"native interface" and "native system services"

Darwin is accessible via a procedural native interface and [ ] accessed via procedural function calls. Spielman Tr. 2109-11; CX-803 at APPHTC_00001901; CX-4379C at APPHTC-S_00001203-04; CX-4374C at APPHTC-S_00001090-91. Moreover, Darwin [ ] Spielman Tr. 2112; CX-803 at APPHTC_00001901.

The third and fourth elements of apparatus claim 1 recite:

> **object oriented methods requiring native system services;**
>
> **procedural program logic code, responsive to invocations of the object-oriented methods during runtime, for causing the procedural operating system to control the computer hardware to perform the required native system services;**

The second element of method claim 7 recites:

> **issuing calls during runtime, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services;**

Apple has satisfied this claim element.

"object oriented methods requiring native system services" (claim 1) and "object-oriented methods requiring such native system services" (claim 7)

It is undisputed that the MacBook Pro includes [ ] which are object-oriented methods requiring native system services. At the hearing, Ms. Spielman detailed [ ] Spielman

2112-17; CX-4363C; CX-4386C; CX-4387C; CX-4388C; CX-4473C; CX-4379C]

"procedural program logic code" (claim 1) and "issuing calls during runtime, compatible with the native interface" (claim 7)

It is undisputed that the MacBook Pro includes procedural program logic code, responsive to invocations of object-oriented methods during runtime, to control the computer hardware to provide native system services. Spielman Tr. 2115-17; CX-4473C at APPHTC-S_00013309; CX-4379C at APPHTC-S_00001203. Specifically, [

] *Id.*

The fifth and sixth elements of apparatus claim 1 recite:

> **executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code;**
>
> **means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory; and**

The third element of method claim 7 recites:

> **determining during runtime if object-oriented methods to be invoked during runtime execution are present in executable program memory associated with the computer hardware; and**

"executable program memory"

It is undisputed that the MacBook Pro includes executable program memory for runtime execution of the Darwin operating system, invocations of Java methods, and related portions of procedural program logic code. Spielman Tr. 2117-18. Specifically, the MacBook Pro provides executable application memory as well as operating system memory. Spielman Tr. 2118; CX-803 at APPHTC_00001894.

> **"means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory"**

It is not disputed that the MacBook Pro makes determinations whether object-oriented [ ] methods to be invoked are present in executable program memory. Spielman Tr. 2118-19. Specifically, the MacBook Pro includes the agreed structure of a CPU that will perform step 308 of FIG. 3 in the '983 patent. Spielman Tr. 2118-19; JX-4, FIGS. 1 and 3. Specifically, the MacBook Pro includes [ ] that checks to see if the [ ] methods are present in executable program memory. Spielman Tr. 2119-20; CX-802 at APPHTC_00001869; CX-4468C at APPHTC-S_00013399. At the hearing, Ms. Spielman described [ ] Spielman Tr. 2119-22; CX-4383C; CX-4390C; CX-4391C; CX-4394C; CX-4467C; CX-4468C.

The last element of apparatus claim 1 recites:

> **a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.**

The last element of method claim 7 recites:

> **selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded.**

Apple has not satisfied these claim elements which require selective loading. Apple has failed to prove that the MacBook Pro running Snow Leopard practices these limitations for the same reason that Apple was unable to demonstrate that HTC infringes these elements. Apple contends that "MacBook Pro includes a runtime loader that selectively loads required object-oriented [ ] methods into executable program memory during runtime, before invocation thereof." Apple Br. at 88. Apple explains that [

] Id. at 88-89.

Thus, as Apple admits, [ ] Indeed, Apple's expert Ms. Spielman admits that [

]

Spielman Tr. 2121-22, 2124-25 [

] *see* Jeffay Tr. 3377-78 [

] CX-4383C at APPTHTC-S_00001281-82 (classLoader.cpp). Thus, this loading is not selective with regard to ***methods*** to be invoked as the claims require.

Accordingly, Apple has failed to show that it practices the "to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods" limitation of claim 1, or the "selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded" limitation of claim 7.

In summary, Apple has not satisfied the technical prong of the domestic industry requirement with respect to the '983 patent.

D. Validity

HTC states that it "does not contend that the '983 patent is invalid under HTC's and the Staff's constructions, which require selectively loading ***methods***. HTC's invalidity theories are advanced solely under Apple's constructions, which remove the limitation of '***selectively*** loading the object-oriented ***methods*'.**" HTC Reply at 59 n.31. The Staff agrees that HTC only argued invalidity of the '983 patent only under Apple's proposed claim constructions. Staff Br. at 57-58.

The undersigned agreed with HTC and the Staff that the proper construction of the last elements of claims 1 and 7 require selective loading of *methods*, which is different from selective loading of *classes*. Thus, HTC's contentions regarding invalidity of the '983 patent is no longer in play.

Accordingly, HTC has not shown by clear and convincing evidence that the NeXTSTEP Release 3 System anticipates asserted claims 1 and 7 of the '983 patent. Further, HTC has not shown by clear and convincing evidence that the combination of Vernon and Gautron references renders obvious asserted claims 1 and 7 of the '983 patent.

IX. U.S. Patent No. 5,946,647

The '647 patent is entitled, "System and Method For Performing An Action On A Structure In Computer-Generated Data." The Summary of the Invention states, in part:

> The present invention overcomes the limitations and deficiencies of previous systems with a system that identifies structures in computer data, associates candidate actions with each detected structure, enables the selection

> of an action, and automatically performs the selected action on the identified structure. It will be appreciated that the system may operate on recognizable patterns for text, pictures, tables, graphs, voice, etc. So long as a pattern is recognizable, the system will operate on it. The present invention has significant advantages over previous systems, in that the present system may incorporate an open-ended number and type of recognizable patterns, an open-ended number and type of pattern analysis units, and further that the systems may enable an open-ended number and type (i.e. scripts, macros, code fragments, etc.) of candidate actions to associate with, and thus perform, on each identified structure.

JX-3 (Summary of the Invention) at col. 2, lns. 4-20.

Apple asserts apparatus claims 1, 3, and 8 and method claims 15 and 19. Claims 3 and 8 depend on independent claim 1 and claim 19 depends on independent claim 15. The asserted claims read as follow:

> **1**. A computer-based system for detecting structures in data and performing actions on detected structures, comprising:
>
> an input device for receiving data;
>
> an output device for presenting the data;
>
> a memory storing information including program routines including
>
> an analyzer server for detecting structures in the data, and for linking actions to the detected structures;
>
> a user interface enabling the selection of a detected structure and a linked action; and
>
> an action processor for performing the selected action linked to the selected structure; and
>
> a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines.

> **3**. The system recited in claim **1**, wherein the input device receives the data from an application running concurrently, and wherein the program routines stored in memory further comprise an application program interface for communicating with the application.
>
> **8**. The system recited in claim **1**, wherein the user interface highlights detected structures.
>
> **15**. In a computer having a memory storing actions, a method for causing the computer to perform an action on a structure identified in computer data, comprising the steps of:
>
> receiving computer data;
>
> detecting a structure in the data;
>
> linking at least one action to the detected structure;
>
> enabling selection of the structure and a linked action; and
>
> executing the selected action linked to the selected structure.
>
> **19**. The method recited in claim **15**, wherein the memory contains strings, and wherein the step of detecting a structure further comprises the steps of retrieving a string from the memory and scanning the data to identify the string.

JX-3 at col. 7, lns. 8-24, 27-32, 50-51; col. 8, lns. 22-33, 47-50.

A. Claim Construction[34]

The parties agree to the meaning of "detecting" / "detected," "structure," "analyzer server," and "application running concurrently."[35] The parties dispute the

[34] With respect to the '647 patent, a person of ordinary skill in the art has at least a B.S. degree in computer science (or equivalent coursework) and two to three years of academic or work experience in the field. Mowry Tr. 2459-62; Olsen Tr. 3839-3841; Staff Br. at 60 n.22.

[35] The parties agree that (1) detecting" / "detected" means "finding and identifying" /

meaning of "linking actions to the detected structures" / "linking at least one action to the detected structure" and "input device." These disputed terms are discussed below.

1. "linking actions to the detected structures" and "linking at least one action to the detected structure"

Claim Term	Apple Construction	HTC and Staff Construction
linking actions to the detected structures (claim 1)	linking detected structures to computer subroutines that cause the CPU to perform a sequence of operations on the particular structures to which they are linked	linking a detected structure to a computer subroutine that causes the CPU to perform a sequence of operations on that particular structure to which it is linked, rather than an informational structure
linking at least one action to the detected structure (claim 15)	linking a detected structure to at least one computer subroutine that cause the CPU to perform a sequence of operations on the particular structure to which it is linked	linking a detected structure to a computer subroutine that causes the CPU to perform a sequence of operations on that particular structure to which it is linked, rather than an informational structure

Joint Claim Construction, App'x A at 16.

The parties' constructions differ in two respects. First, HTC's and the Staff's proposals add the phrase "rather than an informational structure." Second, HTC and the Staff contend that claim 1's "linking actions" term does not require multiple linked actions.

"found and identified;" (2) "structure" means "an instance of a pattern, where a 'pattern' refers to data, such as grammar, regular expression, string, etc., used by a pattern analysis unit to recognize information in a document such as dates, addresses, phone numbers, etc.; (3) "analyzer server" means a program sub-routine that receives data from a document having recognizable structures, and uses patterns to detect the structures;" and (4) "application running concurrently" means "application running during the same run-time." Joint Claim Construction, App'x A at 16; *see* Mowry Tr. 2490; Olsen Tr. 3883-84.

As proposed by Apple, the claim term "linking actions to the detected structures" of apparatus claim 1 is construed to mean "linking detected structures to computer subroutines that cause the CPU to perform a sequence of operations on the particular structures to which they are linked." Similarly, the claim term "linking at least one action to the detected structure" of method claim 15 is construed to mean "linking a detected structure to at least one computer subroutine that causes the CPU to perform a sequence of operations on the particular structure to which it is linked."

During prosecution, the Examiner rejected the pending claims as obvious based on the Sobotka reference in in combination with U.S. Patent No. 5,247,437 ("Vale"). The Examiner contended that Vale disclosed "linking between DIN and HN structure." JX-9 at APPHTC_00338339-40 ('647 File History, Dec. 6, 1998 Office Action at 2-3). In response, the Applicants argued:

> The linked actions of the claimed invention are patentably distinguished from the heading node (HN) structure of *Vale*. The linked actions enable execution of an action, which is a computer subroutine causing a CPU to perform a sequence of operations. Additionally, in the claimed invention, "[a]n action may specify opening another application, loading the identified structure into an appropriate field, and closing the application. An action may further include internal actions… and external actions…." Thus the linked actions can cure deficiencies of prior systems employing laborious and disruptive processes.
>
> In contrast, the HN structure of *Vale* consists of heading nodes, each of which includes the title of its associated index entry and defines information for one of the headings listed in the heading column of a keyword list. Each heading node also stores heading string, a sort string, a see string, and a heading ID. The HN structure is thus used to delineate the structural relationship of the key words or headings for a given index, but cannot cause a CPU to

> perform an operation, or an action, as can the claimed invention.
>
> In brief, *Sobotka* does not teach or suggest *linking a structure to an action. Vale* discloses *linking to an informational structure*, but that does not cure the references' lack of *linking to an action.* Therefore, the claimed invention, which recites *linking to an action*, or as claimed "linking actions to the detected structures," is patentably distinguished from *Sobotka* and *Vale*, either alone or in combination.

JX-9 at APPHTC_00338619-20 ('647 File History, Mar. 15, 1999 Amendment at 8-9) (citations omitted) (emphasis in original).

Applicants distinguished Vale because it does not disclose linking to an action. Rather, Vale describes linking ***only*** to an informational structure, the "HN" or "heading node," which represents a type of index. Importantly, the heading node is not an action, does not contain an action, and does not link to an action. Mowry Tr. 2470-71 ("Vale does not link to an action"), Tr. 2472 (Heading nodes represent information in an index.), Tr. 2474-75 (Heading nodes do not lead to an action and, in Vale, there are no actions, *i.e.*, operations that are performed on a detected structure, because there are no detected structures.); JX-9 at APPHTC_00338620 ("Vale discloses linking to an informational structure, but that does not cure the references' lack of linking to an action.").

Accordingly, the Applicants did not disclaim use of informational structures. Rather, they emphasized that there must be a link to an action. Mowry Tr. 2471-72, 2661-62 (no disclaimer). Apple's proposed construction, which does not include the phrase "rather than an informational structure," more accurately defines the "linking actions" claim terms.[36]

Moreover, the phrase "rather than an informational structure" is unnecessary. Indeed, in its opening statement at trial, HTC based its non-infringement case on a single contention – *i.e.,* that the HTC devices "link[] to an informational structure," and ***not*** to an action. In advancing this argument, HTC counsel stated, "Vale discloses linking to an informational structure. The claimed invention links to an action. Our devices work like Vale, not like the '647." Tr. 169. Later, HTC abandoned this distinction by agreeing that the phrase "rather than an informational structure" adds no meaning. Olsen Tr. 4030 (conceding that HTC accused products allow a user to select from a "list of possible actions"), Tr. 3836-37 & 4023-24 (admitting that Apple's and HTC's constructions "are essentially the same.").[37]

Additionally, Apple's constructions for "linking actions ..." (claim 1) and "linking at least one action..." (claim 15) differ in terms of the number of actions that must be linked to a detected structure. For example, Apple's construction for "linking actions..." in claim 1 requires that multiple actions be linked to a detected structure. HTC and the Staff propose that both phrases be construed in the exact same way such that there need only be one action linked to a detected structure. The intrinsic evidence

[36] The Staff argues that in the above Amendment, "the Applicant made clear that the required linking was to actions [ ] rather than to 'informational structures,' such as the indexes present in Vale." Staff Br. at 62. The Staff is incorrect for the reasons discussed above.

[37] In fact, as noted, HTC states that "only one of the remaining claim construction issues is outcome determinative," *i.e.*, "whether the term 'input device' can include software only or must include some form of hardware." HTC Br. at 104. (HTC did not even brief the present claim limitation.) *See* HTC Reply at 74 (briefing only the "input device" limitation for claim construction).

supports Apple's construction.

The language of claim 1 requires "'linking actions," the word "actions" being plural, to detected structures. *See* Mowry Tr. 2477. In contrast, claim 15 requires linking "at least one action" to a detected structure. *Id.* Apple's constructions reflect that clear distinction.[38] Further, the Summary of the Invention describes "candidate actions" (plural) available "[u]pon selection of a detected structure" (singular). JX-3 col. 2, lns. 42-62; *see* Mowry Tr. 2477-78. Dr. Olsen (HTC's expert) did not offer any opinion to the contrary. Olsen Tr. 3837 (referring to "some quibble about plurals" between the parties but offering no opinion in support of HTC's construction).

2. "input device"

Claim Term	Apple Construction	HTC and Staff Construction
input device	computer software or hardware	plain and ordinary meaning (hardware only)

Joint Claim Construction, App'x. A at 16.

As proposed by HTC and the Staff, the claim term "input device" is given its plain and ordinary meaning, *i.e.*, "computer hardware but not computer software."

The specification of the '647 patent discloses the following regarding "input device":

> Referring now to FIG. 1, a block diagram is shown of a computer system 100 including a CPU 120. Computer system 100 is preferably a microprocessor-based computer,

[38] *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed.Cir.2005) ("Differences among claims can also be a useful guide in understanding the meaning of particular claim terms."); *see also Voda v. Cordis Corp.*, 536 F.3d 1311, 1319-1320 (Fed. Cir. 2008) (considering differences in independent claim language in determining scope of claims).

> such as a Power Macintosh manufactured by Apple Computer, Inc. of Cupertino, Calif. An input device 110, such as a keyboard and mouse, and an output device 105, such as a CRT or voice module, are coupled to CPU 120. ROM 155, RAM 170 and disk storage 175 are coupled to CPU 120 via signal bus 115. Computer system 100 optionally further comprises a printer 180, a communications interface 185, and a floppy disk drive 190, each coupled to CPU 120 via signal bus 115.

JX-3 at col. 3, lns. 22-33 (emphasis added); *see* FIG. 1 (clear denotation of hardware versus software elements).

Thus, the specification shows that "input device" includes hardware that receives input, such as a keyboard and mouse.[39] Indeed, there is nothing in the '647 specification that suggests that the Applicants were importing a special meaning into the term "input device." The disclosure of "input device" in the specification is entirely consistent with the plain and ordinary meaning of the term.

Apple seeks to expand the plain and ordinary meaning of "input device" to include any software that can receive data — regardless of whether hardware is present. Mowry Tr. 2465-66. Dr. Mowry does not dispute that an "input device" can be hardware that receives data. *Id.* at 2467 ("I agree that a hardware input device is an input device."). However, Dr. Mowry then expands such a meaning to conclude essentially that if something (whether it be a program or any software-based application) receives data it must be an "input device." *Id.* at 2466-67. Dr. Mowry reaches this conclusion by asserting that "programs pass information from one program to another. When this occurs, the mechanism that passes the information is software." *Id.* at 2467.

[39] Additionally, Dr. Olsen testified that "input device" includes other hardware such as an input buffer or a touchscreen. Olsen Tr. 4037-38 (describing various portions of the '647 patent and specification in which input devices are physical hardware).

This testimony is far from being convincing. As the Staff notes, simply because software routines pass information between them does not transform them into input devices. Moreover, there is no evidence that anyone in the art would refer to "software" as a device. Indeed, claim 1 of the '647 patent requires that the processing unit be "coupled to" the input device. This requirement would not make much sense if the input device was software running on the processing unit.

Apple's construction runs contrary to the plain meaning of the term "input device" and the intrinsic record and is, therefore, rejected.

B. Infringement

For the reasons set forth below, Apple has shown that HTC's accused products infringe the asserted claims 1, 8, 15, and 19 of the '647 patent. However, Apple failed to show infringement with respect to claim 3.

1. Independent claims 1 and 15

The preamble of independent apparatus claim 1 recites:

> **A computer-based system for detecting structures in data and performing actions on detected structures, comprising:**

The preamble of independent method claim 15 recites:

> **In a computer having a memory storing actions, a method for causing the computer to perform an action on a structure identified in computer data, comprising the steps of:**

Apple has satisfied the preambles of claims 1 and 15. The parties agree that the preambles of claims 1 and 15 are not limitations because they do not give "life, meaning and vitality to the claim[s]." *Altiris, Inc. v. Symantic Corp.,* 318 F.3d 1363, 1371 (Fed.

Cir. 2003); Mowry Tr. 2480. In any event, HTC accused products satisfy the preambles because they are computer-based systems for detecting structures in data and performing actions on detected structures (claim 1) and are computers having memory storing actions that perform an action on a structure identified in computer data (claim 15). Mowry Tr. 2480-81.

The first element of independent apparatus claim 1 recites:

> **an input device for receiving data;**

The first element of independent method claim 15 recites:

> **receiving computer data;**

Apple has satisfied these claim elements. There is no dispute that HTC accused products contain a hardware input device for receiving data and a method of receiving computer data. Mowry Tr. 2481-82; Olsen Tr. 4019-20. These products contain hardware input devices such as wireless-internet adapters for receiving internet data, radios for receiving text messages, a touchscreen, and memory. Mowry Tr. 2482; CPX-3; CPX-4; CPX-5.

The second element of claim 1 recites:

> **an output device for presenting the data;**

Apple has satisfied this claim element. The HTC accused products contain an output device for presenting the data in the form of a touchscreen display. Mowry Tr. 2487; CPX-3; CPX-4; CPX-5.

The third element of claim 1 recites:

> **a memory storing information including program**

routines including

Apple has satisfied this claim element. As confirmed by HTC's user guides and an analysis of the physical devices, the accused products contain memory storing information including program routines. Mowry Tr. 2488-90; CPX-3; CPX-4; CPX-5; CX-391; CX-3510 (Droid Incredible User Guide) at 301.

The fourth element of independent apparatus claim 1 recites:

> **an analyzer server for detecting structures in the data,**
> **and for linking actions to the detected structures;**

The second and third elements of independent method claim 15 recite:

> **detecting a structure in the data;**
>
> **linking at least one action to the detected structure;**

Apple has satisfied these claim elements. As seen by the plain language of the claims, the fourth element of apparatus claim 1 coincides with the second and third elements of method claim 15. Thus, claim 1 has two requirements. First, the analyzer server must detect structures in the data. Second, the same analyzer server must link actions to the detected structures.

As an initial matter, the agreed-upon construction of "structure" requires, among other things, an "instance of a pattern," which is a "positive match of a pattern to something in a document." Mowry Tr. 2491. Examples of recognizable structures having semantic significance include dates, addresses, phone numbers and names. *Id.* 2492-93; JX-3 at col. 1, lns. 14-16.

"analyzer server for detecting structures in the data" (claim 1) and "detecting a structure in the data" (claim 15)

As to the first portion of this claim element, *i.e.*, "detecting structures in the data," HTC concedes that Browser, Android Messaging, and HTC Messages detect structures in data. Olsen Tr. 4020-21; Mowry Tr. 2490. The Browser detects e-mail addresses, phone numbers, and postal addresses, and Android Messaging and HTC Messages detect e-mail addresses and phone numbers. Mowry Tr. 2496, 2499, 2506.

[

]

Android Messaging and HTC Messages [

]

[

]

"analyzer server for ... linking actions to the detected structures" (claim 1) and "linking at least one action to the detected structure" (claim 15)

With respect to the second portion of this claim element, *i.e.*, "linking actions to the detected structures," Browser, Android Messaging, and HTC Messages infringe these limitations. Mowry Tr. 2515-16.

Generally, Browser and Android Messaging [

]

[

]

HTC Messages [

]

[

]

[

]

In other words, [

]

The Browser's analyzer server and method for linking actions includes [

]

[

]

[

[40]

]

[

]

[

[40] [

]

]

HTC contends that Apple has failed to prove infringement because Apple has implicitly conceded that the pointers described in the '647 patent differ fundamentally from HTC's accused Android devices. HTC Br. at 101-104.

HTC states in its brief that it would apply "Apple's proposed claim constructions for purposes of HTC's non-infringement defense." HTC Br. at 104. But, contrary to this representation, HTC rejects Apple's construction in favor of a new one that implies that "linking" must occur through the use of pointers. *Id.* at 103-104. This new construction—improperly raised in the post-trial "background" section—is rejected as untimely under Ground Rule 4(c) because it was not in HTC's pre-hearing statement.

In any event, the unrebutted testimony shows that the plain meaning and proper construction of "linking" is "associating." Mowry Tr. 2475-77. HTC's new construction seeks to improperly limit the patent to the preferred embodiment. JX-3 at col. 3, lns. 65-67; Staff Br. at 66 ("the patent does not require any specific type of linking . . ."); *Altiris Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed. Cir. 2003) (district court wrongly imported limitation from preferred embodiment).[41]

[41] In an attempt to limit the claims to the use of pointers, HTC contends that the patent's reference to "automatically" performing actions is "inconsistent with" the operation of the HTC products. HTC Br. at 102. Per Ground Rule 4(c), HTC has waived this argument by failing to raise it in its pre-hearing statement. Moreover, the word "automatically" is not a part of any claim element and "pointers" are not necessary to enable the performance of a selected action. [

Moreover, HTC's comparison of its linked actions to pointers is incorrect because HTC misidentifies the infringing actions. As explained by Dr. Mowry, the infringing action subroutines [

] The detected structures are linked to these subroutines, which cause the CPU to perform a sequence of operations on the structures. Mowry Tr. 2520-22, 2533, 2535-36.

Likewise, HTC's focus on (1) the ability of users to install third-party applications like Skype and (2) [

]

HTC further submits that Apple has failed to prove infringement because Apple [

]

This new contention, that the HTC [

] Ground Rule 4(c) requires a party to "set[] forth with particularity" all of its contentions in its pre-hearing statement; "[a]ny contentions not set forth in detail as required herein shall be deemed abandoned or withdrawn." Order No. 2 at 4(c). HTC

]

has abandoned the argument that [ ] constitute a single action subroutine by failing to raise it in its pre-hearing statement.

HTC attempts to justify this failure by incorrectly claiming that Dr. Mowry changed his opinions at trial.[42] [ ]

Substantively, HTC is incorrect that [ ] As Dr. Mowry testified, these methods are

[42] HTC incorrectly contends that Dr. Mowry changed his opinion by identifying [ ]

[

]

HTC incorrectly asserts that Dr. Mowry agreed that [

]

HTC contends still further that Apple has failed to prove infringement because

[

]

At the outset, HTC is precluded from making this new non-infringement argument under Ground Rule 4(c), and in any event, is wrong. Dr. Mowry's unrebutted testimony proved that the HTC products link actions before the user's selection of that action. *E.g.* Mowry Tr. 2522-23.

With respect to Browser and Android Messaging, [ ] HTC's argument is based on the incorrect premise that Dr. Mowry [ ] HTC fails to provide a supporting citation for this statement. In fact, Dr. Mowry, Google witness David Sparks, and Dr. Olsen all testified that [ ]

HTC's citations to the transcript are improper because they all refer to the different linking mechanism of HTC Messages, not Browser or Android Messaging. HTC Br. at 112-113; Mowry Tr. 2680:3-7. In HTC Messages, [

]

HTC offers yet another argument that Apple has failed to prove infringement because for claim 15, the claimed method's structure requires performance in order, and Apple concedes that the steps in HTC's Android devices are not performed in the same order as the claim. HTC Br. at 113-115. Specifically, HTC asserts that claim 15's "linking at least one action to the detected structure" must occur before "enabling selection of the detected structure."

Claim 15, however, requires no such order. Mowry Tr. 2517. The Federal Circuit has set forth a two-part test to determine whether steps in a method claim must be performed in a particular order: when (1) "as a matter of logic or grammar, they ***must*** be performed in the order written," and (2) the specification "directly or implicitly ***requires*** such a narrow construction." *Altiris*, 318 F.3d at 1369-70 (internal citations omitted) (emphasis added).

HTC's position relies on two significant misstatements of the law. First, HTC incorrectly contends that when "***most*** of the method steps refer to the completed results from the prior step, then ***all the steps*** must take place in order." HTC Br. at 113 (emphasis added). The Court in *E-Pass* actually stated that when "most" of the steps of a method claim refer to a prior step, then "all of ***those*** steps" must take place in order. *E-Pass Techs., Inc. v. 3Com Corp.*, 473 F.3d 1213, 1222 (Fed. Cir. 2007). The Court did not state that "most" steps should be expanded to "all" steps.

Second, the Court in *Altiris* did not state that a method claim must be performed in order if the "specification ***suggests*** performing method steps in order." HTC Br. at 113 (emphasis added). *Altiris* held that steps in a method must be performed in order when the specification "***requires***" such order. *Altiris*, 318 F.3d at 1370 (emphasis added). The Court further held that the preferred embodiment's use of a certain order does not meet this standard. *Id.* at 1371.

Application of the correct law establishes that claim 15 does not require that "linking at least one action" to occur before "enabling selection of the structure." While other elements of claim 15 refer to prior steps, HTC cannot show the required dependency between the steps of "linking at least one action" and "enabling selection of the structure." *Id.* at 1370 (holding that while some steps of the method claim needed to be performed in order, others did not). HTC attempts to circumvent this result by interpreting "enabling selection of the structure and a linked action" as a single step, even though the parties agree that this claim element requires enabling the ***separate*** selection of a structure and a linked action. Mowry Tr. 2560:-61; Olsen Tr. 4043; *e.g.* JX-3 at Abstract. ("the user interface can[,] . . . upon selection of a detected structure, present the linked candidate actions"). There is no logical or grammatical reason in the claim language that justifies HTC's limitation. *See Morris Reese v. Samsung Telecomms. Am.*, No. 2:05-CV-415-DF, 2006 WL 6112195, at *19 (E.D. Tex. 2006) (dividing two elements of "step (b)" in holding that "the 'assigning' in step (b), must occur after step (a)" but that "[t]he 'generating' in step (b), on the other hand, need not occur after step (a).") Non-asserted dependent claim 21 further undermines HTC's position because it identifies "enabling selection of an action" as its own "step," separate from the step of enabling selection of a detected structure. JX-3 at col. 8, lns. 55-58. That the dependent claim imparts an order implies that claim 15 does not require an order.

The fifth element of independent apparatus claim 1 recites:

> **a user interface enabling the selection of a detected structure and a linked action; and**

The fourth element of independent method claim 15 recites:

> **enabling selection of the structure and a linked action; and**

Apple has satisfied these claim elements. Claims 1 and 15 require enabling the user to select a detected structure and separately select a linked action. Mowry Tr. 2560-61; Olsen Tr. 4043. The HTC accused products comprise such user interface program routines and methods for enabling the selection of a detected structure and a linked action. Mowry Tr. 2506-07.

[

]

[

]

HTC contends that the limitations "a user interface enabling the selection of a detected structure and a linked action" (claim 1) and "enabling selection of the structure and a linked action" (claim 15) are not met because accused HTC products do not give users the option of multiple actions. HTC Br. at 110.

Again contradicting its statement that it would use Apple's constructions, HTC contends for the first time that claim 15 requires linking multiple actions. HTC is precluded from making this new argument under Ground Rule 4(c), and in any event, is wrong. Apparatus claim 1 requires linking more than one action to a detected structure, but method claim 15, by its plain language, only requires "linking at least one action" to a detected structure. Mowry Tr. 2468.

HTC argues that claim 15 requires linking of multiple actions because Dr. Mowry supposedly testified that the "selection of . . . a linked action" requires the user to choose among multiple actions. HTC Br. at 110. But Dr. Mowry did ***not*** testify that claim 15 requires the user be given a choice of multiple linked actions. Mowry Tr. 5017-18. Rather, Dr. Mowry testified that the user must be given a choice to select an "action" such that the selection is separate from, or independent of, the selection of the "structure." *Id.* As explained above, the accused HTC products provide that choice and link multiple actions to a detected structure.

HTC incorrectly asserts that Dr. Mowry "disclaimed" an opinion that the

[

] That testimony is consistent with and buttresses the identified linked actions and does not amount to a disclaimer.

The sixth element of independent apparatus claim 1 recites:

> **an action processor for performing the selected action linked to the selected structure; and**

The fifth element of independent method claim 15 recites:

> **executing the selected action linked to the selected structure.**

Apple has satisfied these claim elements. HTC conceded that [

]

As described above, [

[43]

]

[

]

The last element of claim 1 recites:

> **a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines.**

Apple has satisfied this claim element. The HTC accused products contain a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines. Mowry Tr. 2569-72; CX-3510 at 301; CX-409C at HTC00623252, 60.

[43] [

]

2. Claim 3

Dependent claim 3 recites:

> **The system recited in claim 1, wherein the input device receives the data from an application running concurrently, and wherein the program routines stored in memory further comprise an application program interface for communicating with the application.**

Apple has not satisfied claim 3. Claim 3 requires that “the input device receives the data from an application running concurrently.” [ ] However, this piece of software is not an “input device” under proper claim construction. Indeed, Dr. Mowry admits that an instance of this software exists within the HTC Messages and Android Messaging applications. Mowry Tr. 2483. Thus, the accused HTC devices do not infringe because Dr. Mowry only identifies software as the “input device” of claim 3, and software alone cannot constitute an “input device” under proper claim construction.

Apple also contends that the “input device” limitation is met under the doctrine of equivalents. Mowry Tr. 2486. Dr. Mowry, however, provides no reasoned basis for why this piece of software should be an “input device.” *Id.* (merely asserting “any differences are insubstantial” without further analysis). In fact, Apple’s expert offered no analysis of how software alone would satisfy the function-way-result test. Mowry Tr. at 2575-76. Without such analysis, Apple cannot carry its burden of showing that software performs a function in the same way as hardware does. *Moore USA, Inc. v. Standard Register Co.*, 229 F.3d 1091, 1113 (Fed. Cir. 2000) (“The mere recital of the *Graver Tank* mantra that the accused device performs ‘the same function, in the same way, to achieve the same result,' without more, does not create a genuine issue of material fact as to whether an accused device infringes by equivalents"). Accordingly, the accused devices fail to satisfy claim 3 under the doctrine of equivalents.

3. Claim 8

Dependent claim 8 recites:

> **The system recited in claim 1, wherein the user interface highlights detected structures.**

Apple has satisfied claim 8. The HTC accused products comprise a user interface that highlights detected structures. Mowry Tr. 2576. Android Messaging and HTC Messages highlight detected structures in blue, underlined font. *Id.* 2576-77. Browser highlights detected structures with a green rectangle during the user's selection of a structure. *Id.*

4. Claim 19

Dependent method claim 19 recites:

> **The method recited in claim 15, wherein the memory contains strings, and wherein the step of detecting a structure further comprises the steps of retrieving a string from the memory and scanning the data to identify the string.**

Apple has satisfied claim 19. The accused HTC products containing the Browser infringe claim 19. Mowry Tr. 2577. [

]

5. Indirect Infringement

HTC indirectly infringes, through inducement, method claims 15 and 19. Mowry Tr. 2578. HTC has stipulated that it and its customers use the accused functionalities of Browser, Android Messaging, and HTC Messages as described in Dr. Mowry's Expert Report regarding infringement and in accordance with their user guides. HTC Use Stip. ¶¶ 15-17; Mowry Tr. 2580-81. The user guides provide further evidence that HTC induces its customers to infringe claims 15 and 19. For example, the HTC Droid Incredible User Guide teaches how to practice the '647 patent using HTC Messages and the T-Mobile G1 User Guide teaches how to practice the '647 patent using the Browser. CX-3510C at 104; CX-995C at HTC000005489; Mowry Tr. 2579-80. Additionally, HTC induces infringement by complying with AT&T's device specifications, which require that phones practice the '647 patent. Mowry Tr. 2579-80; CX-3083C at HTC007471669 ("The device shall support the parsing of the message text and address fields for usable items such as URLs and phone numbers.").[44]

HTC counters that "Apple has not shown that HTC's user guides induce infringement; rather, they simply instruct users how to use the phones." As for the AT&T specification, HTC asserts that "it only contains a high-level requirement that devices 'shall support the parsing of the message text and address fields for usable items such as URLs and phone numbers'" and the specification "does not require practicing the claims." HTC Reply at 75.

[44] The Staff agrees with Apple and submits that "HTC has induced infringement of the patent after learning of its infringement." Staff Br. at 66 n.24. The Staff also notes that HTC advertises the use of dialing phone numbers directly from text messages to its customers. *Id.*

HTC's arguments are unpersuasive. In any event, HTC does not seriously contest Apple's assertion regarding induced infringement. In fact, HTC did not challenge Dr. Mowry during the hearing. As noted, HTC has stipulated that it and its customers use the accused functionalities of Browser, Android Messaging, and HTC Messages as described in Dr. Mowry's Expert Report regarding infringement and in accordance with their user guides. Moreover, AT&T specification's requirement that devices support message "parsing" supports Apple's assertion that HTC induces infringement by complying with AT&T's device specifications.

Accordingly, HTC indirectly infringes, through inducement, method claims 15 and 19.

In summary, Apple has shown that HTC's accused products infringe the asserted claims 1, 8, 15, and 19 of the '647 patent. However, Apple failed to show infringement with respect to claim 3.

C. Technical Prong of the Domestic Industry Requirement

Apple argues that it "proved that the iPhone 3GS running Mobile Mail ('iPhone') practices claims 1, 4,[45] 8, 15, and 19 of the '647 patent" and that "Apple's evidence that the iPhone satisfies the domestic industry requirement is undisputed by both HTC and Dr. Olsen." Apple Br. at 123-124.

HTC did not contest Apple's contention that it satisfies the technical prong of the domestic industry requirement.

The Staff submits that Apple practices claims 1, 4, 8, 15, and 19 of the '647 patent in the iPhone. Staff Br. at 67. The Staff explains that "HTC's expert appears to have

[45] It is noted that Apple has not asserted dependent claim 4 for infringement.

failed to provide any testimony regarding Apple's proof on this issue. Therefore, the undisputed evidence is that Apple uses this patent." *Id.*

For the reasons set forth below, Apple has satisfied the technical prong of the domestic industry requirement with respect to the '647 patent.

1. Independent claims 1 and 15

The preamble of independent apparatus claim 1 recites:

> **A computer-based system for detecting structures in data and performing actions on detected structures, comprising:**

The preamble of independent method claim 15 recites:

> **In a computer having a memory storing actions, a method for causing the computer to perform an action on a structure identified in computer data, comprising the steps of:**

Apple has satisfied the preambles of claims 1 and 15. As noted, the parties agree that the preambles of claims 1 and 15 are not limitations. In any event, Apple's iPhone satisfies the preambles of claims 1 and 15. Mowry Tr. 2582; CPX-11 (iPhone 3GS). The iPhone is a computer-based system that, using Apple's Data Detector technology, detects structures in data and performs actions on those detected structures (claim 1) and has memory storing actions that perform an action on a structure identified in computer data (claim 15). Mowry Tr. 2587; Serlet Tr. 4248-49.

The first element of independent apparatus claim 1 recites:

> **an input device for receiving data;**

The first element of independent method claim 15 recites:

> **receiving computer data;**

Apple has satisfied these claim elements. Apple has shown that the iPhone contains hardware-based input devices for receiving computer data, including a touchscreen, wireless internet adapter for receiving email data, a radio for receiving text messages and phone calls, and memory. Mowry Tr. 2584; CPX-11; CX-0384 at APPHTC_00002112 (radio and wireless internet), CX-0386C at APPNOK1209485 (256MB of DDR RAM). Such hardware input devices satisfy "input device" as properly construed. Mowry Tr. 2584-85.

The second element of claim 1 recites:

> **an output device for presenting the data;**

Apple has satisfied this claim element. The iPhone contains an output device for presenting the data in the form of a touchscreen display. Mowry Tr. 2585; CPX-11 (iPhone 3GS); CX-0384 at APPHTC_00002114.

The third element of claim 1 recites:

> **a memory storing information including program routines including**

Apple has satisfied this claim element. The iPhone contains memory storing information including program routines. Mowry Tr. 2585; CPX-11; CX-0384 at APPHTC_00002114 (Flash memory), CX-0386C at APPNOK1209485 (256MB of DDR RAM).

The fourth element of independent apparatus claim 1 recites:

> **an analyzer server for detecting structures in the data, and for linking actions to the detected structures;**

The second and third elements of independent method claim 15 recite:

> **detecting a structure in the data;**
>
> **linking at least one action to the detected structure;**

Apple has satisfied these claim elements as discussed below.

"analyzer server for detecting structures in the data" (claim 1) and "detecting a structure in the data" (claim 15)

Apple has shown that the iPhone includes an analyzer server for detecting structures in data under the parties' agreed-upon constructions of "analyzer server," "detecting," and "structures." Mowry Tr. 2586. Specifically, Mobile Mail finds and identifies phone numbers, URLs, street addresses, and email addresses in mail messages. *Id.* 2586-87.

Dr. Mowry testified that the analyzer server for detecting structures [

] Mowry Tr. 2588; CX-4324C; CX-4325C. [

] used to detect structures [

]

Mowry Tr. 2588; CX-4324C; CX-4334C. [

] *Id*

"analyzer server for ... linking actions to the detected structures" (claim 1) and "linking at least one action to the detected structure" (claim 15)

Apple has shown that the iPhone includes an analyzer server and method for linking actions to detected structures. Mowry Tr. 2591. In Mobile Mail, a user is presented with several actions to take on a detected structure. *Id.* 2593. For example, in the case of a detected phone number, possible actions include launching the phone or text message application with the detected structure. *Id.*

Dr. Mowry testified that the analyzer server for linking actions to the detected structures [

] Mowry Tr. 2591-92; CX-4347C; CX-4348C; CX-4349C; CX-4340C; CX-4332; CX-444. [

] Mowry Tr. 2592. [

] Mowry Tr. 2595.

There is no dispute that the iPhone performs these limitations. Mowry Tr. 2595. Dr. Mowry testified that the linked actions in the iPhone "necessarily lead[] to causing the CPU to perform a sequence of operations on the detected structure. And that's precisely the type of linking to actions as described in the '647 patent." *Id.* 2598.

The fifth element of independent apparatus claim 1 recites:

> **a user interface enabling the selection of a detected**

> **structure and a linked action; and**

The fourth element of independent method claim 15 recites:

> **enabling selection of the structure and a linked action; and**

Apple has satisfied these claim elements. First, Dr. Mowry testified that the iPhone has a user interface and method for enabling the selection of detected structures in an email. Mowry Tr. 2589. That user interface comprises [

] *Id.* 2589-90; CX-4319C; CX-4328C.

Second, the iPhone also has a user interface and method for enabling the selection of a linked action. Mowry Tr. 2598-99. This user interface comprises the

[

] *Id.* 2599-2600; CX-4340C; CX-4347C; CX-4348C; CX-4349C.

The sixth element of independent apparatus claim 1 recites:

> **an action processor for performing the selected action linked to the selected structure; and**

The fifth element of independent method claim 15 recites:

> **executing the selected action linked to the selected structure.**

Apple has satisfied these claim elements. Apple has shown that the iPhone has an action processor and method for executing or performing the selected action linked to the detected structure. Mowry Tr. 2600. Dr. Mowry testified that the action processor [ ] Mowry Tr. 2600; CX-4342C; CX-4354C; CX-4329C. [ ] Mowry Tr. 2600-01; CX-0385C at APPHTC_00652555 (describing [ ] Neither HTC nor Dr. Olsen contests that the iPhone satisfies these limitations.

The last element of claim 1 recites:

> **a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines.**

Apple has satisfied this claim limitation. The iPhone contains a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines. Mowry Tr. 2602-03; CPX-11; CX-0386C at APPNOK1209485 (ARM Cortext-A8 processor).

2. Claim 4

Dependent claim 4 recites:

> **The system recited in claim 1, wherein the analyzer server includes grammars and a parser for detecting structures in the data.**

Apple has satisfied claim 4. Apple proved that the iPhone's analyzer server uses grammars and a parser to detect structures as required by claim 4. Mowry Tr. 2604; CX-4314C; CX-4315C; CX-4337C; CX-4338C. For example, [

] Mowry Tr. 2604; CX-4315C.

3. Claim 8

The system recited in claim 1, wherein the user interface highlights detected structures.

Apple has satisfied claim 8. The iPhone highlights detected structures and therefore satisfies claim 8. Mowry Tr. 2605-06. As described above, the iPhone [ ] to highlight and underline structures in blue. *Id.*

4. Claim 19

The method recited in claim 15, wherein the memory contains strings, and wherein the step of detecting a structure further comprises the steps of retrieving a string from the memory and scanning the data to identify the string.

Apple has satisfied claim 19. As required by claim 19, the iPhone scans a mail message to identify strings retrieved from a string library. Mowry Tr. 2606-07; CX-4351C; CX-4353C. For example, the iPhone includes files listing strings [ ] which are retrieved from memory and used in a parser to detect structures in data. *Id.*]In summary, Apple has satisfied the technical prong of the domestic industry requirement with respect to the '647 patent.

D. Validity

HTC contends that the asserted claims of the '647 patent are anticipated or rendered obvious by the Perspective System and Handbook. HTC Br. at 117. HTC further argues that the NeXTSTEP reference manual ("Manual") (RX-889) and the NeXTSTEP System each anticipate the asserted claims of the '647 patent. *Id.* at 130. Additionally, HTC urges that U.S. Patent No. 5,859,636 ("Pandit") (RX-4603) anticipates claims 1, 8, 15, and 19 under the parties' claim constructions and that Pandit anticipates claim 3 under Apple's construction. *Id.* at 138.

The Staff, in agreement with HTC, submits that all asserted claims[46] of the '647 patent are anticipated and/or rendered obvious by the Perspective System and Handbook. Staff Br. at 67. The Staff further contends that "if the Perspective product and handbook do not anticipate every asserted claim of the '647 patent, those claims are anticipated also by, at least, U.S. Patent No. 5,859,636 to Pandit." *Id.* at n.26.

Apple argues that Perspective fails to disclose, among other things, claim elements in independent claims 1 and 15 that require linking actions to the detected structure, enabling the selection of a linked action, and performing the selected action on the selected structure. Apple Br. at 84. Apple also contends that NeXTSTEP Spell Checker does not anticipate claims 1 and 15 because it does not disclose detecting structures, linking actions to detected structures, and enabling the selection of a detected structure. *Id.* at 140-142. Apple further urges that dependent claims 3 and 19 are not anticipated and that the asserted claims are not rendered obvious by NeXTSTEP. *Id.* at

[46] The Staff, however, states that "[c]laim 3 is not invalid under the Staff's construction of "input device," but it is invalid under Apple's construction." *Id.* at n.25.

142-143. As to the Pandit patent, Apple contends that Pandit does not anticipate the asserted patents because Pandit is not prior art and Pandit also lacks detecting a structure, and enabling the selection of a detected structure. *Id.* at 128-135, 143-144.

As explained below, HTC and the Staff have not shown by clear and convincing evidence that the asserted claims of the '647 patent are (1) anticipated or rendered obvious by Perspective; (2) anticipated by NeXTSTEP;[47] and (3) anticipated by the Pandit patent.

1. Perspective

Perspective is a "personal information manager . . . that allowed the user to keep track of contacts and appointments and notes." Schaffer Tr. 3700-01. Perspective is a relational database—it marks a location of text with a database record number and relates that number to a database entry irrespective of the text at the marked location or the text within the database entry. Mowry Tr. 4890-92.

The testimony of Dr. Mowry shows that Perspective fails to disclose, among other things, claim elements in claims 1 and 15 that require linking actions to the detected structure, enabling the selection of a linked action, and performing the selected action on the selected structure.

"linking actions to the detected structures"

The fourth limitation of apparatus claim 1 requires "linking actions to the detected structures" and the corresponding third limitation of method claim 15 requires "linking at least one action to the detected structure." As discussed below, Perspective does not link

[47] It is noted that the Staff did not assert NeXTSTEP as an invalidating prior art to the '647 patent.

actions to the detected structures and thus does not anticipate claims 1 and 15.

HTC alleges that in Perspective the detected structure is the text "Bob" and the "double-tap" and "D" gestures invoke linked actions. HTC Br. at 123; Olsen Tr. 3883. But neither of these gestures links actions or is itself an action linked to the text Bob because (1) they are simply gestures and not computer subroutines and (2) neither gesture invokes code that is associated with or operates on the text "Bob." Mowry Tr. 4890-92.

It is undisputed that each party's construction of "linking actions" and "linking at least one action" requires the action subroutine operate on the particular structure to which it is linked. Mowry Tr. 4887-88. In Perspective, "double-tapping" a name opens a profile in the Address Book and writing a "D" on a name brings up a dialer. *Id.* 4886-87; Olsen Tr. 3893. Both operations invoke the database record number at the location of the gesture without knowledge of or performing operations on the detected name. Mowry Tr. 4890-92. It is undisputed that "[a]t no point along the way is the value in that text used. It is not necessary. Relational databases don't need them and they don't use them." Mowry Tr. 4891.

As a relational database, Perspective uses "standard relational techniques to use integers to connect together different entries in different components of the database." *Id.* 4908. When opening a profile by double-tapping on a name or opening a dialer by writing "D" over a name, the value of the name is never used or operated on. *Id.* 4890-92. Perspective simply follows the number underlying the chosen text to open a profile or display the dialer. *Id.* The alleged action subroutines identified by Dr. Olsen never operate ***on*** a detected structure and therefore do not constitute a claimed "action."

In fact, the source code that Dr. Olsen relies on to demonstrate this functionality supports Apple's position. *Id.* 4899-4901. Simply put, Perspective only cares "about ***where*** you have tapped, not ***what*** you have tapped on." *Id.* 4897 (emphasis added).

In another scenario, where the user writes a name and the contact list contains more than one entry for that name, Perspective does not link to a subroutine that causes the CPU to perform operations on a detected structure as it simply marks the name's location in text with a database record number. *Id.* 4903-04. In the last scenario, where the user writes the word "Meet" followed by a name that is ***not*** in the contacts list, Perspective does not detect the name and thus does not link actions to a detected structure. *Id.*

HTC and the Staff incorrectly argue that the HTC products do not use the detected structures in performing actions (HTC Br. at 124), and "Apple has not shown that the actual text of 'Bob' is acted on" when a dialer is launched in the HTC products (Staff Br. at 69). Contrary to the Staff's contention, Apple did not suggest that the HTC products will detect the text "Bob" and then launch the dialer with that text. Rather, Apple has shown that the HTC products operate on a detected structure. For example, unlike Perspective, the HTC accused products detect phone numbers and will launch a dialer ***with the detected number*** when a user selects the detected number and then selects the associated "Call" action. *E.g.*, Apple Br. at 111-112.

HTC also argues that Dr. Mowry testified that the HTC products use the "value" rather than the actual structure. But Dr Mowry's testimony contradicts that contention.[48]

[48] HTC is precluded from arguing that [ ] HTC Br. at 124. First, under Ground

E.g., Mowry Tr. 2553 (testifying that HTC products "cause the CPU to perform operations on a structure"), 2637-38 ("The structure is the value . . . a structure is an instance of a pattern, so the structure is the value."). And HTC's expert agrees with Dr. Mowry that the HTC products "detect structures and ***allow the user to perform actions on these structures***." Olsen Tr. 4021 (emphasis added).

HTC further alleges that the database record number is a "proxy" for a detected structure. HTC Br. at 124. The claims, however, require that the action operate on the detected structure. Dr. Mowry's testimony, cited by HTC, makes clear that Perspective does not act on the detected name or even care about its value. Mowry Tr. 4891-92, 4896-4902. As the Staff admitted, in Perspective, using the detected text "would be useless." Staff Br. 69.

Moreover, HTC misstates its own expert's testimony to support a new theory of linking actions—contending that "Perspective's Associate linked actions . . . using specific identifier numbers for the method to be invoked." HTC Br. at 123. Notably, on direct, HTC's expert disagreed and stated that these "specific identifier numbers" are "actually ***not*** part of Perspective. This is part of Penpoint." Olsen Tr. 3892 (emphasis added). Penpoint was an operating system distinct from Perspective. Schaffer Tr. 3700-01. HTC did not assert Penpoint against the '647 patent at the hearing or cite any

Rule 4(c), HTC waived this argument by failing to raise it in its pre-hearing statement. Second, Apple objected to HTC making this exact same argument at trial and HTC represented that it was not doing so. Tr. 4025-26 ("MR. DONOVAN: I have an objection . . . If there is a new non-infringement object based on that the [ ] it is not in the report . . . . MR. VAN NEST: I don't think that's part of what Dr. Olsen is talking about, Your Honor. At least it is not what I intend to have him talk about."). In any event, this new argument is without merit.

Penpoint functionality, including "specific identifier numbers," in its pre-hearing statement. Thus, this new argument is substantively wrong and is procedurally barred.

HTC also criticizes Apple as resorting to a "hyper-technical (and incorrect) reading of the agreed-upon portion of the claim construction."[49] HTC argues that "[i]n an effort to distinguish Perspective, Dr. Mowry grasps at a thin reed—the word 'on'—arguing that it requires the action—the subroutine—to use the 'value' of the detected structure." HTC further contends that the '647 patent "does not support Apple's meta claim construction" and that the word "'on' simply means 'associated with' or 'related to'; there's no basis for reading in the further limitation 'on the value of'." HTC Reply Br. at 79.

The '647 patent, however, explains that a "structure" is in fact more specific than as proposed by HTC.

> Much data that appears in a computer user's day-to-day activities contains <u>recognizable structures that have semantic significance such as phone numbers, e-mail addresses, post-office addresses, zip codes and dates</u>. In a typical day, for example, a user may receive extensive files from word-processing programs and e-mail that contain several of these structures. However, visually searching data files or documents to find these structures is laborious and cognitively disruptive, especially if the document is lengthy and hard to follow. Furthermore, missing a structure such as a date may lead to missing an important meeting or missing a deadline.
>
> To help facilitate searching a document for these structures, programmers can create or employ pattern analysis units, such as parsers, to <u>automatically identify the structures</u>. <u>For</u>

[49] While the parties dispute the proper construction of this claim element, they agree that it should include "linking detected structures to computer subroutines that cause the CPU to perform a sequence of operations ***on the particular structures*** to which they are linked." Joint Claim Construction, App'x A at 16 (emphasis added).

> the purposes of the present description, the term "pattern" refers to data, such as a grammar, regular expression, string, etc., used by a pattern analysis unit to recognize information in a document, such as dates, addresses, phone numbers, names, etc. The term "structure" refers to an instantiation of a pattern in the document. That is, a "date" pattern will recognize the structure "Oct. 31, 1995." The application of a pattern to a document is termed "parsing."

JX-3 at col. 1, lns. 13-35 (Description of the Background Art) (emphasis added).

Thus, the '647 patent shows that a "structure" is more specific than as proposed by HTC. The patent explains, for example, that "a 'date' pattern will recognize the structure 'Oct. 31, 1995'." Although HTC argues that the word "on" simply means "associated with" or "related to" and that there's no basis for reading in the further limitation "on the value of," the patent itself discloses that "structures" have "semantic significance such as phone numbers, e-mail addresses, post-office addresses, zip codes and dates." The patent teaches that a specific "pattern" will recognize a specific "structure" since the "term 'structure' refers to an instantiation of a pattern in the document." For example, a date pattern will recognize a hypothetical structure "July 15, 2011" and a zip code pattern will likewise recognize a hypothetical structure "20436."

The Perspective prior art lacks this required pattern-structure element. First, HTC's and the Staff's exemplary structure "B-O-B" in Perspective does not have a corresponding pattern. Thus, Perspective is not able to "automatically identify the structures" which is a key feature of the disclosed invention of the '647 patent and required by the fourth claim element "an analyzer server for detecting structures in the data." Even if "B-O-B" in Perspective is a "structure," Perspective does not have "an analyzer server for detecting structures in the data." Rather, a person using Perspective has to manually detect the alleged structure "B-O-B" in the data by either double clicking or writing a "D" over the alleged structure.

Moreover, the Staff concedes that, in Perspective, the detected "text is not important for the action." Staff Br. at 69. The Staff now asserts that the "detected structure" is not the name, but rather a phone number or an entry in the database (contact list). *Id.* But this contention cannot meet the claims because neither the phone number nor the text within the database is detected, is a structure, or is selected by the user—all of which are claim requirements. Thus, properly understood, the Staff's position on Perspective shows that Perspective is not the same as the claimed invention.

At bottom, HTC contends that the actions need only be linked to the detected structures but not operate on those particular detected structures. HTC Br. at 78-81. The plain language of the claims, the parties' agreed-upon construction of the term "action," and the entire specification make clear that the action must operate on the detected structure. *See, e.g.*, JX-3, preamble of claim 1 ("performing actions on detected structures"), preamble of claim 15 ("to perform an action on a structure identified in computer data"), col. 2, lns. 31-34 (defining "action" as a computer subroutine that "perform[s] a sequence of operations on the particular structure to which it is linked"); Staff Br. at 61 (adopting the same definition).

Accordingly, Perspective does not link actions to the detected structures and thus does not anticipate claims 1 and 15.

"performing the selected action linked to the selected structure"

The sixth limitation of apparatus claim 1 requires "performing the selected action linked to the selected structure" and the corresponding fifth limitation of method claim 15 requires "executing the selected action linked to the selected structure."

HTC contends that the [

] satisfy these elements because they are called by the "D" and double-tap gestures. HTC Br. at 127. However, for the reasons discussed above, Perspective does not perform (or execute) an action linked to a selected structure because it performs no operations on the detected text. In fact, Dr. Mowry testified that the [

] do not use or even contain the detected structure. Mowry Tr. 4899-4902.

"enabling the selection of a detected structure and a linked action"

The fifth limitation of apparatus claim 1 requires "enabling the selection of a detected structure and a linked action" and the corresponding fourth limitation of method claim 15 requires "enabling selection of the structure and a linked action."

HTC contends that a user can simultaneously select a structure and a linked action when a user double-taps or writes a "D" on a name. HTC Br. at 125. As shown above, Perspective fails to disclose "linking actions" and thus it necessarily fails to disclose enabling the selection of a "linked action." Further, it is undisputed that these elements require the ***separate*** (independent) selection of a detected structure and a linked action. Mowry Tr. 2560-61; Olsen Tr. 4043. Yet, Perspective offers no such separate selection—the user cannot select "Bob" and also separately choose to open the dialer or contact. The only "selection" identified by Dr. Olsen is the selection of a recognized name (Mowry Tr. 5018); this does not enable the separate or independent selection of an action. *Id.* 4905-06, 5010, 5018-19. In an attempt to show separate selection, HTC and the Staff argue that selection of a structure occurs by "putting the stylus above the bolded word 'Bob' as opposed to the other words on the screen." Staff Br. at 70; Olsen Tr. 3917-18. But the claims require program routines and methods for selecting a structure, not a user's mental choice to hover the stylus over a piece of text. JX-3, claims 1, 15; Mowry Tr. 4905-06.

Moreover, contrary to HTC contention that Dr. Mowry testified that a single selection step satisfies these elements, Dr. Mowry instead testified that a single selection step satisfies the claim elements ***only if*** the selection of the structure and selection of the action are "***independent***," *e.g.*, when all of the structures and associated actions are presented to the user, and Perspective does not allow for this independent selection. Mowry Tr. 5017-19 (emphasis added).

HTC then wrongly contends that Dr. Mowry construed the claims to require a particular user interface for displaying actions. HTC Br. at 126. Dr. Mowry testified repeatedly that the asserted gestures in Perspective (1) do not constitute an independent selection of a structure and action; (2) are not presented as actions associated with a structure; and (3) could not have been presented to the user because the "D" gesture was a hidden option unknown to Perspective users. Mowry Tr. 4905-06, 5010-14, 5017-18. In fact, the Handbook never mentions the "D" gesture scenario. Mowry Tr. 5045.

In summary, HTC and the Staff have not shown by clear and convincing evidence that the asserted claims of the '647 patent are anticipated by Perspective because it fails to disclose, among other things, claim elements in independent claims 1 and 15 that require linking actions to the detected structure, enabling the selection of a linked action, and performing the selected action on the selected structure.

Obviousness of Claims 1, 3, 8, 15, and 19

HTC contends that the Perspective "system" and Handbook separately combined with the "state of the art" render the asserted claims obvious. Olsen Tr. 3947-49. Yet, Dr. Olsen's obviousness analysis is rejected due to his application of an entirely inconsistent and significantly incorrect level of ordinary skill. Mowry Tr. 4880-83. In addition, both of Dr. Olsen's Perspective combinations suffer the same problems: Dr. Olsen failed to explain what constitutes the "state of the art," what elements of the "state of the art" make up for Perspective's shortcomings, or why a person of ordinary skill would combine the state of the art with the disclosures of the Perspective system. Simply put, Dr. Olsen's conclusory statements regarding an unidentified "state of the art" cannot establish that each element of any asserted claim was obvious to one of ordinary skill. *See id.* 4908-09.

2. NeXTSTEP Spell Checking Tool

For the reasons set forth below, HTC has not shown by clear and convincing evidence that NeXTSTEP Spell Checking Tool anticipates or renders obvious the asserted claims of the '647 patent.

As an initial matter, HTC has not identified a proper NeXTSTEP "System." HTC alleges that the NeXTSTEP "system" anticipates claims 1, 3, 8, 15, and 19. Olsen Tr. 3855. But Dr. Olsen never defined the purported NeXTSTEP "system," and thus HTC cannot show that a NeXTSTEP "system" invalidates the claims. Olsen Tr. 3961 (failing to define the NeXTSTEP "system"). Nor can HTC contend that the individual references relied on by Dr. Olsen constitute a single system. Dr. Olsen based his opinion on references from different versions of NeXTSTEP: (1) source code, including release

] (2) the NeXTSTEP General Reference book ("Reference") describing ***version 3.0***; and (3) devices running an ***unknown version*** that were not admitted into evidence. Olsen Tr. 3950 (relying on source code, Reference, and devices), 4046-47 (software is version 3.2); MowryTr. 5025 (same), 4911-12; RX-889 at HTC007279798 (Reference describes version 3.0). [

] HTC's contention that they count as a single system is inconsistent with the understanding of one of ordinary skill and is unsupportable under the law. Mowry Tr. 4911-12; *Studiengesellschaft Kohle,* 726 F.2d at 726–27.

"detecting structures"

The '647 patent requires detection of a "structure," which the parties agree means an "instance of a pattern" where a pattern is used to "recognize information." Olsen Tr. 4096; Mowry Tr. 2458-59, 2491-93, 4912-13. Structures are thus semantically significant nuggets of text (e.g., telephone numbers) that are "positive match[es] of a pattern." Mowry Tr. 2491-93; JX-3 at col. 1, lns. 13-16; col. 2, lns. 10-13; col. 2, lns. 28-32; JX-9 at APPHTC_00338320.

In plain contrast, former NeXTSTEP employee Bertrand Serlet testified that misspellings are "just words that were not in the dictionary"—i.e., text that is ***not recognizable***, ***has no semantic significance, and does not match a pattern***. Serlet Tr. 4250-51 [ ] 4250 [

] Mowry Tr. 4913-14; RX-2346C at L.89. Misspellings are therefore meaningless "non-instances of patterns"—the "polar opposite" of the parties' definition of "structure." Mowry Tr. 4913-14. Indeed, Dr. Olsen's contention that

NeXTSTEP anticipates is admittedly based on his claim that "the absence of a particular pattern" counts as a pattern. Olsen Tr. 4103. This contention would render the term "structure" meaningless. Mowry Tr. 4914-15 ("[T]he universe of possibilities is a non-instance of a pattern. There would be no rhyme or reason to that."), 5049-51.

"linking actions to the detected structures"

The parties agree that these limitations require the linked action to operate ***on the detected structure***. Mowry Tr. 4887-88; Olsen Tr. 3838. HTC's contention that options to "correct," "find next," or "ignore" unrecognized text constitute linked actions fails because none of these options operate on a structure. Mowry Tr. 4918-20; Olsen Tr. 3969-70. Further, even assuming for the sake of argument that [

] Mowry Tr. 4918-20, 5051; RX-2355C at L.130-153. [

] Mowry Tr. 4918-20; RX-2344 at L.1051-57. And [

] Mowry Tr. 4918-20.

"enabling the selection of a detected structure and a linked action"

These limitations require a user to be able to "select" a particular detected structure, such as by clicking it with a mouse. Mowry Tr. 4915-16; JX-3 at col. 4, lns. 11-17; col. 4, lns. 23-27. [

] Mowry Tr. 4916-17.

HTC's contention that the "find next" button satisfies these limitations fails because it improperly conflates selection and detection. Olsen Tr. 3979-80. [

] Mowry Tr. 4916-18; RX-2344C at L.1051-1057.]

In summary, HTC has not shown by clear and convincing evidence that NeXTSTEP Spell Checking Tool anticipates asserted claims 1 and 15 of the '647 patent.

Additional limitations of claim 3 and 19

Dr. Olsen testified that NeXTSTEP had a software input device that satisfies claim 3. Olsen Tr. 3988; Mowry Tr. 4920. Under a proper claim construction, input device does not include software. Thus, HTC has failed to show that NeXTSTEP anticipates claim 3.

For the string-matching limitation of claim 19, HTC contends that the opposite of string matching—***failing*** to match a string—satisfies the step of detecting structures. Olsen Tr. 3993-94; 3968-69. This contention must fail and in fact [

]

Mowry Tr. 4920-21.

Obviousness of Claims 1, 3, 8, 15, and 19

HTC contends that the NeXTSTEP "system" and Reference each combined with the "state of the art" render the asserted claims obvious. Olsen Tr. 3987-88, 3997-98. Dr. Olsen's conclusory obviousness analysis is rejected due to his application of inconsistent and incorrect levels of ordinary skill. Mowry Tr. 4880-83. Dr. Olsen's NeXTSTEP combinations suffer from the same flaws as his Perspective combinations: he does not identify what aspects of the "state of the art" make up for NeXTSTEP's shortcomings, where the missing elements can be found in the "state of the art," or why a person of ordinary skill would combine those pieces of the "state of the art" with the NeXTSTEP "system" or Reference. *Id.* at 4922.

3. U.S. Patent No. 5,859,636 ("Pandit")

For the reasons set forth below, HTC has not shown by clear and convincing evidence that the Pandit patent anticipates the asserted claims of the '647 patent.

HTC alleges that Pandit anticipates the asserted claims of the '647 patent. Olsen Tr. 4011-12, 4014. HTC asserts Pandit as § 102(e) prior art. Pandit was filed on December 27, 1995, only five weeks before Apple filed for the '647 patent. RX-4603; JX-3 (filing date of February 1, 1996). But Pandit is not prior art because the inventors conceived and reduced to practice the claimed inventions of the '647 patent well before Pandit's filing date. Mowry Tr. 4923-24; *see infra* (conceived in late 1994; reduced to practice in mid-1995).

"detecting structures"

In addition to not being prior art, Pandit fails to disclose detecting structure(s). Mowry Tr. 4923-24. The parties agree that "detecting" means "finding and identifying." Olsen Tr. 3884. But Pandit does not "find and identify" structures. Rather, Pandit requires that users "find" and accent a ***single*** piece of text before even trying to recognize the text. Mowry Tr. 4924-25; RX-4603 at FIG. 2. Pandit may not even recognize the user-accented text. Mowry Tr. 4925-26. Thus, Pandit does not "detect" a structure, let alone multiple structures, as required by claims 15 and 1 respectively. Mowry Tr. 4926-28.

HTC asserts that Apple "admits that Pandit recognized structures" and that "Apple's sole dispute is that Pandit did not 'find' structures because Pandit 'requires that users 'find' and accent a single piece of text'." HTC explains that "Apple's argument relies on an erroneous reading of the '647 patent and ignores Pandit's plain disclosure." HTC Br. at 87.

HTC's argument falls short. As an initial matter, it is undisputed that the parties agreed that "detecting" means "finding and identifying." Staff Br. 60; Apple Br. 103 n. 27; HTC Br. 139 n.31. But Pandit does not "find" structures, and HTC cannot plausibly contend otherwise because a user must accent a single piece of text before Pandit will try to recognize it. This does not satisfy the "finding" requirement of the claims.

HTC attempts to overcome this critical omission by contending, for the first time, that there is no support for the "finding" requirement. HTC Br. 139-140. First, HTC is precluded from disavowing an agreed-upon construction. Second, HTC is misguided. Claim 1 requires that the computer system have the ability to detect multiple structures at one time in the data received by the input device. Further, the specification makes clear that detecting structures includes finding them within a user's documents. *See, e.g.*, JX-3, col.1, lns.13-27 (describing problem that "visually searching data files or documents to ***find*** these structures is laborious and cognitively disruptive, especially if the document is lengthy and hard to follow") (emphasis added); Bonura Tr. 2295-96 (describing same).

As HTC notes, the specification also discloses analyzing an entire document or a portion of the document. HTC Br. at 140 (quoting JX-3, col. 5, lns. 55-56). In contrast, Pandit does not analyze the content or a portion of a document; Pandit could only look at the single piece of user-accented text. *See, e.g.*, RX-4603, FIG. 2. And, although Pandit may use pattern matching, that does not satisfy the "finding" requirement. Mowry Tr. 5035. In short, Pandit does not detect structures because it does not "find" them; Pandit relies on the user to do the work of finding and accenting text.

"enabling the selection of a detected structure and a linked action"

Pandit does not disclose this limitation. Mowry Tr. 4923-24. As explained above, Pandit does not detect structures. In Pandit, the user must manually "find" and accent text before Pandit will attempt to recognize it. As such, the accenting of unrecognized text cannot be the selection of a detected structure. Mowry Tr. 4928.

Additional limitations of claims 3, 8, 19

Dr. Olsen testified that Pandit had a software input device that satisfies claim 3. Olsen Tr. 4012. Moreover, Dr. Olsen failed to identify the purported software that is the claimed input device. *See* Mowry Tr. 4928-29. Under a proper claim construction, input device does not include software. Thus, HTC has failed to show that Pandit anticipates claim 3.

Regarding claim 8, because Pandit does not detect structures, there are no detected structures to highlight. Further, even if the user-accented text is recognized, Pandit does not disclose text that is highlighted. *Id.* Regarding claim 19, Pandit fails to disclose string matching. *Id.* Dr. Olsen's conclusory and unsupported testimony regarding claims 8 and 19 cannot satisfy HTC's burden of proof on invalidity. Olsen Tr.

4014.

4. Conception and Reduction to Practice

Conception in Late 1994 to Early 1995

Apple has shown that the inventors—Thomas Bonura, Jim Miller, Bonnie Nardi, and David Wright—conceived the inventions of claims 1, 3, 8, 15, and 19 of the '647 patent as early as September 25, 1994, and no later than March 1995. Mowry Tr. 4942, 4974; Bonura Tr. 2301, 2305. The inventors of the '647 patent were a multidisciplinary group of researchers in the Intelligent Applications subgroup of the Advanced Technology Group—Apple's research arm. Bonura Tr. 2292-94. The evidence shows that [ ] the inventors had begun working on the "structure detectors" [

] *Id.*, *id.* at 2294-99; CX-2407C. That project led to the conception and eventual reduction to practice of the '647 inventions.

Inventor Testimony and Dr. Miller's Septemter 1994 Email

One of the inventors, Dr. Bonura, testified at the hearing that by late summer to early fall of 1994 the inventors had formed in their minds a definite and permanent idea of the '647 inventions. Bonura Tr. 2300-01, 2305-08; Mowry Tr. 4974; CX-2404C.

[

]

CX-2404C; Bonura Tr. 2303-05; Mowry Tr. 4966. [ ] shows conception of a computer system and method for automatically finding and identifying interesting "nuggets" of information (i.e., structures), associating actions to those structures, allowing users to select and initiate those actions, and providing a user interface for controlling the functionality. CX-2404C; Bonura Tr. 2300-01; 2303-04.

[ ] for detecting structures and linking actions to detected structures (the analyzer server), enabling the selection by the user of a detected structure and a linked action (the user interface), and performing the selected action linked to the detected structure (the action processor). CX-2404C; Mowry Tr. 4963-65. [ ] had an input device for receiving data, an output device for presenting data, memory storing information [ ] and a processor unit coupled to the input device, the output device and the memory. Mowry Tr. 4961-63, 4965. [ ] also shows that the system would have string-matching capabilities to detect information [ ] (claim 19). *Id.* 4965-66; CX-2404C.

[ ]

Apple has shown that [ ] further evidences conception of claims 1, 3, 8, 15, and 19 of the '647 patent [ ] Mowry Tr. 4966-67, 4974. Dr. Bonura testified that [ ] Bonura Tr. 2307. [ ] Bonura Tr. at 2308-10; Mowry Tr. 4966-67; CX-2267C. For example, [ ] Bonura Tr. 2328-30; Mowry Tr. 4968; CX-8002C; CX-3382C. [ ] Bonura Tr. 2324-25, 2328-30; CX-8002C – CX-8006C. Further, Dr. Mowry [ ] and concluded that it showed conception. Mowry Tr. 4957, 4959, 4974; CX-3382C; CX-3383C; CX-3385C; CX-3391C; CX-8002C.

Claims 1 and 15

[ ] Bonura Tr. 2328-30, 2335; CX-8003C. [ ] and highlights detected structures [ ] Bonura Tr. 2335-36; CX-8004C; CX-3383C. The user selects one of the structures [ ] Bonura Tr. 2336-37; CX-8005C; CX-3382C. For example, [ ] *Id.* [ ] Bonura Tr. 2337; CX-8006C; CX-3391C.

[ ] therefore is a computer-based system with an input device for receiving data (*e.g.*, [ )] an output device for presenting data, memory storing information including program routines, and a processing unit coupled to the input device, the output device and the memory. Mowry Tr. 4969-70; CX-8002C; CX-8003C; CX-3382C; CX-3383C. [ ] also shows routines that detect structures, link actions to detected structures, enable the separate selection of a detected structure and linked action, and perform the selected action on the detected structure.

Mowry Tr. 4970-72; CX-8004C; CX-3383C; CX-8005C; CX-3385C; CX-8006C; CX-3391C.

Claims 3, 8, and 19

Apple argues: [ ] shows conception of claim 3 because [

] Apple Br. at 131, citing Mowry Tr. 4973; CX-8003C; CX-8004C; CX-3383C. Apple has not shown conception of claim 3 because [ ] is not included under a proper claim construction of "input device."

[ ] shows conception of claims 8 and 19 because it highlights detected structures, [ ] which are detected using string matching. Mowry Tr. 4974; CX-8004C; CX-3383C.

[ ] Reduction to Practice

Apple proved, and HTC did not dispute, that the inventors worked diligently in late 1994 to reduce to practice the claimed invention [

] Bonura Tr. 2338-40, 2347, 2357-58; Mowry Tr. 4987, 4977-78.

Dr. Bonura testified that the inventors [

] Bonura Tr. 2338-39. [

]

Bonura Tr. 2338-46; CX-2178C; CX-2192C; CX-2193C; CX-2195C; CX-2183C; CX-

2184C; CX-2186C. Dr. Mowry testified that [

] Mowry Tr. 4975-77, 4987.

Dr. Bonura also testified that, [

] Bonura Tr. 2347, 2357-58; Mowry Tr. 4978; CX-2297C. [

] Bonura Tr. 2347-50, 2358; CX-8007C – CX-8011C. Dr. Mowry [

] and concluded that [ ] showed actual reduction to practice. Mowry Tr. 4958-60, 4987; CX-3394C – CX-3397C; CX-3401C; CX-3402C]

Independent claims 1 and 15

The Preambles

[ ] are computer-based systems for detecting structures in data and performing actions on detected structures (claim 1) and are computers having memory storing actions that perform an action on a structure identified in computer data (claim 15). Mowry Tr. 4979-80, 4983-84; Bonura Tr. 2339-40.

> **an input device for receiving data (1) / receiving computer data (15); an output device for presenting the data (1); a memory storing information including program routines (1)**

Because they run on a computer with a user interface, [

] each had an input device for receiving data and a method for receiving computer data, an output device for presenting the data, and a memory storing information including program routines. Mowry Tr. 4979-80]

an analyzer server for detecting structures in the data and for linking actions to the detected structures (1) / detecting a structure in the data; linking at least one action to the detected structure (15)

[ ] each satisfy these limitations. Mowry Tr. 4980-82; Bonura Tr. 2355-56. [ ] detects structures in data and, [ ] links actions to detected structures. Mowry Tr. 4980-82; CX-2193C at BONURA00000145-146, 157; CX-2178C; CX-2184C; CX-2183C; CX-2195C; Bonura Tr. 2341, 2346. [ ] Mowry Tr. 4978-79; CX-2195C.

[ ] shows the detection of structures in the data and the linking of actions to those structures. Mowry Tr. 4980-82; Bonura Tr. 2351-56; CX-8008C; CX-8010C; CX-3396C.

a user interface enabling the selection of a detected structure and a linked action (1) / enabling selection of the structure and a linked action (15)

[ ] each include a user interface enabling the separate selection of a detected structure and a linked action. [ ] highlights detected structures [ ] to select them. Mowry Tr. 4980-81; CX-2193C at BONURA00000149, 151-152. Moreover, [ ] could select an action linked to the detected structure. Mowry Tr. 4982-83; CX-2192C at BONURA00000136 [

] Mowry Tr. 4980-81; Bonura Tr. 2351-52, 2355-56; CX-8008C; CX-3396C. Selecting one of these detected structures [ ] Mowry Tr. 4982-83; Bonura Tr. 2353-54; CX-8010C; CX-3401C.

an action processor for performing the selected action linked to the selected structure (1) / and executing the selected action linked to the selected structure (15)

[ ] each comprise an action processor for performing the selected action linked to the selected structure. Mowry Tr. 4983-84; Bonura Tr. 2354-55; CX-2183C at BONURA00000100-101,104; CX-8011C; CX-3402C. [ ] initiate the performance of the selected action. Mowry Tr. 4983-84. [ ] Mowry Tr. 4983-84; Bonura Tr. 2354-55; CX-8010C; CX-8011C.

a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines (1)

[ ] ran on a computer with a user interface and thus each used a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines. Mowry Tr. 4979-80.

Claim 3, 8 and 19

Apple contends that [ ] practices claim 3 because [

] Apple Br. at 135, citing Mowry Tr. 4984-85; Bonura Tr. 2358-59; CX-8007C; CX-3394C; CX-2290C. Apple has not shown conception of claim 3 because software is not included under a proper claim construction of "input device."

[ ] highlight detected structures and therefore practice claim 8. Mowry Tr. 4980-81, 4986; CX-2193C; CX-8008C; CX-3396C. [ ] practice claim 19 by using string matching to detect structures. Mowry Tr. 4986-87 (describing string matching in the [ ]CX-2186C; CX-8008C; CX-3396C.

HTC argues that "because of evidentiary flaws at trial," Apple cannot support its argument that the invention of Pandit patent was conceived and diligently reduced to practice before the filing of the patent application. HTC Br. at 143-144. Specifically, HTC explains that Apple relies on the testimony of Dr. Bonura but the bulk of his testimony was based on screen shots from a disk that was not allowed into evidence. HTC urges that Dr. Bonura's testimony regarding the screen shots should be stricken or given no weight and similarly, Dr. Mowry's testimony about screen shots taken from Miller-1154 should be stricken or given no weight because those screen shots were never authenticated.

HTC's argument is unconvincing. The undersigned ***unconditionally*** admitted the screen shots and invited HTC to renew its objections to the extent issues regarding the workability of the CD remain unresolved. Bonura Tr. 2333-39. As noted at trial, HTC's sole basis for objection to the CD—its assertions that it could not open or understand the contents of the CD—was inconsistent with the record, including that its expert had no issues with the timely produced CD and had in fact used and testified about them in his report. Bonura Tr. 2317-18, 2321. After HTC's objection to the CD at trial and before Dr. Mowry's testimony, Apple inquired as to whether HTC needed additional disks or information about them and HTC declined, thus waiving any "conditionality." Mowry Tr. 4944-45. HTC's objections are without merit and refuted by its own expert, as well as the personal knowledge of two separate witnesses, whose testimony was without objection.

Indeed, Dr. Bonura testified that he [ ] Bonura Tr. 2307, 2309-10, 2324-25, 2328, 2338; 2347-50. Dr. Mowry subsequently testified that the screen shots admitted into evidence during Dr. Bonura's testimony were consistent with and "identical" in substance to those he prepared and relied on in forming his opinions regarding conception and reduction to practice. Mowry Tr. 4957-60. HTC made no effort to cross-examine either Dr. Bonura or Dr. Mowry regarding the substance, origins, or genuineness of these screenshots, which comes as no surprise given that HTC was well aware of and had access to the demo and prototype. *See* Bonura Tr. 2323 (HTC admitted that "[t]hose screen shots are things the experts have talked about. We all know where the screen shots came from.").

Moreover, the record is clear that HTC welcomed testimony regarding the screen shots prepared by Drs. Bonura and Mowry in lieu of admission of the disks from which they came. As HTC must concede, ***HTC did not renew its objections***. In fact, HTC expressly stated that it had no objections to the screen shots and testimony regarding them. *See, e.g.*, Mowry Tr. 4943-44 (HTC stating no objection to testimony regarding screen shots), 4950-51 (same), 4955 ("And we have no objection to the screen shots. . ."), 4957-58 (no objection to admission of screen shots CX-3382C, CX-3383C, CX-3385C, CX-3391C, CX-3394C – CX-3397C, CX-3401C, CX-3402C). In contrast to its post-hearing brief, HTC represented that it ***did not think*** the underlying CD from which the screen shots came would be useful to the undersigned. *Id.* 4955-56 (Staff representing the same). It is disingenuous for HTC to argue that the CD is unnecessary and should not be admitted because the screen shots have been admitted, but then move to strike testimony concerning the screen shots because the CD was not admitted. *See, e.g.*, Bonura Tr. 2327 (overruling HTC's objection, stating "[y]ou can't have it both ways. You wanted a screen shot. You didn't want the CD to be used, but you said that counsel can use the screen shot."). As a result, it is meritless for HTC to now raise objections to this evidence.

X. U.S. Patent No. 5,481,721

The '721 patent is entitled, "Method For Providing Automatic And Dynamic Translation Of Object Oriented Programming Language-Based Message Passing Into Operation System Message Passing Using Proxy Objects." JX-1. The invention of the '721 patent provides a method and apparatus for the distribution of objects and the sending of messages between objects that are in different processes. *Id.,* (Abstract). *See* Spielman Tr. 2703 ("Generally, the '721 patent is about an object-oriented message that is used in an interprocess communication making use of an operating system-based message."). The invention relates to the field of object-oriented programming and distributed computing. *Id.,* col. 1, lns. 15-16 (Field of the Invention).

Apple asserts independent claim 1 and dependent claims 5 and 6. The asserted claims read as follow:

> **1**. A method for sending an object oriented programming language based message having dynamic binding from a first object in a first process to a second object in a second process, said method comprising the steps of:
>
> transmitting, using a first processing means, said object oriented programming language based message to a first proxy in said first process;
>
> using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;
>
> transmitting said operating system based message to said second process in said second processing means at run time;
>
> decoding, using a second process, said operating system based message into a language based message;
>
> transmitting, using said second processing means, said object oriented programming language based message to said second object in said second process;
>
> executing said object oriented programming language based message by said second object in said second process.
>
> **5**. The method of claim **1** wherein the step of executing said object oriented programming language based message further includes the steps of:

said second object determining, using said second processing means, whether additional information is needed to execute said object oriented programming language based message;

said second object generating, using said second processing means, an object oriented programming language based query if it is determined that additional information is needed;

encoding, using said second processing means, said object oriented programming language based query into an operating system based query at run time if it is determined that additional information is needed;

transmitting said operating system based query to said first process at run time, using said second processing means if it is determined that additional information is needed;

decoding, using said first processing means, said operating system based query into an object oriented programming language based query at run time if it is determined that additional information is needed;

transmitting, using said first processing means, said object oriented programming language based query to said first object if it is determined that additional information is needed.

6. The method of claim **5** further including the steps of:

said first object generating, using said first processing means, an object oriented programming language based reply to said object oriented programming language based query;

encoding said object oriented programming language based reply into an operating system based reply at run time, using said first processing means;

transmitting, using said first processing means, said operating system based reply to said second process at run time;

decoding, using said second processing means, said

> operating system based reply into an object oriented programming language based reply at run time;
>
> transmitting, using said second processing means, said object oriented programming language based reply to said second object.

JX-1 at col. 71, lns. 2-23; col. 71, ln. 46 – col. 72, ln. 24.

A. Claim Construction[50]

1. "first processing means" and "second processing means"

Claim Terms	Apple Construction	HTC and Staff Construction
"first processing means" (1, 5, 6, 19, 20, 21)	This term is not a means-plus-function term, and should be construed to mean: a processor.	*Function*: • transmitting said object oriented programming language based message to a first proxy in said first process (1, 19) • encoding said object oriented programming language based message into an operating system based message at run time (1, 19) • decoding said operating system based result into an object oriented programming language based result at run time (2, 20) • transmitting said object oriented programming language based result to said first object (2, 20) • decoding said operating system based query into an object oriented programming language based query at run time if it is determined that additional information is needed (5, 21) • transmitting said object oriented programming language based query to said first object if it is determined that additional information is needed (5, 21) • said first object generating an object oriented programming language based reply to said object oriented programming language based query (6) • encoding said object oriented programming language based reply into an operating system based reply at run time (6)

[50] A person of ordinary skill in the art would have a Bachelor's degree in computer science, or its equivalent, and at least three years in programming or software design and implementation, or its equivalent. Rinard Tr. 4369-70; Spielman Tr. 5094.

		• transmitting said operating system based reply to said second process at run time (6) *Corresponding Structure:* a computer processor configured to perform the steps set out in Figures 3A, 3B, 3C, 4 & 5; and as described in the '721 patent at col. 7:30-59, 8:29-45, 10:53-11:65, 12:21-13:15, 14:50-15:26; *see also* claim 24
"second processing means" (1, 5, 6, 19, 20, 21)	This term is not a means-plus-function term, and should be construed to mean: a processor.	*Function*: • transmitting said operating system based message to said second process at run time (1) • transmitting said object oriented programming language based message to said second object in said second process (1, 19) • encoding said object oriented programming language based result into an operating system based result at run time (2, 20) • transmitting said operating system based result to said first process at run time (2, 20) • said second object determining whether additional information is needed to execute said object oriented programming language based message (5, 21) • said second object generating an object oriented programming language based query if it is determined that additional information is needed (5, 21) • encoding said object oriented programming language based query into an operating system based query at run time if it is determined that additional information is needed (5, 21) • transmitting said operating system based query to said first process at run time if it is determined that additional information is needed (5, 21) • decoding said operating system based reply into an object oriented programming language based reply at run time (6) • transmitting said object oriented programming language based reply to said second object (6) • decoding said operating system based message into a language based message (19) • said second object executing said object oriented programming language based message and generating an object oriented programming language based result (20) *Corresponding Structure:* a computer processor configured to perform the steps set

		out in Figures 3A, 3B, 3C, 4 & 5; and as described in the '721 patent at col. 7:30-59, 8:29-45, 10:53-11:65, 12:21-13:15, 14:50-15:26; *see also* claim 24

Joint Claim Construction, App'x A at 10-12.

Apple argues that "[p]roperly understood, the 'processing means' terms should be construed as 'a processor'," citing a recent Federal Circuit decision that complainants submit rejects HTC's and the Staff's position that terms with an ordinary meaning are means-plus-function terms simply because they include the word "means." Apple Br. at 148.

HTC and the Staff contend that the "first processing means" and "second processing means" are means-plus-function terms and that the specification contains the corresponding structure for the functions performed. HTC Br. at 147-157; Staff Br. at 74-79.

For the reasons set forth below, as proposed by HTC and the Staff, the claim terms "first processing means" and "second processing means" are found to be means-plus-function terms and the functions and the structures of these terms are construed as proposed by HTC and the Staff.

As an initial matter, Apple's proposed construction and its expert's reports did not identify alternative functions to those identified by HTC and the Staff. (Apple disputed whether there were any functions at all recited in the claims, but not which functions correspond if the terms are means-plus-function). Thus, the parties' only dispute here involves structure. Apple has preserved no dispute regarding the functions of the first and second processing means.

It is settled law that "[t]he use of the word 'means,' which is part of the classic template for functional claim elements, gives rise to a presumption that the inventor used the term advisedly to invoke the statutory mandates for means-plus-function clauses." *Sage Prods. v. Devon Indus, Inc.*, 126 F.3d 1420, 1427 (Fed. Cir. 1997) (quotes and citation omitted); *see York Products, Inc. v. Central Tractor Farm & Family Center*, 99 F.3d 1568, 1574 (Fed. Cir. 1996) (word "means" in a claim creates a rebuttable presumption that § 112, ¶ 6 applies). The presumption in favor of means-plus-function construction is rebutted if the patentees show (1) that the claim specifies no function performed by the "means," or (2) the claim itself recites sufficient structure to perform the recited function entirely. *Sage Prods.*, 126 F.3d at 1427-28.

A claim recites sufficient structure to perform a function where the claim uses language with precise structural character that can perform the entire recited function. Thus, a claim requiring the function of "tearing" recited sufficient structure where it identified "perforations" in the claim itself. *Cole v. Kimberly-Clark Corp.*, 102 F.3d 524, 531 (Fed. Cir. 1996).

With respect to the '721 patent, intrinsic evidence shows that the "first processing means" and "second processing means" are used to perform specific functions. For example, claim 1 states that the first processing means is used for the function of "transmitting said object oriented programming language based message to a first proxy in said first process." It is also used for the function of "encoding said object oriented programming language based message into an operating system based message at run time." JX-1 at col. 71, lns. 10-12.

Claims 5 and 6 likewise disclose functions performed using the first or second processing means. The disclosed functions include "determining whether additional information is needed to execute said object oriented programming language based message," "generating an object oriented programming language based query if it is determined that additional information is needed," "generating an object oriented programming language based reply to said object oriented programming language based query," and "encoding said object oriented programming language based reply into an operating system based reply at run time." JX-1 at col. 71, ln. 46 – col. 72, ln. 23.

Additionally, the '721 patent's file history demonstrates the context in which these "means" terms were introduced to the claims. It is clear that the Applicants intended to take advantage of section 112, paragraph 6 to overcome multiple rejections of their application.

The '721 patent specification and claims underwent several rejections under section 112. JX-7 at APPHTC-00008074-8076 (rejection under section 112, paragraphs 1 & 2), APPHTC-00008204-05 (same). In order to overcome these rejections, the Applicants amended the claims to add a "means" limitation. JX-7 at APPHTC-00008182. The Applicants further argued that the specification disclosed structure for implementing the invention, including a "forward::" method that enabled certain claimed functions. JX-7 at APPHTC-00008188-8189 (identifying source code and patent specification disclosures), APPHTC-00008017 (specification discloses "[t]he receiver proxy 904 is an object that executes a *forward*:: method"), APPHTC-00008035 (source code identified to PTO invokes "forward::" method), APPHTC-00008237 (directing PTO to "proxy 904 that executes a *forward*:: method to encode the message and transmit the message to the remote object").

The Applicants presumably were aware that "means" terms carry a special meaning in patent claims, and knew what signal the word "means" would send to the examiner. *See Brasseler, U.S.A. I. L.P. v. Stryker Sales Corp.*, 267 F.3d 1370, 1385 (Fed. Cir. 2001) ("knowledge of the law is chargeable to the inventor," and "inventors represented by counsel are presumed to know the law"). The applicants consciously chose to use "means" language; if the applicants instead wanted to describe a "computer," or a "processor," they knew how to do so. (JX-1 at col. 73, lns. 45-46 (claim 14) ("first and second computers"), col. 7, lns. 5-6 ("general purpose computer system"), col. 7, lns. 30-31 (same), col. 7, lns. 35-36 ("computer system of FIG. 4"); col. 7, ln. 35 ("CPU"); col. 7, ln. 39 (same), col. 7, ln. 49 (same), col. 7, ln. 53 ("microprocessor"), col. 7, ln. 59 (same).

Applicants, however, did ***not*** use any of those words. Rather, they used the word "means" in response to a section 112 rejection, knowing the special meaning and import that word would carry. Applicants argued that these new "means" limitations represented "a computer actually implementing these steps" – *i.e.*, a processor programmed with algorithms for implementing the claimed functions. JX-7 at APPHTC-00008188-8189. Further, the Applicants simultaneously pointed the PTO to algorithms in the specification, including "forward::", that enabled and implemented the functions carried out by the claimed "means." JX-7 at APPHTC-00008188-8189, 8017, 8035, 8237. This context does not rebut the presumption of means-plus-function treatment; it confirms that means-plus-function treatment is appropriate.

As to the structure corresponding to the functions, HTC's expert, Dr. Martin Rinard, discussed at length how portions of the patent specification provide structure for the claim 1 functions of "transmitting, using a first processing means, said object oriented programming language based message to a first proxy in said first process" and "encoding said object oriented programming language based message into an operating system based message at run time." Dr. Rinard explained that the computer system of FIG. 4 is an element of the corresponding structure, but that this computer system and its processor require algorithms to perform the claimed functions. Rinard Tr. 4381-82.

The algorithm for the "transmitting" function is described by elements 501, 502, and 504 of FIG. 5, as well as portions of the patent specification that discuss elements of FIGS. 3A, 3B, and 3C (*e.g.*, JX-1 at col. 11, lns. 8-12), and that discuss relevant elements of FIG. 5 (*e.g.*, JX-1 at col. 14, lns. 59-67), including the step of invoking the "forward::" method. Rinard Tr. 4396-99.

Likewise, Dr. Rinard explained that the algorithm for the "encoding" function is provided by the execution of the "forward::" method, which encodes and transmits the message. Rinard Tr. 4405. The execution of the "forward::" method is described by elements 505, 506, 511, 512, and 507 of FIG. 5, as well as portions of the patent specification that discuss encoding. This included disclosures that "the present invention" was able to determine "how to encode the arguments for each message as it is encountered" so that the encoding algorithm could "'learn' how to talk to another program" during runtime (*e.g.*, JX-1 at col. 12, lns. 21-45). Rinard Tr. at 4400-04.

As Dr. Rinard explained, "forward::" implements a single, general-purpose encoding algorithm that can dynamically learn how to send new messages. It enables a proxy object to obtain the method signature for new and unknown messages, thereby learning how to send those kinds of messages. Rinard Tr. 4403, 4399-4400. As a result, the algorithms used by the "present invention" in the "transmitting" and "encoding" steps are different than the prior art because they use no "pre-defined set of messages," and require no "code generation step." JX-1 at col. 10, lns. 1-7; Rinard Tr. 4405-07. The prosecution history confirms this. The Applicants specifically explained to the PTO that "forward::" was structure for encoding and transmitting functions: "proxy 904 that ***executes a forward:: method to encode the message and transmit the message to remote object 902***." JX-7 at APPHTC_00008258.

Apple's expert, Ms. Spielman, likewise confirmed that "forward::" is corresponding structure. She admitted that in her own alternative proposal for corresponding structure, "[t]he receiver proxy 904 is an object that ***executes a forward:: method***." JX-1 at col. 11, lns. 9-10 (emphasis added); Spielman Tr. at 2902, 2921 (discussing forward:: method; "I'm including that as part of the structure"). Ms. Spielman further acknowledged that the steps of Figure 5, including elements 502 and 504, are a flow diagram of "forward::" as used in the "present invention" of the '721 patent. Spielman Tr. 2921-22, 2924-25. Apple's expert admitted that the applicants repeatedly directed the PTO to the "forward::" method – both narrative descriptions of the method, and source code illustrating the method's use. Spielman Tr. 2902-03, 2905-06, 2917, 2923-25. These conclusions accord with Dr. Rinard's testimony. *See* Tr. 4405, 4407-08. Thus, the "forward::" method is corresponding structure.

Accordingly, as proposed by HTC and the Staff, the claim terms "first processing means" and "second processing means" are found to be means-plus-function terms and the functions and the structures of those terms are construed as proposed by HTC and the Staff.

As noted, Apple submits that the plain language of the method claims at issue supports its construction of "a processor," arguing that the Federal Circuit recently held that "claiming a means for processing or processing means 'may simply claim a general purpose computer, although in means-plus-function terms'." Apple Br. at 148, citing *In Re Katz*, 639 F.3d 1303, 1316 n. 11 (Fed. Cir. 2011).

Apple's position that "[t]hese terms are not means-plus-function terms and should be construed to mean: a processor" is unconvincing. Apple Br. at 147. In that regard, *Katz* held that all terms at issue were means-plus-function terms. 639 F.3d 1303, 1315-16. These terms included (1) means-plus-function terms reciting specialized computer-implemented functions that required algorithms, and (2) means-plus-function terms reciting potentially generic functions that hardware alone could perform.[51] Thus, *Katz* does not support Apple's view that these terms are not means-plus-function terms.

Apple next asserts that "invoking means-plus-function treatment in a method claim requires that both a method step and a separate 'function'" be present in the claim. Complainants argue that the only "functions" that HTC identifies are the method steps of the asserted claims. Apple Br. at 150, citing *Epcon Gas Sys. Inc. v. Bauer Compressors Inc.*, 279 F.3d 1022, 1028 (Fed. Cir. 2002).

Apple misreads *Epcon Gas*. *Epcon Gas* addressed "step plus function" claims, ***not*** "means plus function" claims. *Id.* at 1028 (noting distinction between means-plus-function form, which uses "means" language, and step-plus-function form, which uses

[51] *Katz* also held that a "processing" function may still require a particular algorithm, depending on how that function is described in the patent. *Katz*, 639 F.3d 1303, 1317 (remanding to district court for further claim construction, with instruction to "determine whether the functions [of 'processing' 'storing' and 'receiving'] . . . can be performed by a general purpose processor or, instead, constitute specific computer-implemented functions as to which corresponding algorithms must be disclosed.").

"step for" language). The method claim in *Epcon* ***did not include the word "means,"*** and the Court's analysis in *Epcon* related solely to "step-plus-function" format in the context of a method claim. *Id.* at 1025-26 (reproducing claim 2, which contained no "means" term), 1028 (claim 2 not in step-plus-function format where "claim includes no words indicating 'step plus function' form, such as 'step for'").

Apple also contends that "[t]he specification and file history confirm that 'processing means' refers to a general purpose computer, or in other words, 'a processor'. First, the specification explains that the 'invention may be implemented in any type of computer system or programming or processing environment' and that 'any [] suitable microprocessor or microcomputer may be utilized.' Second, the file history demonstrates that both the Examiner and the Applicants understood 'processing means' to refer to 'a computer' or 'a processor' and intended it to have that meaning." Apple Br. at 149 (citations omitted).

As noted above, the presumption of means-plus-function construction can be rebutted if the claim itself recites sufficient structure to perform the recited function entirely. *Sage Prods.*, 126 F.3d at 1427-28. In that regard, Apple argues that the term "processing means" imports sufficient structure to rebut this presumption because the "first processing means" and "second processing means" would be understood as a general purpose computer processor, which Apple asserts is sufficient structure to perform the claimed functions. Spielman Tr. 2868-69, 2872-74 ("transmitting using a first processing means would be understood to be a processor").

Apple's argument is erroneous. A general purpose computer processor, without additional structure in the form of an algorithm, cannot perform the functions recited in claim 1 of the '721 patent. Rinard Tr. 4377-79. Indeed, Ms. Spielman admitted that a processor, on its own, cannot perform the "transmitting" function in claim 1 because this function requires software containing an algorithm. Spielman Tr. 2885-86 ("Q. [A processor] doesn't transmit said object-oriented programming language-based message? *A.* ***No, that's the software that's doing that.***"). Ms. Spielman further admitted that a person of ordinary skill would need to "infer" the appropriate undisclosed algorithm to implement that function.

> Q. So is it your opinion that someone could infer [an algorithm for "transmitting"], but it is not in there? Or do you think it is actually in the words here?
>
> A. I think it is in the – I think it is in the words ***by inferring*** from someone of ordinary skill.
>
> Q. Which words?
>
> A. What I just stated, 'transmitting using a first processing means,' somebody of ordinary skill would understand that they would be able to implement transmitting from that using a processor.

Spielman Tr. 2880; *see* Tr. 2788, 2872-80.

In addition, Apple's arguments regarding the file history are misplaced. As noted above, the '721 claims were rejected multiple times under § 112. The Applicants responded by adding "means" terms to the claims and in so doing specifically elected to use "means" language. If they wanted to describe a "computer," or a "processor," they knew how to do so. In fact, where the Applicants wanted to claim a computer, they did so explicitly; for example, claim 14 claims "first and second computers." The Applicants did ***not*** use those words in the asserted claims. They instead used the term "means," with the full knowledge of the special meaning and import that a "means" term would carry with an Examiner evaluating a response to a section 112 rejection.

Moreover, Apple's argument that "computer" and "processor" mean the same thing, and that each of these terms is synonymous with "processing means," must fail. Claim 14 uses the word "computer," but claims 1, 5, and 6 do not – further supporting HTC's proposed construction, which differentiates these terms. *See Forest Labs., Inc. v. Abbott Labs.*, 239 F.3d 1305, 1310 (Fed. Cir. 2001) ("Where claims use different terms, those differences are presumed to reflect a difference in the scope of the claims.").

Apple's emphasis on the Examiner's reference to "a computer-implemented action" is likewise unpersuasive. A "computer-implemented action" describes a *function*, not a processor. The Examiner's statements ask for clarity on the *function*, which the Examiner noted was *computer-implemented.* JX-7 at APPHTC_00008205. This discussion of the claims is not only consistent with means-plus-function treatment, but echoes the Federal Circuit's words describing means-plus-function terms. *See, e.g., Aristocrat*, 521 F.3d at 1333 (discussing "a § 112 ¶ 6 claim for a computer-implemented function").

Citing various examples, Apple further claims that "the 'processing means' terms cannot be in means-plus-function form because HTC cannot identify a legally proper structure corresponding to the supposed 'function'." Apple Br. at 150-152.

As discussed above, the evidence shows that HTC and the Staff identified the appropriate corresponding structure. Apple argues that certain elements of HTC's structure are not necessary to perform the claimed functions, including boxes 505, 506, 512, and 513 of FIG. 5. As Dr. Rinard explained, however, these portions of "forward::" are clearly corresponding structure. They represent steps used by "forward::" to determine "how to encode the arguments for each message as it is encountered" so that the encoding algorithm can "'learn' how to talk to another program" during runtime. JX-1 at col. 12, lns. 21-45; Rinard Tr. 4400-04. Without performing these functions, "forward::" could not "locate code that responds to a variety of different messages, thus avoiding the necessity of having to write a separate method for each selector." JX-1 at col. 14, lns. 50-53. This functionality is not merely useful; the Applicants themselves used it to distinguish the patent over the prior art. JX-1 at col. 10, lns. 1-7 (describing advantages over prior art, which had "no pre-defined set of messages" and "no code generation step"). As a result, this structure is linked and associated with the claims. Rinard Tr. 4405.

Apple further argues that HTC and the Staff cannot propose the same structure for the "first" and "second" processing means. Apple Br. at 151. This argument fails to appreciate the fact that the patent itself states that the "first processing means and said second processing means *are the same processing means*." JX-1 at col. 72, lns. 25-27 (claim 7), col. 76, lns. 1-3 (claim 24) (emphasis added). Moreover, FIG. 5 links "forward::" with *multiple* functions, including functions performed by the first and second processing means. There is no route other than "forward::" to carry out these functions.[52]

Further, Apple ignores the recursive operations required of the first and second processing means. The patent uses "nested, recursive, remote messages," and notes that "messages may be nested arbitrarily deep." JX-1 at col. 11, lns. 21-26. These "nested, recursive, remote messages" are passed using the algorithms executed by the first and

[52] *See* JX-1 at FIG. 5 (illustrating "first processing means" functions of transmitting and encoding, including at boxes 502 and 504, and "second processing means" functions of transmitting to the process and then to the object, including at boxes 507 and 508).

second processing means, including to obtain information needed by remote objects. JX-1 at col. 11, lns. 54-65. Because multiple nested calls may go back and forth, both the first and second "means" must have identical programming; otherwise only one "means" would be able to encode and transmit a message, and the other "means" could not recursively "send other messages back to the local process . . . before providing a reply to the initial message." *Id.*

Apple's arguments regarding claim 5 structure likewise ignore the recursive nature of the claims. A recursive function makes a call *to itself. See, e.g.*, RX-0902 at 109-110. Claim 5 is a recursive call to the structure used in claim 1. Ms. Spielman admitted this during the hearing, noting that the "same mechanism is used" in claims 1, 5, and 6 to perform remote method invocation. Spielman Tr. 2745-48. This comports with the file history, in that the Applicants argued claim 5 was enabled by "the recursive nature of the present invention." JX-7, APPHTC_00008278; *id.* APPHTC_00008237 (claim 5 enabled because "[w]here additional information is needed, the present invention supports nested, recursive, remote messages").

The structure for claim 5 *must* be the *same structure* as for claim 1, because claim 5 recites the *recursive execution of the same structure* used in claim 1. JX-1 at col. 71, ln. 46 – col. 72, ln. 6. This includes the "determining" and "generating" functions of claim 5, which the patent describes as part of this recursive operation. JX-1 at col. 11, lns. 8-20 (disclosing use of "forward::" method to encode and transmit a message), col. 11, lns. 21-42 (describing recursive execution of same method), col. 11, lns. 54-65 (generating and determining explained as part of "recursive nature of the present invention").[53]

Importantly, the '721 patent illustrates "forward::" as the sole algorithm for the claimed functions. JX-1 at FIG. 5. The prosecution history explicitly states that "proxy 904 that executes a forward:: method *to encode the message and transmit the message* to remote object 902." JX-7 at APPHTC_00008258. The patent claims that "forward::" and its capabilities distinguished it over the prior art (JX-1 at col. 10, lns.1-7), and the patentees repeatedly pointed the Examiner during prosecution to "forward::" as described in the specification and as used in the source code appendix. Even Apple's expert admitted that "forward::" is corresponding structure. Spielman Tr. 2902, 2921. Apple's arguments regarding structure other than "forward::" do not alter the core role of "forward::" in carrying out the claimed functions.

Finally, Apple argues that HTC is trying to "change" its construction, citing testimony regarding the "mouse" in FIG. 4. Apple Br. at 152. Apple's argument is unpersuasive. The entirety of HTC's proposal identifies appropriate corresponding structure. This includes FIG. 4, which is "a block diagram illustrating a general purpose computer system for implementing the present invention." JX-1 at col. 7, lns. 5-6. This computer system is linked and associated with the recited functions. It is what "implement[s] the present invention." *Id.* Neither HTC, nor Dr. Rinard, has ever stated that no other "general purpose computer system" could be equivalent to the system in

[53] Additionally, claim 5 is *part of* the steps recited in claim 1 and recites functions that occur *during* "the step of executing" in claim 1, and the "said first processing means" and "said second processing means" in claim 5 refer to the same "first processing means" and "second processing means" in claim 1. JX-1. These "means" *must* contain the structures that they use during the "step of executing" recited in claim 1, including structures for functions that are part of this "step of executing."

FIG. 4. Dr. Rinard's testimony on what is "required" by the claims merely notes that computer systems without a "mouse" could still practice the invention. Rinard Tr. 4604-05.

2. "proxy"

Claim Term	Apple Construction	HTC and Staff Construction
"proxy"	plain and ordinary meaning: an object that acts as a local receiver for objects in the local process on behalf of another object	an object created in the same process as a sender object and acts as a local receiver for all objects in the process

Joint Claim Construction, App'x A at 10.

As proposed by Apple, the claim term "proxy" is construed to mean "an object that acts as a local receiver for objects in the local process on behalf of another object."

In general, claim terms are to be given their ordinary meaning to a person of skill in the art. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005). Here, both experts agree on the plain and ordinary meaning of "proxy." Ms. Spielman explained that, as used in the art, the point of a proxy is "to act on behalf of another object." Spielman Tr. 2757-58. Dr. Rinard agreed, admitting that the "general understanding of proxy" is "something that represents something else." Rinard Tr. 4616.

The specification confirms that "proxy" should be given its plain and ordinary meaning, as it repeatedly uses the term to refer to an object that represents, or acts for or on behalf of something else. For example, the specification describes how a "local object communicates with a 'proxy' ***that locally represents the remote object.***" JX-1 at col. 3, lns. 59-61. Likewise, in describing an embodiment of the invention, the specification states that "[f]or a task to communicate with a 'receiver' object in another address space it must . . . then create a local 'proxy' ***for the object.***" JX-1 at col. 9, lns. 17-20. Further, in when describing how proxies are created the specification notes that "[p]roxies ***for any other objects*** to be communicated are created dynamically as they are encountered." JX-1 at col. 12, lns. 15-17. The relationship between a proxy and an object it represents is illustrated in FIGS. 3A-3C, where each "proxy" is shown aligned with the object for which it is a proxy. JX-1 at FIG. 3A-3C; Spielman Tr. 2758-59.

HTC and the Staff propose a construction for "proxy" that reads out any requirement that a proxy act on behalf of something else and, instead, requires only that a proxy "act[] as a local receiver for all objects in the process." Both experts agree that this is not the term's plain meaning. Rinard Tr. 4616; Spielman 2756. A deviation from a term's plain meaning is only appropriate if the patentee acted as his own lexicographer, by using a "special definition of the term [that] is clearly stated." *Laryngeal Mask. Co. Ltd. v. Ambu A/S*, 618 F.3d 1367, 1372 (Fed. Cir. 2010) (citation omitted). HTC and the Staff have not attempted to show that the '721 patent "clearly state[s]" a special definition for the term "proxy" and have not argued that the patentee acted as his own lexicographer.

The evidence that HTC cites in support of its position is a single sentence from the Abstract (and the Summary of the Invention), which states that a "proxy acts as a local receiver for all objects in the local program." JX-1 at Abstract; col. 6, lns. 56-57. However, this is consistent with the plain meaning as it describes the proxy encoding a message and transmitting it to software in another process so it can be executed there – that is, acting "on behalf of" the receiving object in the second process. JX-1 at Abstract.

HTC cannot show that the '721 patent's use of the term "proxy" deviates from the plain and ordinary meaning of that term.

"Unicity"

HTC and the Staff argue for a revised construction to add a new limitation – "unicity." This term, however, is not related to the claims and is found nowhere in the language of HTC's construction.[54] As explained below, HTC's proposal that the term "proxy" itself, properly construed, required "unicity" was an untimely change of position and is, therefore, rejected. Rinard Tr. 4411-16.

HTC's attempt to inject "unicity" into claim 1 began with its opening expert report. From that point to trial, HTC argued that "unicity" was required by the "decoding" step of claim 1 – a method step that does not use the term "proxy." In fact, HTC's expert explicitly "note[d] that the term 'unicity' is not part of the claim construction for the claim term 'proxy'." Rinard Tr. 4618. Instead, Dr. Rinard stated that this property ***"is ensured by the decoding algorithm disclosed in the '721 patent to perform the decoding function***." Rinard Tr. 4618 (quoting Rinard 2/15/11 Rebuttal Rpt. at 98 n. 43). In keeping with this position, Dr. Rinard admitted that "unicity" was not part of the construction for "proxy." Rinard Tr. 4619 ("Q: And I note that you indicate in the footnote on this page that unicity is not part of the claim construction [of] proxy? A: Yes.").

Moreover, the Staff's examination confirmed that, up until trial, Dr. Rinard's position was that "unicity" was part of the "decoding" step, rather than of the

[54] HTC and its witnesses referred both to "unicity" and a "one-to-one relationship" between proxies and objects they represent. Dr. Rinard agreed these refer to the same concept. Rinard Tr. 4610.

construction for "proxy." Dr. Rinard testified:

> Q: However, you indicate that *unicity is achieved by the decoding algorithm* that's disclosed in the patent *that performs the decoding function*? Do you see that?
>
> A: Yes.

Rinard Tr. 4643.

In any event, HTC's attempt to read "unicity" into the construction of the term "proxy" is inconsistent with the plain language of the claims. Claim 1 recites "a first proxy." Dr. Rinard conceded that the ordinary meaning of "proxy" in computer science does not require a "one-to-one" relationship. Rinard Tr. 4616. The fact that the claim recites "***a*** proxy" supports the conclusion that as a general rule, "an indefinite article 'a' or 'an' in patent parlance carries the meaning of 'one or more'" – not one and only one as HTC proposes. *KCJ Corp. v. Kinectic Concepts Inc.*, 223 F.3d 1351, 1356 (Fed. Cir. 2000). Thus, as long as "a first proxy" exists, the claim is satisfied – the presence of other "proxies" is irrelevant.

While HTC seems to suggest that "unicity" is required by the phrase "for all objects" in its construction, whether a particular object is "a local receiver for all objects in the process," has nothing to do with whether other distinct objects also satisfy that criteria. Significantly, HTC's proposed construction never mentions "unicity," nor does it have any language requiring a "one-to-one" relationship between a proxy and the object it represents.

Importantly, a unicity requirement is inconsistent with the specification of the '721 patent. HTC and the Staff ignore that the single reference in the specification to "unicity" relates only to a specific set of circumstances – "[w]hen an object is [*sic*] passed by reference is decoded." JX-1 at col. 10, lns. 57-60. The '721 patent, however, explicitly says that an object need not be passed by reference: "*[i]n the present invention*, an implementation of an object class is free to choose to implement a different encoding scheme*, for example one that encodes the object by value*." JX-1 at col. 12, lns. 64-66 (emphasis added). Likewise, the '721 patent states that not all proxies are created in the decoding step: "The creation of the *first proxy is provided automatically in the present invention."* JX-1 at col. 11, lns. 66-67 (emphasis added). Notably, these alternative embodiments for passing objects and creating proxies – with no mention of unicity – are described as part of "the present invention." JX-1 at col.11, lns. 66-67; col. 12, lns. 64-66. This undercuts HTC's argument that unicity is required because the single embodiment HTC focuses on is described as "the present invention." HTC Br. at 159. Embodiments with, and without, "unicity" are "the invention."

Additionally, the Staff does not explain why unicity arises out of the construction it proposes. Also, HTC's inaugural effort in its Post-hearing brief is conclusory and unclear, as it must be inasmuch as "one-to-one" and "unicity" are simply not required by the words proposed.

Under HTC's and the Staff's construction, an object is a proxy if it is "***a*** local receiver for all objects in the process." While HTC simply assumes that only one object can be "a local receiver" for all other objects in the process, this is wrong. HTC Br. at 158. Likewise, HTC states that "one proxy ('a proxy object') is created to represent a remote object, and that one proxy ('***this***' proxy – which is '***an*** object,' not '***several objects***') represents the remote object." HTC Br. at 158 n.36. Those words do not appear in the claims or HTC's construction.

In sum, whether one proxy object is "a local receiver for all objects" has nothing to do with whether another proxy object is also "a local receiver for all objects." Indeed, Ms. Spielman explained why a system might have multiple objects that act as local receivers for all objects in the local process – to perform "load balancing." Spielman Tr. 2751.

HTC and the Staff also assert that Ms. Spielman believed that unicity is required by the claims, referring to statements made in her opening expert report. HTC Br. at 158; Staff Br. at 81. HTC's and the Staff's argument that Ms. Spielman agrees with their construction is without merit.

In that regard, the Staff's assertion that "Apple's expert[] originally agreed that the claims require 'unicity' of proxies" is based on a reference in Ms. Spielman's opening expert report to a "one-to-one relationship." Staff Br. at 81. This statement was discussed extensively at the hearing, through direct and cross-examination, and Ms. Spielman repeatedly testified that she did not, and had never, believed that "unicity" was a requirement of the claims. *See* Spielman Tr. 2952. Similarly, when asked whether "the concept of unicity was in the claims," Ms. Spielman explicitly stated that "that term [unicity] is not required, nor is the functionality part of the claim." Spielman Tr. 2749.

Accordingly, the claim term "proxy" is construed to mean "an object that acts as a local receiver for objects in the local process on behalf of another object."

3. "operating system based message"

Claim Term	Apple Construction	HTC Construction	Staff Construction
"operating system based	Plain and ordinary meaning:	Plain and ordinary meaning:	Plain and ordinary meaning:

message"	A message that is based, or dependent, on an operating system	Data sent by an operating system	A message that is system-dependent

Apple Br. at 158; HTC Br. at 162;[55] Staff Br. at 82.

As the Staff explains, "All parties originally agreed that this term should be given its plain and ordinary meaning. However, it is clear that the parties' experts disagree on exactly what that plain meaning is." Staff Br. at 82. Indeed, that is the case.

As proposed by Apple (and in accord with the Staff's proposal), the claim term "operating system based message" is construed to mean "a message that is based, or dependent, on an operating system." The plain language of the claim term "operating system based message" indicates a message that is based on an operating system or, in other words, dependent on an operating system.

The specification of the '721 patent describes a Mach message as an example of the claimed "operating system based message." JX-1 at col. 11, lns. 12-16 ("In the preferred embodiment of the present invention, the proxy 904 encodes the message, (which is a language based message such as, for example, an objective C message), as an operating system message, such as a Mach message 907"). Consistent with the specification, Ms. Spielman explained that Mach messages have a specific message format that is based on, and understood by, the Mach operating system. Spielman Tr. 2852, 2855.

[55] HTC states, "HTC proposes that 'operating system based message' is a broad term with a simple plain and ordinary meaning: a message based on an operating system." HTC, however, also inconsistently states that "the operating system based message is data sent by the operating system." *See* HTC Br. at 162.

Moreover, that the claim term "operating system based message" means "a message that is based, or dependent, on an operating system," is supported by the '721 patent's file history. During prosecution, the Examiner rejected certain claims over a reference ("McCullough") relating to a Distributed SmallTalk system. JX-7 at APP_HTC00008203-10 ('721 File History, June 2, 1994 Office Action). In rejecting the claims, the Examiner equated the "operating system based message" with a "system-dependent" form of a message, which he claimed to find in McCullough's use of an "Ethernet packet." *Id*; Spielman Tr. 2783-84.

In responding to this rejection, the Applicants adopted the Examiner's understanding that an "operating system based message" was system based or "system-dependent." The Applicants then argued that the standardized Ethernet packets identified by the Examiner were not, in fact, "system-dependent," as required by the claims:

> Regarding the McCullough reference, ***Applicant contends that it does not teach a system-dependent form of the message. In fact, McCullough teaches away from a system-dependent form of the message. . . . An Ethernet package is not system dependent. Ethernet is a standardized communication protocol.*** It is therefore not system-dependent. There is nothing in McCullough that describes the encoding of an object oriented programming language based message into an operating system based message as provided in claim 1. . . .

JX-7 at APP_HTC00008240 ('721 File History, Aug. 23, 1994 Amendment and Response at 10); *see* Spielman Tr. 2784-85.

The Examiner agreed that this argument overcame his rejection over McCullough. JX-7 at APP_HTC00008269 ('721 File History, Dec. 22, 1994 Office Action at 5);

Spielman Tr. 2785.[56] Thus, the prosecution history shows that both the Applicants and the Examiner agreed that an "operating system based message" must be system-based, or "system-dependent."

Additionally, as Ms. Spielman explained, messages that are based on an operating system have a format that is understood by, or dependent on, that system. Thus, whether a message is understood by an operating system is evidence of whether it is an operating system based message. Spielman Tr. 5066-67.

HTC's attempt to equate a "message" with any "data" that is sent by an operating system is incorrect. As explained by Inventor Avadis Tevanian, one of the developers of the Mach operating system, not all operating systems are message based. RX-6558C, Tevanian Dep. at 281. Message-based operating systems receive data in discrete, well-defined "messages;" non-messaging systems can simply receive unrecognized data with no known beginning or end and undefined contents. Spielman Tr. 5067-68.

HTC's proposed construction of "message" would cover data that is not a message. HTC claims that this term covers generic protocols such as the User Datagram Protocol ("UDP") which, as Ms. Spielman explained, are specifically designed to be system-independent. Spielman Tr. 5069-70. If the Applicants had wanted to claim any "data sent by an operating system," they could have done so. Instead, they used the narrower "operating system based message."

HTC also accuses Apple of changing its claim construction position. HTC Br. at

[56] The Office Action stated that this rejection had been overcome by "amendments." However, the Applicants had not made any amendments relevant to the McCullough rejection. The only change to claim 1 – changing the word "providing" to "transmitting" – had nothing to do with the McCullough rejection. See Spielman Tr. 2785.

163-164. This accusation is not supported by the record evidence. What HTC presents as Apple's "self-serving shifting proposals" are various forms of evidence that something is, or is not, based or dependent on an operating system. Thus, the ability of an operating system to understand a message is evidence that a message is an "operating system based message." Spielman Tr. 5066-67. Similarly, that a protocol is "standardized," is evidence that it is not based on, or dependent upon, an operating system. Spielman Tr. 2783-85, 5069-70.

Furthermore, although an operating system based message is not restricted to messages unique to a single operating system, evidence that a particular format is used in a heterogeneous environment, namely one with multiple different operating systems, suggests that the message is not based on an operating system. Spielman Tr. 2973-74, 5079-81. Thus, the evidence shows that Apple has consistently argued that "operating system based message" should receive its plain and ordinary meaning – *i.e.,* "a message that is based, or dependent, on an operating system."

Next, HTC asserts that the file history supports its position because the patent Examiner's multiple rejections over prior art that disclosed operating system based messages emphasized the "primitives" those systems used to send messages. HTC Br. at 162-163. HTC contends that the file history shows that the rejection over McCullough was overcome by amendments, not by arguments, and Apple's proposed construction is therefore inconsistent with the disputed term's plain and ordinary meaning. *Id.* at 164.

HTC's file history argument is without merit. HTC's contention that the "file history shows that the rejection over McCullough was overcome by amendments, not by arguments" is wrong because (as noted earlier) there were no amendments made to overcome the rejection. Indeed, the response in question made only two changes to claim 1 – both swapping the word "transmitting" for "providing." JX-7 at APPHTC_00008231-32. These changes explicitly responded to an indefiniteness rejection that "the steps involving 'providing' are unclear" and had nothing to do with the prior art rejection over McCullough. JX-7 at APPHTC_00008205, 8238.

Likewise, while HTC focuses on a sentence in the file history that "Applicant's arguments ... have been fully considered but they are not deemed to be persuasive," was not directed at the McCullough reference. JX-7 at APPHTC_00008286. The "not deemed to be persuasive" remark related to the arguments with respect to the Bennett reference. *Id.* at APPHTC_00008270. It cannot reasonably be disputed that the Applicant's response to the McCullough rejection was "persuasive," as the Examiner concluded that the rejection "has been overcome." *Id.* at APPHTC_00008269.

Accordingly, the claim term "operating system based message" is construed to mean "a message that is based, or dependent, on an operating system."

4. "dynamic binding"

Claim Term	Apple Construction	HTC Construction	Staff Construction
"dynamic binding"	Permitting messages to be bound to the actual methods to be invoked during runtime	Binding messages to the actual methods to be invoked depending on the class of the receiver, allowing objects of any class to be substituted for the target object at runtime	Permitting messages to be bound to the actual methods to be invoked depending on the class of the receiver during execution of a program

Joint Claim Construction, App'x A at 10.[57]

Similar to the construction proposed by HTC, the claim term "dynamic binding" is construed to mean "permitting messages to be bound to the actual methods to be invoked depending on the class of the receiver, allowing objects of any classes that implement a given method to be substituted for the target object at run time."

The specification of the '721 discloses the following with respect to "dynamic binding" of messages:

> One feature of objective C is "dynamic binding" of messages to the actual methods to be invoked, depending on the class of the receiver. A programmer writing code in objective C can create code that sends a message "doSomething" to an object. The actual method corresponding to the class of the target object does not need to be determined until the message must be sent. This allows objects of any classes that implementing the doSomething method to be substituted for the target object at run time without having to modify the part of the program that sends the message. Also, in objective C, programs have run time access to method "signatures," that encode a method's argument and return types for each class. The method signature provides a way for two programs to agree on the format of messages. Moreover, there is a way to extract arguments from the stack using the signature.

JX-1 at col. 8, lns. 29-45 (emphasis added).

Thus, the specification shows that "dynamic binding" of messages is integrally linked to "the actual methods to be invoked." The specification confirms the "actual method corresponding to the class of the target object does not need to be determined until the message must be sent." The specification also shows that this delayed determination of the actual method "allows objects of any classes" that implement a

[57] The Staff did not address this claim term in its briefs.

given method "to be substituted for the target object at run time."

Additionally, a technical treatise cited in the '721 patent also discusses when the binding is done in dynamic binding. Immediately before its only discussion of the term "dynamic binding," the specification references a textbook by Brad Cox:

> The preferred embodiment of the present invention implements an object-oriented programming system using objective-C language. . . . This language model is partially derived from SmallTalk and has been described in "Object-Oriented Programming: An Evolutionary Approach," Brad J. Cox, Addison-Wesley 1986.

JX-1 at col. 8, lns. 18-27 (emphasis added).

The Cox textbook provides insight into the meaning of the claim term "dynamic binding." The textbook states: "[d]elayed binding (also known as late binding or dynamic binding) means that binding is done later than compile-time, generally when the program is running." CX-780 at 13; Rinard Tr. 4621.

Additionally, statements by NeXT (the assignee of the '721 patent and a party to this investigation) provide further guidance relating to the claim term "dynamic binding." A NeXT publication explained the distinction between "late binding" systems, which use constrained languages, and "dynamic binding," which uses unconstrained languages. RX-84 at BOYNTON000001159 (NeXTSTEP Object-Oriented Programming and the Objective C Language (Release 3, 1990) at 21).

Specifically, NeXT defined "dynamic binding" as an "unconstrained" form of binding that was free of "strict compile-time type constraints." *Id.* This is distinct from "late binding" that "carries with it strict compile time type constraints." *Id.*[58] NeXT's definition of "dynamic binding" is consistent with the '721 patent specification, which similarly describes an unconstrained binding system that "allows objects of any classes" that implement a given method "to be substituted for the target object at run time." JX-1 at col. 8, lns. 29-45; Rinard Tr. 4452-55.

Apple argues that "the NeXTSTEP book actually provides a definition for dynamic binding in its glossary, where it defines 'dynamic binding' nearly identically to Apple." Apple Br. at 162. The NeXTSTEP book defines "dynamic binding" in its glossary as follows: "Binding a method to a message – that is, finding the method implementation to invoke in response to the message – at runtime, rather than at compile time." RX-84 at 229.

Apple's argument falls short, however. This definition in the NeXTSTEP glossary does not negate what the very same book states on page 21: "As discussed here (and implemented in Objective C), 'dynamic binding' is unconstrained." As noted above, this explanation by NeXTSTEP is consistent with the '721 patent specification, which similarly describes a "dynamic binding" system that "allows objects of any classes" that implement a given method "to be substituted for the target object at run time."

[58] As seen from the discussions in the Cox and the NeXTSTEP publications, there is a conflict between the two publications. On the one hand, Cox states that "late binding" is synonymous with "dynamic binding." NeXTSTEP, however, explicitly states that while "late binding" is "'dynamic' in the sense that it happens at run time, it carries with it strict compile time type constraints. As discussed here (and implemented in Objective C), 'dynamic binding' is unconstrained." Thus, NeXTSTEP makes clear that "dynamic binding" is unconstrained.

Accordingly, the claim term "dynamic binding" is construed to mean "permitting messages to be bound to the actual methods to be invoked depending on the class of the receiver, allowing objects of any classes that implement a given method to be substituted for the target object at run time."

B. Infringement

The accused HTC devices do not contain the required limitations "first processing means" or "second processing means" or "dynamic binding" of independent claim 1, as properly construed. Thus, the devices do not infringe claim 1. The devices likewise do not infringe claims 5 or 6, which depend from claim 1, and which likewise require these limitations.

The preamble of claim 1 recites:

> **A method for sending an object oriented programming language based message having dynamic binding from a first object in a first process to a second object in a second process, said method comprising the steps of:**

Apple has not satisfied the preamble because accused HTC devices do not support the required "dynamic binding."

Android does not support the "dynamic binding" required by claim 1. Android is based on the Java and C++ programming languages, which do not support dynamic binding, unlike Smalltalk or Objective C. Rinard Tr. 4478-79. At compile-time, these languages constrain the types of messages that can be sent to a given object while a program is running to a predefined set. Rinard Tr. 4450-52, 4478-79.

For example, if a programmer writes code that specifies that it will send a "fly" message to an airplane, the compiler will not allow a bird object to be substituted for the airplane object while a program is running, even though the bird object can understand the "fly" message. For the compiler to accept such a substitution, the programmer would have to modify and recompile the source code to specify that the "fly" message could be sent to a bird object. With dynamic binding, while a program is running, the program could send the "fly" message to any object that implements the "fly" message, including a "bird" object, with no need for a programmer to modify the code. *Id.*

[

]

In addition, Apple failed to show that Android meets the "dynamic binding" limitation under the doctrine of equivalents. Constrained languages such as Java and C++ are substantially different from unconstrained languages that support dynamic binding. Rinard Tr. 4479-80. Programs written in languages with strict compile-time constraints are not interchangeable with those written in unconstrained languages. *Id.* A compiler designed for a language that uses type constraints would not be able to properly type check a program written for an unconstrained language. *Id.*

Moreover, the function of the dynamic binding of these unconstrained languages is substantially different from the constrained binding in Android. Unconstrained languages allow a first object being sent a message to be substituted with any other object that implements a method corresponding to that message (*e.g.*, having a compatible method signature). Android's substitution is constrained by the compile time requirement that the substituted object be in the same class hierarchy or declare the same Java interface as the target object. The way in which the binding occurs is substantially different because in unconstrained languages, the compiler is not involved in the binding of these messages to the target object. The result is substantially different because with unconstrained languages that support dynamic binding, the programmer is afforded more flexibility, but there is an increase in overhead and a higher chance for errors. *Id.*

The first, second, and third elements of claim 1 recite:

> **transmitting, using a first processing means, said object oriented programming language based message to a first proxy in said first process;**
>
> **using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;**
>
> **transmitting said operating system based message to said second process in said second processing means at run time;**

Apple has not satisfied these claim elements because accused HTC devices do not contain the required "first processing means" and "second processing means."

The accused HTC devices do not contain the "first processing means" or the "second processing means." The accused Android devices lack structure that is identical or equivalent to the required "forward::" method. Dianne Hackborn, the Google engineer who first developed Binder, confirmed that Android lacks the features that "forward::" provides – *i.e.*, no code generation and no predefined set of messages. Android's Binder implementation generates code to create predefined interfaces that only handle predefined messages. Hackborn Tr. 4263-64, 4277-78, 4288-89. These generated pieces of code are called "stubs." *Id.*, Tr. 4280-81. Android "stubs" assign a number to each type of message they understand, and the "stub" Android generates can only understand messages corresponding to the numbers assigned when that "stub" was first generated. *Id.*, Tr.4286

Thus, Android's Binder implementation cannot dynamically learn how to send new types of messages. *Id.*, Tr. 4288-89. Ms. Hackborn illustrated this "stub" approach to predefining a set of messages; Android offers an AIDL tool that takes an input file defining a protocol for communication, uses that protocol to assign numbers to each type of message in the protocol, and then generate "stubs" that use the assigned numbers to identify specific methods. RX-5806. Ms. Hackborn's testimony shows that Android uses the very "prior art approach" for IPC that the patent specifically teaches away from; Android "specif[ies] protocols" that identify a fixed set of messages that will be sent between processes, and then "generat[es] . . . stubs" that will send and receive that fixed set of messages. JX-1 at col. 3, lns. 41-51.

Dr. Rinard concurred with Ms. Hackborn's testimony, concluding that Android lacks the features of "forward::" described in the patent. Rinard Tr. 4462-63, 4456-57. In Android, there is no single general-purpose encoding algorithm equivalent to "forward::" that can learn new messages. Android generates multiple pieces of "stub" code that can only encode the messages predefined in a particular interface. *Id.*, Tr. 4458-61; RX5806 at 1, RX6454 at 5-6.)[59]

[59] Moreover, Android has no test to determine whether a proxy implements a method, and therefore does not use this test to trigger an all-purpose encoding algorithm for every message that a proxy receives. That mechanism is described at element 502 of FIG. 5 and is used in the patent to forward all messages that the proxy receives. In contrast,

Apple's expert, Ms. Spielman, agreed that Android does not use a forward:: method or its equivalent, because it has no general-purpose algorithm to encode and transmit arbitrary messages, but instead uses predefined message interfaces. She testified:

> Q. Now, you don't have a view on whether Android performs steps 502 or 504 [invoke "forward::"], do you?
>
> A. From the code I have examined, ***it does not perform those because the interfaces are predefined***.

Spielman Tr. 2929; *see id.* at 2930 ("If you are talking about the invoke forward::, that's not a method that is implemented in Java."). Ms. Spielman likewise agreed that Android does not test for whether an object implements methods in the manner described in FIG. 5, element 502 – a test that is not necessary in constrained language like Java and C++. Spielman Tr. 2929-31; Rinard Tr. 4462-63.

Dr. Rinard concurred with these conclusions. Dr. Rinard testified that his noninfringement opinions were based on Android's lack of structure for "forward" – related functions performed by the "first processing means" in claim 1. Rinard Tr. 4564-65, 4591. Thus, the evidence establishes that the accused devices do not infringe.

Accordingly, Apple has not satisfied the "first processing means" or the "second processing means" limitation of the first, second, or third element of claim 1. Additionally, Apple has not satisfied the "second processing means" limitation of the fifth element of claim 1 for the same reasons.

In sum, the accused HTC devices do not infringe claim 1 because Apple cannot

Android's proxies implement every message they receive, and Android's complier guarantees that an implementation of the message will exist in the proxy. Rinard Tr. 4461-63.

show that (1) the accused HTC devices support the required "dynamic binding, and (2) the accused HTC devices satisfy the "first processing means" or the "second processing means" limitation as required by multiple elements of claim 1. Dependent claims 5 and 6 depend from and contain all the limitations of claim 1. Inasmuch as Apple is unable to show that the accused HTC devices infringe independent claim 1, it cannot show that accused products infringe dependent claim 5 and 6. *Jeneric/Pentron, Inc. v. Dillon* Co., 205 F.3d 1377, 1383 (Fed. Cir. 2000); *Becton Dickinson*, 922 F.2d at 798; *Wahpeton Canvas Co. v. Frontier, Inc.*, 870 F.2d 1546, 1553 (Fed. Cir. 1989).

C. Technical Prong of the Domestic Industry Requirement

Apple argues that "[t]he undisputed evidence establishes that a MacBook Pro running Mac OS X v10.6 ('OS X') practices claim 1, using the preferred embodiment of the '721 patent: Objective-C and Mach messages." Apple Br. at 177.

HTC did not brief the technical prong of the domestic industry requirement.

The Staff submits that "Apple did not prove that the technical prong is met under Staff's constructions" and that "Apple did not even put forth evidence on this point." Staff Br. at 89.

As argued by the Staff, Apple has not satisfied the technical prong. Inasmuch as the undersigned has adopted HTC's and the Staff's proposed construction for the "first processing means" and the "second processing means" limitations, Apple has failed to prove that it practices any of the asserted claims of the '721 patent.

For example, Apple argued that "under Apple's claim construction, a MacBook Pro has the claimed 'processing means', a processor." Apple Br. at 178, citing Spielman Tr. 2845-46; *see* CX-802. The evidence cited by Apple shows that Apple indeed relied on a "processor" to meet the first and second "processing means" limitations rather the properly construed structures for the means-plus-function term.

Accordingly, Apple failed to prove that it practices any of the asserted claims of the '721 patent.

D. Validity

HTC asserts that under Apple's claim constructions, claim 1 is invalid as anticipated by the Bennett Thesis and that claim 1 is also invalid as obvious under Apple's construction over the Bennett Thesis in light of Mach (as described in the Jones and Anderson references). HTC Br. at 179. HTC further asserts that dependent claims 5 and 6 are obvious in light of the Bennett Thesis in combination with additional references, including the Nelson Thesis and ANSA, as well as invalid as obvious separate from any specific combination under *KSR Int'l.*, 550 U.S. 398. *Id.*; HTC Reply at 114 (arguing that Bennett discloses "a first and second processing means under Apple's construction—*i.e.*, a processor" but presenting no arguments under HTC's and Staff's construction of first and second "processing means").

The Staff submits that "[b]ased upon the Staff's claims constructions, the asserted claims of the '721 patent would not have been obvious in light of the Bennett Thesis (or "Distributed SmallTalk")." The Staff explains that "HTC does not argue that any references anticipate the '721 patent under the Staff's constructions" and that "the Bennett Thesis, alone or in combination, does not disclose a first or second 'processing means' under Staff's construction." The Staff states that "HTC's expert never performed an analysis of whether the first and second processing means, as defined by the Staff, would have been obvious based on Bennett." Staff Br. at 89-90.

Based on the arguments presented by HTC and the Staff, it is clear that HTC is not asserting invalidity under HTC's claim constructions. The undersigned agreed with HTC's and the Staff's proposed construction for the "first processing means" and the "second processing means" limitations.

Accordingly, HTC has not shown by clear and convincing evidence that the Bennett Thesis anticipates asserted independent claim 1 of the '721 patent. For the same reason, HTC has not shown by clear and convincing evidence that the combination of the Bennett and Mach references renders obvious independent claim 1 of the '721 patent. Finally, for the same reason, HTC has not shown by clear and convincing evidence that the combinations of (1) Bennett in view of ANSA; (2) Bennett in view of Nelson; and (3) Bennett in view of "common sense" renders obvious dependent claims 5 and 6 of the '721 patent.

XI. Remedy and Bond

A. Limited Exclusion Order

The Commission has broad discretion in selecting the form, scope, and extent of the remedy in a section 337 proceeding. *Viscofan, S.A. v. United States Int'l Trade Comm'n*, 787 F.2d 544, 548 (Fed. Cir. 1986).[60] A limited exclusion order ("LEO") directed to respondents' infringing products is among the remedies that the Commission may impose. *See* 19 U.S.C. § 1337(d). In this investigation, Apple seeks a limited

[60] In determining whether to issue an exclusion order or a cease and desist order, the Commission must consider statutory public interest factors. *Certain Ground Fault Circuit Interrupters and Products Containing Same*, Inv. No. 337-TA-615 ("*GFCIs*"), Comm'n Op. at 21 (Mar. 26, 2009).

exclusion order.

Apple asserts that the Commission is required to issue limited exclusion orders against respondents if they are found to have imported, sold for importation, or sold after importation infringing articles and should cover all such infringing articles and that the orders should also apply to respondents' "affiliated companies, parents, subsidiaries, or other related business entities, or their successors or assigns." Apple Br. at 194.

The scope of an ITC investigation is defined by the notice of investigation. *Certain Chemiluminescent Compositions*, Inv. No.337-TA-285, Commission Order (Jan. 13, 1989). Thus, any exclusion order may cover all products within that scope, *i.e.*, "the articles concerned." 19 U.S.C. § 1337(d)(1). Moreover, Commission remedial orders cover all products that infringe and are not limited to specified models or products. *Certain Optical Disk Controller Chips and Chipsets and Products Containing Same, Including DVD Players and PC Optical Storage Devices*, Inv. No. 337-TA-506, Commission Opinion at 56 (August 7, 2006) (public version).

As noted, the Commission instituted this investigation "to determine whether there is a violation of subsection (a)(1)(B) of section 337 in the importation into the United States, the sale for importation, or the sale within the United States after importation of *certain personal data or mobile communications devices or related software* that infringe one or more of claims [of the asserted patents] and whether an industry in the United States exists as required by subsection (a)(2) of section 337." 75 Fed. Reg. 17434 (2010) (emphasis added).

It is, therefore, recommended that the Commission issue a limited exclusion order in the event that a violation of section 337 is found. Such a limited exclusion order should be directed to *certain personal data or mobile communications devices or related software* of this investigation which have been found to infringe the '263 and '647 patents. This limited exclusion order remedy, of course, is not limited to any representative list of accused products. Rather, it encompasses all personal data or mobile communications devices or related software that infringe the '263 and '647 patents. Additionally, in order to facilitate the limited exclusion order's proper enforcement, it should contain a certification provision.[61]

B. Cease and Desist Order

Section 337 provides that in addition to, or in lieu of, the issuance of an exclusion order, the Commission may issue a cease and desist order as a remedy for a violation of section 337. 19 U.S.C. § 1337(f)(1). The Commission "generally issues a cease and desist order only when a respondent maintains a commercially significant inventory of infringing products in the United States." *GFCIs*, Comm'n Op. at 24. The fact that a respondent is a foreign entity does not prevent the issuance of a cease and desist order against it. *See Certain Abrasive Products Made Using a Process for Powder Preforms, and Products Containing Same*, Inv. No. 337-TA-449, 67 Fed. Reg. 34728, Comm'n Notice (May 15, 2002) (issuance of limited exclusion order, and cease and desist order against Taiwan respondent) (vacated on other grounds, 69 Fed. Reg. 35675 (2004)).

[61] An exclusion order may contain a provision that permits entities whose products are potentially excludable under the Commission's order to certify, pursuant to procedures to be specified by U.S. Customs and Border Protection, that they are familiar with the terms of the order, that they have made appropriate inquiry, and thereupon state that, to the best of their knowledge and belief, the products being imported are not excluded from entry under the order. *Certain Semiconductor Chips with Minimized Chip Package Size or Products Containing Same*, Inv. No. 337-TA-605, Comm'n Op. at Section II.D.2. (July 29, 2009).

Apple submits that a cease-and-desist order should also issue against HTC because it maintains a commercially significant level of inventory within the U.S. Apple Br. at 194.

HTC contends that "HTC's inventories of accused products in the U.S. are for testing purposes only, are not approved by the U.S. government, are not for sale, and therefore cannot by definition be 'commercially significant'" and that "HTC surrenders all title and interest to its commercial products when they arrive and are warehoused in the United States." HTC Br. at 189.

The Staff agrees with HTC, asserting that "[t]he evidence fails to support that HTC maintains commercially significant inventory in the United States" and that "HTC does maintain a small number of Android devices in the United States, which it uses for testing and marketing purposes, but does not store thousands of devices as Apple has implied." Staff Br. at 98.

The record evidence establishes that HTC's inventories of accused products in the U.S. are for testing purposes only, are not approved by the U.S. government, and are not for sale. Mackenzie Tr. 938-39. Moreover, the evidence shows that HTC surrenders all title and interest to its commercial products when they arrive and are warehoused in the United States. *Id.* at 924. Apple has failed to carry its burden of proof on this issue.

Accordingly, in the event that a violation of section 337 is found, it is not recommended that the Commission issue a cease and desist order.

C. Bond

The administrative law judge and the Commission must determine the amount of bond to be required of a respondent, pursuant to section 337(j)(3), during the 60-day

Presidential review period following the issuance of permanent relief, in the event that the Commission determines to issue a remedy. The purpose of the bond is to protect the complainant from any injury. 19 U.S.C. § 1337(j)(3); 19 C.F.R. 210.42(a)(1)(ii), 210.50(a)(3).

When reliable price information is available, the Commission has often set the bond by eliminating the differential between the domestic product and the imported, infringing product. *Certain Microsphere Adhesives, Processes for Making Same, and Products Containing Same, Including Self-Stick Repositionable Notes*, Inv. No. 337-TA-366, Comm'n Op. a 24 (1995). In other cases, the Commission has turned to alternative approaches, especially when the level of a reasonable royalty rate could be ascertained. *Certain Integrated Circuit Telecommunication Chips and Products Containing Same, Including Dialing Apparatus*, Inv. No. 337-TA-337, Comm'n Op. at 41 (1995). A 100 percent bond has been required when no effective alternative existed. *Certain Flash Memory Circuits and Products Containing Same*, Inv. No. 337-TA-382, USITC Pub. No. 3046, Comm'n Op. at 26-27 (July 1997) (a 100% bond imposed when price comparison was not practical because the parties sold products at different levels of commerce, and the proposed royalty rate appeared to be *de minimis* and without adequate support in the record).

Apple asserts that "[a] bond of 100% of the entered value of the infringing articles is appropriate with respect to all Respondents because a direct price comparison between Complainants' domestic products and the Respondents' infringing products is impractical, and no reliable royalty rates from licenses are available." Apple Br. at 194-195.

HTC contends that Apple bears the burden of proof and that Apple presented no evidence on the method of bond calculation. Thus, HTC submits that the bond should be set at zero. HTC Br. at 189-190.

The Staff agrees with HTC, arguing that "it does not appear that Apple presented any evidence that there is no adequate royalty information or that due to the varying prices of the Accused HTC [ ] Products a price comparison is not practical." The Staff states that "in such cases, the Commission has not imposed a bond." Staff Br. at 99.

As HTC and the Staff argued, Apple has failed to carry its burden of proof on this issue. Apple states that "reliable price differentials cannot be calculated as numerous handset models are accused for [ ] HTC [ ], all of which are sold at varying prices through multiple sales channels" and that "HTC's accused handsets are sold at various prices ranging from $270 to $430." Apple Br. at 195. Apple cites to *Unified Comm. Sys.*, 337-TA-598, Initial Determination, 2008 WL 683369 (Jan. 28, 2008), in which Chief Judge Luckern recommended a bond in the amount of 100% of entered value "in view of the wide ranges of prices for infringing products."

Apple's argument is rejected. This is not a case where "it is impossible for the Commission to calculate what level of bond based on price differentials will protect a complainant from any injury." *See* Apple Br. at 195 n.56. For example, the maximum price differential can be calculated as 37% (using the price range of $270 - $430) in this investigation. Instead, this is a case where Apple did not put in evidence on the bond at the hearing and is attempting to remedy this failure by citing (post-hearing) to a different investigation (*Unified Comm. Sys.*) whose facts are different from the circumstance of the present investigation.[62]

Accordingly, in the event that a violation of section 337 is found, it is recommended that respondents not be required to post a bond during the Presidential review period.

XII. Conclusions of Law

1. The Commission has subject matter, personal, and *in rem* jurisdiction in this investigation.

2. The importation requirement is satisfied as to all respondents.

3. Respondents' accused products infringe asserted claims 1, 2, 24, and 29 of the '263 patent and asserted claims 1, 8, 15, and 19 of '647 patent.

4. Respondents' accused products do not infringe claim 3 of the '647 patent or the asserted claims of the '983 and the '721 patents.

5. It has not been shown by clear and convincing evidence that any asserted claim of the '263, '647, '983, and the '721 patents is invalid.

6. The domestic industry requirement is satisfied with respect to the '263 and the '647 patents.

7. The domestic industry requirement is not satisfied with respect to the '983 and the '721 patents.

[62] In that regard, the complainant in *Unified Comm. Sys.* had questioned a fact witness regarding product pricing. Apple, in contrast made no such attempt here. *See Certain Liquid Crystal Display Devices and Products Containing the Same*, Inv. No. 337-TA-631, Comm'n Op. at 28 (July 15, 2009) (complainant "should not benefit from a lack of any effort" and "is required to do more than just assert a lack of 'meaningful price comparison'.").

8. There is a violation of section 337.

XIII. Initial Determination and Order

Accordingly, it is the INITIAL DETERMINATION of the undersigned that a violation of section 337 (19 U.S.C. § 1337) has occurred in the importation into the United States, the sale for importation, or the sale within the United States after importation of certain personal data or mobile communications devices or related software with respect to asserted claims 1, 2, 24, and 29 of U.S. Patent No. 6,343,263 and asserted claims 1, 8, 15, and 19 of U.S. Patent No. 5,946,647. A violation of section 337 has not occurred with respect to asserted claim 3 of the '647 patent or the asserted claims of U.S. Patent Nos. 6,275,983 and 5,481,721. Finally, it is the initial determination that U.S. Patent No. 6,343,263, U.S. Patent No. 5,946,647, U.S. Patent No. 6,275,983, and U.S. Patent No. 5,481,721 are not invalid.

Further, this Initial Determination, together with the record of the hearing in this investigation consisting of:

(1) the transcript of the hearing, with appropriate corrections as may hereafter be ordered, and

(2) the exhibits received into evidence in this investigation, is CERTIFIED to the Commission.

In accordance with 19 C.F.R. § 210.39(c), all material found to be confidential by the undersigned under 19 C.F.R. § 210.5 is to be given *in camera* treatment.

The Secretary shall serve a public version of this ID upon all parties of record and the confidential version upon counsel who are signatories to the Protective Order (Order

No. 1) and the Amended Protective Order (Order No. 12) issued in this investigation, and upon the Commission investigative attorney.

To expedite service of the public version, each party is hereby ORDERED to file with the Commission Secretary by no later than July 27, 2011, a copy of this Initial Determination with brackets that show any portion considered by the party (or its suppliers of information) to be confidential, accompanied by a list indicating each page on which such a bracket is to be found. At least one copy of such a filing shall be served upon the office of the undersigned, and the brackets shall be marked in red. If a party (and its suppliers of information) considers nothing in the Initial Determination to be confidential, and thus makes no request that any portion be redacted from the public version of this Initial Determination, then a statement to that effect shall be filed in lieu of a document with brackets.

Pursuant to 19 C.F.R. § 210.42(h), this Initial Determination shall become the determination of the Commission unless a party files a petition for review pursuant to § 210.43(a) or the Commission, pursuant to § 210.44, orders on its own motion a review of the ID or certain issues herein.

Carl C. Charneski
Carl C. Charneski
Administrative Law Judge

Issued: July 15, 2011

CERTAIN PERSONAL DATA AND MOBILE COMMUNICATIONS DEVICES AND RELATED SOFTWARE **337-TA-710**

PUBLIC CERTIFICATE OF SERVICE

I, James R. Holbein, hereby certify that the attached **INITIAL DETERMINATION** has been served by hand upon the Commission Investigative Attorney Erin D. E. Joffre, Esq, and the following parties as indicated, on AUG 24 2011.

James R. Holbein
U.S. International Trade Commission
500 E Street, SW, Room 112A
Washington, D.C. 20436

FOR COMPLAINANTS APPLE INC., f/k/a APPLE COMPUTER INC., AND NEXT SOFTWARE, INC. f/k/a NeXT COMPUTER:

F. Christopher Mizzo, Esq.
KIRKLAND & ELLIS LLP
655 Fifteenth Street, N.W.
Washington, D.C. 20005-5793

( ) Via Hand Delivery
(✓) Via Overnight Mail
( ) Via First Class Mail
( ) Other: _________

FOR RESPONDENT HIGH TECH COMPUTER CORP. a/k/a/ HTC CORP., HTC AMERICA INC. AND EXEDEA, INC.:

James B. Coughlan, Esq.
PERKINS COIE LLP
700 13th St., N.W.
Suite 600
Washington, D.C. 20005

( ) Via Hand Delivery
(✓) Via Overnight Mail
( ) Via First Class Mail
( ) Other: _________

CERTAIN PERSONAL DATA AND MOBILE COMMUNICATIONS DEVICES AND RELATED SOFTWARE **337-TA-710**

PUBLIC MAILING LIST

Heather Hall
LEXIS-NEXIS
9443 Springboro Pike
Miamisburg, OH 45342

Kenneth Clair
Thomson West
1100 Thirteen Street, NW, Suite 200
Washington, D.C. 20005