disclosure in Ground 2 that
8 highlights a plurality of occurrences of both the first and second
9 string, which is required by the '139 and '490 highlighted
10 claims. And so the Ground 2 combination does not render
11 these claims obvious.

12 Moving to Slide 107; as a fallback, the petition and Dr.
13 Terveen improperly rely on the knowledge of POSITA to meet
14 highlighting limitations without pointing to the exact prior art
15 disclosures that it's relying on, and this is improper.

16 Skipping to Slide 110; as my co-counsel said earlier,
17 relying on the knowledge of a POSITA is a common theme in
18 this petition. We see it happen again for Claims 9, 14, 19, and
19 then the parallel Claims 11, 16, and 21 in the '139 Petition.
20 And this is, once again, improper reliance on knowledge of a
21 POSITA.

22 And then moving to Slide 111, Petitioner does this again
23 for Claim 17 of the '139 Patent. And my co-counsel discussed
24 Claim 17, but it requires that the first string searches on all the
25 textual content of the document, and the second string is limited
26 to just a single field. And Petitioner improperly cites

1 background art that only includes examples of the second
2 requirement that the second string is limited to a single field.
3 But Petitioner never explains why a POSITA would combine a
4 first-string full-text search and a second-string single-field
5 search into one search field.

6 So does the Panel have any further questions?

7 JUDGE SAINDON: I don't see any.

8 MS. MURRAY: Okay.

9 MS. LAROCHELLE: If it's okay, I'd like to take a few
10 minutes from what time is left.

11 JUDGE SAINDON: You'll have a response time, if
12 you'd rather save it.

13 MS. LAROCHELLE: No. I only need just a few
14 minutes here. I actually want to address a question that you
15 had, Judge Saindon. So you had a lot of questions about list
16 filtering and how is that different from locating documents.
17 And so I want to, if we could -- let's see if I can find the slide
18 that has the patent claims on it. So, in Patent Owner's
19 demonstratives, I'm going to go to -- it's around Slide 11.

20 Okay. So on Slide 11, we have Claim 1 of the '139 and
21 the '035 Patents, and I think -- I can't remember if it was Judge
22 Moore or Judge Saindon. One of you was pointing specifically
23 to the '035 Patent. And so I want to make sure that we're
24 covering these limitations very clearly. This is a method of
25 performing a search, and requires, in response to receiving the
26 first search string, incrementally locating a first group of

1 documents that has at least a first word that begins with the first
2 string.

3 So the -- I think the questions you were asking my
4 colleague, Ms. Murray, had to do with, well, how, you know,
5 struggling a little with how list filtering is different from
6 searching for and locating documents. And so a couple of -- I
7 have a couple of responses to that, and one is that all of the list
8 filtering disclosures that we've seen from Petitioner disclose
9 basically what Entourage does, which, to us, to the best of our
10 ability to tell, what Entourage is doing is essentially extracting
11 these attributes from the actual documents themselves and
12 giving you a list of them.

13 So I feel like it's akin to an Excel spreadsheet, right? So
14 you've got basically columns with your attributes and rows that
15 are the different, you know, whether it's a contact with their
16 name and their address and their phone number or whatever it
17 might be. So you have this existing list. And when you're
18 typing into that filter field, you're actually removing documents
19 from the list rather than going out and locating documents and
20 bringing them to the search results field.

21 And as my colleague, Ms. Murray explained, certainly
22 there are dependent claims that would appear to -- or that do
23 require searching on these fields, the attribute fields, such as
24 emails. There are other dependent claims that have different
25 limitations. And Claim 1 must be broad enough to encompass
26 all of them. So to the extent, Claim 1, you have a dependent

1 claim that requires searching the body of a document, Claim 1
2 has to be broad enough to cover that dependent claim as well.
3 You can't -- if a dependent claim is infringed, you also infringe
4 the broadest independent claim that it depends from. So,
5 essentially, you know, you're not incrementally locating your
6 group of documents if you're searching something that is akin
7 to an Excel spreadsheet that just has the extracted attributes in
8 it.

9 I wanted to say one other thing that I forgot to say on my
10 initial presentation. Petitioner said that Spaey and Nguyen, that
11 we don't agree -- that Patent Owner does not disagree that they
12 teach incremental search across files. I want to be clear that we
13 do disagree, and there's nothing in the record that I am aware of
14 where we didn't dispute that -- we do dispute that they teach
15 incremental search across documents.

16 So -- and then finally, I think, Judge Moore, I think you
17 had asked me for evidence in the record about any calculations
18 we have done. There are some calculations about sizes of
19 indices in the supplemental Haeberli declaration, Exhibit 2020.
20 In the 253 IPR for the '035 Patent, that's at paragraphs 47
21 through 48.

22 And one more thing, I just want to make sure the record
23 is clear. In a brute-force search that takes a long time to return
24 search results, this is not going to be incremental. If it's so slow
25 that it can't return results as you type, that is not incremental.
26 And we just want to make sure that we're clear on the record

1 about that. Okay. And we'll reserve any remaining time for
2 rebuttal.

3 JUDGE SAINDON: Okay. Thank you. Okay. So we
4 can take a quick pause here and go off the record.

5 (Whereupon, the above-entitled matter went off the
6 record at 3:25 p.m. and resumed at 3:40 p.m.)

7 JUDGE SAINDON: Please be seated. And Petitioner,
8 you'll be starting out. Looks like you're already set up. I just
9 need to get the clock going. All right. You may begin when
10 ready.

11 MR. THACKER: All right. May it please the Board,
12 Sam Thacker on behalf of Petitioner, Microsoft. And I'd like to
13 cover four topics in the rebuttal. One, some high level themes.
14 Two, new argument that was raised for the first time today.
15 And then, I'll go into Ground 1 and Ground 2 specifics.

16 So starting with high-level themes; what Patent Owner
17 seems to be doing throughout the papers and again here today at
18 hearing is to impose an overly strict obviousness standard,
19 where essentially it seems they're faulting references for not
20 disclosing every single claim feature. They're faulting
21 references for not having a perfect one-to-one correspondence,
22 and they seem to be ignoring what the Board stated in its
23 institution decision, that the test for obviousness is whether the
24 teachings of the prior art, taken as a whole, would have been
25 obvious. And nonobviousness cannot be established by
26 attacking references individually, and the issue, as the Board

1 stated, is what inferences and creative steps a person of skill in
2 the art would have employed when designing a new method
3 and system for search indexing. That's the proper test for
4 obviousness.

5 And again, Patent Owner relies on this bodily
6 incorporation argument that, oh, well, you know, Petitioner
7 didn't consider the context of Raskin or the context of True.
8 And it's not necessary for the entirety of a secondary reference
9 to be bodily incorporated into another reference, or for a skilled
10 artisan to read a single reference and understand that the
11 different embodiments or different features that are described
12 are not compatible and can't be combined.

13 So, with that sort of framing in mind, I'd like to turn to
14 the grounds and walk through them. But first, I want to address
15 this argument that I heard for the first time, which is that -- and
16 I guess, I think that Patent Owner walked this back, but that the
17 claims require searching the entire content of documents. And
18 that's just simply wrong. The dependent claims recite searching
19 within specific search fields. The independent claims have to
20 be broad enough to cover that. This was not argued in any of
21 the briefs, and I would ask the Board to disregard this
22 argument.

23 So now, I'd like to turn to some Ground 1 specifics.
24 First, just as a housekeeping matter, Patent Owner stated that in
25 the reply, we imposed this two-document requirement. That's
26 actually just totally false. '490 Reply at 5: A POSITA would

1 have been motivated with a reasonable expectation of success
2 to combine Lotus Notes with Raskin and Entourage with True,
3 at least because 2002-era computers could provide the response
4 times needed to perform an incremental search across a
5 plurality of files, especially while using known indexed data
6 structures such as Lotus Notes' full-text index when searching
7 across larger volumes of files. I don't see anything in there
8 that's saying we're limiting it to two files.

9 All right. So now, I want to actually get into the real
10 substance of Ground 1. First, Patent Owner talked about how
11 the patent provides details on how to solve its alleged
12 scalability problem. But if you notice, they didn't actually walk
13 through any of that disclosure in the patent. They also didn't
14 explain it in the sur-reply. And as I explained in my opening,
15 those passages just don't provide any support to solve this
16 alleged scalability problem.

17 And then I want to turn to the Raskin -- actually first,
18 backing up. I want to focus on the highlighting, because Patent
19 Owner repeatedly accuses us of relying on background
20 knowledge which they say is insufficient. Now, what we
21 explained in the reply is that Dr. Terveen -- we cited to Dr.
22 Terveen. And they said, oh, we didn't provide any pin cites.
23 Well, Dr. Terveen's declaration has a robust state-of-the-art
24 section with pin cites to record evidence. And this is exactly
25 the kind of situation where the Federal Circuit has found that

1 reliance on background knowledge is sufficient where it is
2 supported by record evidence.

3 This is just one example, shown here on Slide 67, where
4 we cited to background references showing highlighting within
5 a document in the same user -- in the same window, in the same
6 window as the search results, Google and xInquery. Patent
7 Owner has just not addressed this.

8 And just to be clear, if we go back to Slide 65, Entourage
9 already discloses this same thing as the patent, which is to show
10 a preview of the message in the same window as the other
11 search results. So all it's missing here is express disclosure of
12 highlighting the search strings in that preview pane, which --

13 JUDGE MOORE: Counselor. I believe -- I believe this
14 is the window that Patent Owner said was not displaying the
15 incremental search. This was something else. Can you respond
16 to that?

17 MR. THACKER: Yes. So first, this is actually showing
18 Exhibit 1004, which is one of the references that we are
19 expressly relying on. And then on that point, that we were
20 pointing to disclosure in Entourage from Exhibit 1043 that's not
21 actually part of the grounds; well, if you look at Exhibit 1004,
22 which is part of the grounds, at page 8, which is cited by the
23 petition at pages 64 and 66 and cited by Dr. Terveen in his
24 supplemental declaration at paragraph 26, it says, quote,
25 filtering is the fastest way to perform a simple search.

1 And I don't want to waste time on this, but you can look
2 at our demonstratives and our briefing, but the patent repeatedly
3 refers to searching and filtering interchangeably.

4 JUDGE MOORE: Right. But did that specific figure
5 that you were just showing, did that figure display an
6 incremental search? I guess I want to -- I thought I understood
7 Patent Owner to say that this was something different than the
8 filtering interface. And I see at the top it has subject, and then
9 contains, and then a field. So I'm wondering if you can respond
10 to their assertion as I understood it that this was not a filtering
11 interface, that this was something different.

12 MR. THACKER: Well, that's just simply wrong. As
13 you can see the laser pointer here is showing that quick find
14 search field right here, and with the corresponding tabs for the
15 subject and the field. So, right here, this is the quick find filter.
16 And when you search, it's showing you the results right here.
17 This is the message list. And then, when you select one of
18 these messages, it is shown down here in this preview pane.

19 JUDGE SAINDON: Counsel, I guess, maybe the
20 question is -- and if it's not, then I have the additional
21 question -- the incremental portion more than the search. Like,
22 okay, I buy this is a search. How do we know this is the
23 incremental one just like in the filtering dialogue box? Because
24 it just has a box. We don't know if it's picking up part of a
25 word or not.

1 MR. THACKER: You're saying this specific user
2 interface that's shown here?

3 JUDGE SAINDON: Figure 1.4, yeah. It says subject,
4 contains, and then box. How do we know that that is an
5 incremental search as opposed to a delimited search?

6 MR. THACKER: Because the delimited -- the Advanced
7 Find and Find searches are different user interfaces. And here,
8 I don't see, like, an enter button. All I see is a search field just
9 like we saw in those other figures. So let me go to the other --

10 JUDGE SAINDON: I suppose we could say your
11 position is by inference as opposed to anything explicit?

12 MR. THACKER: Right. I mean, for example, if we look
13 at Slide 45, we have the same sort of interface feature there.
14 And then if we go to Slide 46, again, we have that same exact
15 user interface feature.

16 JUDGE SAINDON: And this one does explicitly say as
17 you type, so.

18 MR. THACKER: Yeah.

19 JUDGE SAINDON: Okay.

20 MR. THACKER: And again, the Exhibit 1004 also says
21 that as well, not just 1043.

22 JUDGE MURRAY: Counsel, while we're talking about
23 the highlighting, I asked opposing Counsel, and I want to give
24 you a chance to answer as well. Which of the search functions
25 in Londergan or Wilcox do you rely on as teaching the

1 displaying and highlighting limitations of Claims 10 and 11 in
2 the '490 Patent?

3 MR. THACKER: Yeah. So, I think that Patent Owner
4 incorrectly stated that we only rely on Londergan for the
5 highlighting functionality. But actually on that same slide,
6 which I think was Slide 47 of Patent Owner's, that same slide
7 also shows a citation to Wilcox where Wilcox says that it uses a
8 red box to highlight search strings. And so -- so sorry. You
9 were talking about Entourage, and I was getting ahead of
10 myself and talking about Lotus Notes.

11 For Entourage, it is true that the only -- the express
12 disclosure of highlighting that we're relying on is in the Find
13 feature. But as Dr. Terveen explained, it would have been
14 obvious to a POSITA, given the known benefits of highlighting
15 and given the express disclosure of highlighting in the Find
16 tool, that you would want to provide the highlighting
17 functionality in an incremental search tool like the quick find
18 tool. So within, like, what's shown in 65, it would have been
19 obvious to take that functionality from the Find tool and
20 highlight matching search strings in the preview pane.

21 JUDGE MURRAY: Thank you. I appreciate that. My
22 question actually was about the Lotus Notes one, though, and
23 not so much whether it was Wilcox or Londergan, but which
24 search function? Is it Query Builder, Search Builder, the
25 Search Bar, or Forms? Which one teaches highlighting?

1 MR. THACKER: Yeah. So, in Wilcox and Londergan,
2 when you use either the Search Bar or the Query Builder or
3 Search Builder, and the search -- once the search is performed,
4 you get to that window which is shown in Wilcox, for example,
5 at page 165, where all the results are shown. And once you
6 click on one of those files, or select one of those files, then the
7 highlighting is shown within that document, and that's in both
8 Wilcox and Londergan. So that functionality is part of the
9 Search Builder tool and -- I'm sorry, the Query Builder tool and
10 the Search Bar.

11 JUDGE MURRAY: Okay. Thank you.

12 MR. THACKER: Okay. Next, I would like to turn to
13 Raskin, which is shown on Slide 26 and 27. Again, Patent
14 Owner states that Raskin only searches within files, but doesn't
15 address this page 132 in Raskin, where it's clearly disclosing
16 incrementally searching and then using an index to search
17 across files. And what Londergan -- or what Raskin discloses
18 is that this is beneficial, for example, if the user did not know
19 what document the target resided in or -- so it's saying you can
20 perform a specific search from the beginning which would find
21 the target. So that's what Raskin is talking about. It's saying,
22 sure, if you know what the document is, you can just open that
23 document, and you can search within it, but if you don't know
24 what the document is, then you want to be able to provide this
25 interfile search.

1 And again, we've relied on this disclosure in Raskin in
2 direct response to the POR's argument that you wouldn't have
3 been motivated to combine Raskin's teachings with Lotus Notes
4 because Raskin is only an interfile search. Again, Lotus Notes
5 already discloses an interfile search, so we didn't need Raskin
6 for that. What Raskin provides is incremental search. But if
7 they're going to dispute reasonable expectation of success and
8 motivation, well, Raskin discloses that you can perform an
9 incremental search across files.

10 JUDGE SAINDON: So, Counsel, these citations on
11 page, or Slide 26 of your demonstratives, from what I
12 remember -- and I don't remember exactly, the Raskin book
13 talks about incremental search, and it talks about these snippets
14 that you have here, but they're in different sections, and Raskin
15 is just hitting idea after idea. So I didn't take that this was in
16 any way connected to the incremental search, or am I
17 misremembering that?

18 MR. THACKER: Well, so, on page 132 of Raskin,
19 Raskin is discussing both delimited searches and incremental
20 searches. And here, at the very top of Raskin, it's saying
21 indexing can make searching over an entire system or network
22 take only milliseconds. So in the initial page that we relied on
23 in the petition, page 129, as well as page 132, it is reinforcing
24 that incrementally searching across documents is possible when
25 you use an index.

1 JUDGE SAINDON: So, do you dispute Patent Owner's
2 characterization that Raskin is just teaching a brute-force with
3 no indexing?

4 MR. THACKER: That's correct.

5 JUDGE SAINDON: Okay.

6 MR. THACKER: But, again, even if Raskin didn't, we
7 submit that this is improper bodily incorporation.

8 Okay. Next, I want to discuss indexing of partial search
9 strings. And for this, I want to turn to the actual claim
10 language. So let's look at Slide 5, which actually doesn't have
11 the exact claim language that it was. So the claim language
12 here is that it doesn't say an indexed data structure comprising
13 associations between partial search strings and associated files.
14 It says, between a plurality of character strings and associated
15 files. So, I don't see a requirement in the '490 Patent of
16 indexing partial search strings.

17 JUDGE MOORE: Counselor. In that regard, though,
18 don't both parties agree that incremental searching requires
19 pulling up lists as each character is entered into the field?

20 MR. THACKER: Incremental search requires displaying
21 results as the user types, correct.

22 JUDGE MOORE: Character by character? And didn't
23 Dr. Terveen testify that that type of search would require that
24 the index contain all the prefixes? The example he gave is
25 that -- the exact example, I believe it was the word
26 "declaration." You have to have D in the index. You have to

1 have DE in the index. You have to have DEC in the index,
2 DECL in the index.

3 MR. THACKER: Yeah. So that was -- that testimony
4 was in response to a question about specifically an index that
5 contains prefixes, like a prefix index, is my recollection of that
6 testimony, specifically, that was shown on that slide. And so he
7 was saying that, yeah, if you're indexing prefixes, then for the
8 word "declaration," you would have to, according to a prefix
9 index, index each prefix for that word.

10 But prefixing -- indexing prefixes is not per se required
11 by the claims, nor is it required to perform an incremental
12 search. As Dr. Terveen explained, again, Exhibit 1100,
13 paragraph 47, that one solution is to index the partial search
14 strings themselves. Another is to index only the full words and
15 join the rows for the corresponding words in the index that start
16 with the search string.

17 And speaking of prefixing, I'd like to discuss that really
18 quick. So both Londergan and Wilcox at -- so Londergan at
19 144, Wilcox at 159, which is cited by the petition at 21,
20 described using a full-text index. And this index will allow you
21 to find documents that contain various combinations of text
22 strings -- I'm reading from Wilcox, Exhibit 1059 at 159 -- will
23 allow you to find documents that contain various combinations
24 of text strings. A text string is any combination of characters
25 that may or may not be a word. And then similarly,
26 Wilcox -- Londergan at 161, cited by the petition at 22,

1 similarly discloses using an asterisk to perform a prefix search.
2 So, for example, if the query "hub" would find the words
3 "hubbub," "hubcap," and "hub."

4 Baeza-Yates, Exhibit 1023, also discloses using -- or,
5 prefix searches. So this is Baeza-Yates at 195, which is cited
6 throughout the petition. And it's discussing on page 195 an
7 inverted index, searching using an inverted index. And then it
8 says, prefix and range queries can also be solved with binary
9 search trees or B-trees. So what it's saying is, you have an
10 inverted index, you use these specific binary searches, trees, or
11 B-trees to quickly be able to find prefixes on that inverted
12 index.

13 And then finally, Green, at Column 1, lines 38 to 43,
14 expressly discloses, quote, a complete indexing of characters in
15 each word permits very fast, exact match in prefix searches. So
16 we have several references disclosing that you can use an index
17 and you can use certain techniques to be able to find
18 prefix -- you can be able to solve prefix search queries very
19 quickly.

20 All right. So now, I would like to -- one other thing that I
21 wanted to make sure I address is that Patent Owner said that
22 they were disputing whether Spaey and Nguyen teach
23 incrementally searching across files. And Haeberli's
24 declaration, all he says about Spaey and Nguyen is that they
25 were cited during prosecution. He doesn't say, oh, they don't
26 teach incrementally searching across files. And in fact, Spaey

1 was cited during prosecution. And at the time, Applicant didn't
2 try and distinguish Spaey based on searching across files.
3 What -- how Applicant distinguished Spaey was based on the
4 fact that it had different search fields for different search
5 strings.

6 So with that, I'd like to turn to the actual background
7 references that disclose incrementally searching across files.
8 So, this is on Slide 30, this is Spaey, and it's showing searching
9 for different CD information here. And then on Slide 31,
10 Nguyen is disclosing searching across different help articles for
11 problems that happen in network printers. Again, it's searching
12 across help articles. And then, Siitonen -- again this is Slide
13 32 -- is disclosing searching across contacts on a PDA.

14 And this gets to -- I want to talk about Entourage next,
15 and its search across files. So it appears that Patent Owner does
16 not dispute that what it's searching, what Entourage is
17 searching, is files or documents. And what it -- it's trying to
18 distinguish a filter from a search. But the patent refers to
19 searching and filtering interchangeably. Its figures, such as
20 Figures 3G and 3H, showing very similar user interface that
21 works the same way as Entourage, and I want to show you that
22 right now. Let's look at the patent. First of all, I want to show
23 the patent language here which is referring to searching and
24 filtering interchangeably. But I want to direct your attention to
25 their Figure 3G, because what's happening here is that it's
26 showing you a list, and then as you type in another character,

1 shown on 3H, that list shortens. And just like, you know, in
2 Entourage, it's showing a preview of a selected
3 document -- here it's the first document -- but it's not showing
4 you the whole content of every document. There's nothing in
5 the claims or the description that say you have to search
6 through all the content of all documents. What it's showing
7 here is that there are specific fields for entering your search
8 strings. So you can search within the subject or the sender or
9 recipient fields, just like in Entourage.

10 So continuing with Ground 2, I'd like to talk about
11 the -- what we're relying on the Find tool for. This goes to
12 Judge Murray's question, but Patent Owner's Counsel said that
13 we're relying on the quick find, the Find, and the Advanced
14 Find. Now, what we pointed to in the Advanced Find in the
15 petition was merely the fact that the Advanced Find suggests
16 entering multiple search strings. But as we explained in the
17 petition, we're relying on True for its disclosure of searching for
18 multiple search strings entered in a single search field. And so,
19 the quick find tool does satisfy all elements of the independent
20 claims, which is what I stated in my opening. Now, the Find
21 tool, we're relying on for certain dependent claims, for example,
22 Claim 11 in the '490 Patent, but for that feature only.

23 JUDGE MOORE: Just one point of clarity. Patent
24 Owner said several times that Entourage is not an indexed
25 system. Does Petitioner dispute that point?

1 MR. THACKER: Petitioner acknowledges that
2 Entourage never expressly discloses using an index, but as
3 explained by Dr. Terveen in his initial declaration, a skilled
4 artisan would understand when implementing Entourage's
5 teachings that you would want to use an index. For example,
6 Baeza-Yates discloses that an index is at the core of every
7 modern information retrieval system. It doesn't treat indexes as
8 optional. These are -- and a skilled artisan would understand
9 that you would want to use an index to make sure that the
10 search is happening quickly. So we did rely on Baeza-Yates,
11 because Baeza-Yates expressly discloses this use of an index.
12 But a skilled artisan -- we submit that a skilled artisan would
13 not understand that Entourage necessarily does not use an
14 index. It's just silent.

15 JUDGE MOORE: Okay. Thank you.

16 MR. THACKER: Okay. So next, I want to address an
17 argument that somehow we're relying on True for something we
18 didn't rely on in the papers, which I don't understand that
19 argument. But to the extent I said something in the opening to
20 the contrary, I want to be clear that we're relying on True for its
21 disclosure to search for multiple search strings. Now, the fact
22 that it also discloses that in the context of an incremental
23 search, just like Entourage, speaks to their natural
24 combinability. And so, sure, both references disclose
25 incrementally searching. That only helps the obviousness
26 showing. But what we are relying on True for is this disclosure

1 to incrementally search for multiple search strings. And that
2 was explained in the petition. It was explained in the reply.

3 And I will invite the Board to Slide 42, for example,
4 which shows an excerpt of the petition. And it says that we're
5 relying on Entourage for its teachings to incrementally locate
6 files containing at least one search string and True's teachings
7 to do so for multiple search strings.

8 JUDGE MOORE: What do you say to Patent Owner's
9 argument that True doesn't search for a search string, it implies
10 a different type of fuzzy search, for lack of a better term, that's
11 different from what the claims recite?

12 MR. THACKER: Are you asking specifically about the
13 confidence levels?

14 JUDGE MOORE: Yes.

15 MR. THACKER: So, Patent Owner brought this up in its
16 Patent Owner's Response. And then in response to that, I
17 questioned their expert about True at deposition. And then we
18 cited the same portion of True that I questioned their expert
19 about on this point, and he admitted that True discloses
20 searching for search strings that have -- or performing a search
21 on text that has a confidence level of 1, or 100 percent accurate
22 text.

23 And I will invite the Board to Slide 98. Here --

24 JUDGE MOORE: Okay. Thank you.

25 MR. THACKER: Okay. Sorry, I just needed a second to
26 make sure that I covered everything. Hold on one second.

1 So in the event that Patent Owner addresses -- in its
2 rebuttal attempts to argue that -- I'm speaking to Ground 1 here
3 and reliance on Raskin, page 132. So to the extent that Patent
4 Owner gets up here and argues that this was an improper new
5 argument and shouldn't be considered, we have several reasons
6 why that's wrong.

7 So first, the petition cited to this page in Raskin. The
8 petition at page 16 cites to Raskin when explaining that Raskin
9 is analogous art to the '490 Patent because it teaches locating
10 files, plural, containing the search strings. So Patent Owner
11 was on notice. Second, this page in Raskin was raised in direct
12 response to this unforeseeable argument that the claims
13 somehow require searching across a large volume of files, and
14 so I'd invite the Board to the cases we cited, such as *Axonics*
15 and *Rembrandt*, on that point.

16 And most importantly, reliance on this page in Raskin is
17 not a meaningfully distinct unpatentability argument. We're
18 maintaining the exact same combination. All we're doing is
19 pointing to this additional evidence that was already cited in the
20 petition to rebut this notion that Raskin's incremental search
21 couldn't be applied across files.

22 And as we explained in the introduction of our reply, the
23 arguments from Patent Owner have changed over the course of
24 time. And during prosecution, they attempted to distinguish art
25 on the basis that it incrementally searched only within separate
26 search fields, and it didn't allow for multiple search strings in a

1 single search field. And then, in response to our art that clearly
2 discloses this well-known ability to enter multiple search
3 strings in a single search field, it's now in the POR and
4 sur-reply raised for the first time, this no reasonable expectation
5 of success because you couldn't search across -- incrementally
6 search across a large volume of documents, argument.

7 And we've pointed to numerous exhibits, and Dr.
8 Terveen cited a plethora of background references in his
9 declaration explaining that a POSITA would have reasonably
10 expected success in implementing incremental search across
11 documents, as evidenced by our primary reference, Entourage,
12 for example, and as evidenced by Raskin, for example. If the
13 Board has no further questions, I will cede my remaining time.

14 JUDGE MURRAY: None here.

15 JUDGE MOORE: Nothing here.

16 JUDGE SAINDON: Thank you.

17 MS. LAROCHELLE: I'm going to try to do this without
18 my computer, but I might point to slides so it's in the record.
19 But I'm going to see if I can just point you to the documents in
20 the record.

21 I'm going to make sure I have my patents with me. So
22 thank you, again. Christie Larochelle for Patent Owner. I want
23 to return to this issue of the list filtering. And my
24 understanding, I think, that what Petitioner is referring to every
25 time my colleague here says quick find, I think what they're
26 referring to is the list filtering. To my knowledge, the

1 references never call it quick find, so I'm proceeding on the
2 understanding that what they're referring to is the list filtering
3 tool.

4 And what I fear my opposing Counsel is trying to do here
5 is conflate what the user interface looks like with what's going
6 on in the background. And so, you saw him; he put up the
7 figure from the patent, and he tried to say, well, this looks like
8 the Entourage user interface. And just because the user
9 interfaces might look similar, and I'm not conceding that they
10 do without looking at them side by side, but that doesn't mean
11 the same thing is going on in the background.

12 And so what's happening in Entourage, you have these
13 two different tools. You have the Find and the accompanying
14 Advanced Find, and you have the list filtering. And the Find
15 tool, the references tell us that the reason we have these two
16 separate tools is because the Find tool is going to be slow,
17 right? It's delimited and it's going to take a long time, because
18 you're searching, going out there, searching for and locating
19 documents.

20 The list filtering is a lightweight tool that allows you to
21 simply filter over attributes. And so what you're looking at in
22 that user interface is a pre-populated list of basically rows. Like
23 I said before, it's kind of like its own document, it's like an
24 Excel spreadsheet. So you basically have rows that have the
25 attribute columns, and so what you're able to do is simply filter
26 out the ones that don't match your search string. So you're not

1 going out and locating documents. You're not searching for and
2 returning documents as search results. You're simply taking a
3 list that already exists and filtering out ones that maybe the
4 name field doesn't match your search string. But the bottom
5 line is, what's going on in the background is different in these
6 two user interfaces. What the patent describes is literally going
7 out, locating documents, returning them into this search results
8 pane.

9 I want to point out a couple of other things that I just saw
10 on rebuttal here. Petitioner, again, showed Slide 46, which has
11 an image from Exhibit 1043 that they say is from the Entourage
12 references. That's not one of their primary references for the
13 ground. I don't know why they're showing this reference and
14 this figure that was never shown in the petition, and it's not one
15 of their primary references. So Exhibit 1043 is not one of the
16 primary references for Ground 2 and Slide 46 should be
17 disregarded. That is not one of their Entourage primary
18 references.

19 I also want to respond that we are not holding them to a
20 higher standard of obviousness. We're simply asking that they
21 be held to the regular standard of obviousness. And as Your
22 Honors know, even under the flexible approach of *KSR*, the
23 Petitioner still needs to identify where each element is disclosed
24 in the prior art, and why a POSITA would be motivated to
25 combine them with a reasonable expectation of success. And
26 that is what our argument is here. We're not holding them to a

1 higher standard. We're holding them to this standard. They
2 have failed to do this. They have failed to identify where each
3 element is disclosed in the art and why a POSITA would be
4 motivated to combine those with a reasonable expectation of
5 success.

6 Another thing that Petitioner said on rebuttal is that the
7 two-document requirement that we said in our presentation,
8 they said that's absolutely false. And I want to point you to the
9 '139 Petition, pages 4 to 5, where the Petitioner says the claims
10 recite searching among a set of, i.e., two or more documents,
11 not the large-volume set alleged by the POR. The same
12 sentence, except with slightly different language to
13 accommodate the claim language, appears in each of the other
14 petitions, the '035 Petition at pages 4 to 5 and the '490 Petition
15 at page 4. The claims recite searching among a plurality of, i.e.,
16 two or more files, not the large-volume set alleged by the POR.

17 JUDGE MURRAY: Counsel, is that in the petition or in
18 the reply?

19 MS. LAROCHELLE: That's in the -- oh, I'm sorry. That
20 is in the reply. You are absolutely correct. So that's in the
21 reply for each of the three --

22 JUDGE MURRAY: Okay. Thank you.

23 MS. LAROCHELLE: Apologies for the mistake.

24 Petitioner also showed you Slides 26 and 27 again. And
25 I just want to emphasize that never in the petition or in the reply
26 did Petitioner argue that Raskin searches combined incremental

1 search in multiple files. And Petitioner spent a long time
2 talking about page 132 in his rebuttal, and I encourage you to
3 go to page 132 and read the context. On page 132, Raskin is
4 still talking about a sequential search of a single open
5 document. This is not -- the mention of indexing in a footnote
6 doesn't mean that Raskin is talking about indexing in the
7 context of incremental search.

8 One other thing. I think there were some questions about
9 Figure 1.4 on Petitioner's Slide 65 from Exhibit 1004. This is
10 cited in the petition only for the displaying claims. It is not
11 cited anywhere, we can't find it anywhere in the petition, for the
12 proposition that the search using that interface is incremental.
13 Again, I think that we all can agree that Petitioner is inferring
14 that it was incremental, but we don't concede. We don't think
15 there's any disclosure that says that that's an incremental
16 search. And if it were, it would be a list filtering and not a
17 search.

18 And then, oh, one last thing. For the knowledge of a
19 POSITA, Petitioner cites the *Shockwave* case. And I just want
20 to point out that in *Shockwave* the issue there was about using
21 applicant-admitted prior art as background references, and that's
22 not the situation here. They have gone out and they have these
23 prior art disclosures that are not thoroughly -- they're not
24 actually used in the petition itself. Co-counsel noted that Dr.
25 Terveen discussed them in his background state-of-the-art
26 section, but never does Dr. Terveen or Petitioner point to, with

1 any specificity, exactly what they're relying on for those -- from
2 those references in the grounds as they attempt to make these
3 combinations. So, that is a failure of proof.

4 If you give me just a second. Unless Your Honors have
5 questions, I will cede the rest of my time.

6 JUDGE MURRAY: None here.

7 JUDGE MOORE: None here.

8 JUDGE SAINDON: Not seeing any.

9 MS. LAROCHELLE: Thank you all.

10 JUDGE SAINDON: All right. Thank you. Okay. The
11 Board thanks the parties for their arguments today. That was
12 very helpful. The case is submitted. We are adjourned.

13 (Whereupon, the above-entitled matter went off the
14 record at 4:25 p.m.)

IPR2025-00253 (Patent 7,370,035 B2)
IPR2025-00254 (Patent 10,552,490 B2)
IPR2025-00255 (Patent 9,633,139 B2)

PETITIONER:

Todd Siegel
Amy Haspel
Andrew Mason
Samuel Thacker
KLARQUIST SPARKMAN, LLP
todd.siegel@klarquist.com
amy.haspel@klarquist.com
andrew.mason@klarquist.com
samuel.thacker@klarquist.com

PATENT OWNER:

Lori Gordon
Christie Laroche
Sanjeet Dutta
GOODWIN PROCTER LLP
gordon-ptab@goodwinlaw.com
clarochelle@goodwinlaw.com
sdutta@goodwinlaw.com