United States District Court,
S.D. California.

LUCENT TECHNOLOGIES, INC., and Multimedia Patent Trust,
Plaintiffs.
v.
MICROSOFT CORPORATION,
Defendant.
and Related Claim,
and Related Claims.

No. 06-CV-0684-H (CAB)

Feb. 28, 2008.

David A. Hahn, Hahn and Adema, San Diego, CA, Howard D. Shatz, James E. Marina, John M. Desmarais, Michael P. Stadnick, Paul A. Bondor, Robert A. Appleby, Elizabeth T. Bernard, Jennifer J. Schmidt, Jonas Reale Mcdavit, Jordan N. Malz, Todd M. Friedman, William Fink, Kirkland and Ellis LLP, New York, NY, Carl J. Blickle, Kirkland & Ellis LLP, Chicago, IL, Corey J. Manley, Karen Michelle Robinson, Valerie Gutmann, Kirkland & Ellis LLP, Washington, DC, for Plaintiffs.

Christopher Scott Marchese, Desa L. Burton, Fish and Richardson, San Diego, CA, Erik M. Andersen, Payne and Fears, Irvine, CA, John E. Gartman, John P. Schnurer, Shekhar Vyas, Jaime Olin, John M. Farrell, Joseph Patrick Reid, Juanita R. Brooks, Lara Sue Garner, Fish and Richardson, San Diego, CA, Andrew R. Kopsidas, Fish and Richardson, Washington, DC, Bradley D. Coburn, Conor M. Civins, Roger A. Denning, Ross Spencer Garsson, Fish & Richardson PC, Austin, TX, Cathy L. Reese, Linhong Zhang, Fish & Richardson P.C., Brian M. Rostocki, Raymond N. Scott, Jr., Richard C. Weinblatt, Sean T. O'Kelly, Cross & Simon LLC, Wilmington, DE, John M. Helms, Fish & Richardson, Dallas, TX, for Defendant.

ORDER REGARDING CLAIM CONSTRUCTION AND SUMMARY JUDGMENT MOTIONS FOR U.S. PATENT NOS. 5,838,319 **AND** 5,977,971 (THE GUZAK PATENTS)

MARILYN L. HUFF**, District Judge.**

On January 4, 2008, Lucent brought a motion for summary judgment including requests that the Court determine: (1) that U.S. Patent Nos. 5,838,319 ("Guzak '319") and 5,977,971 ("Guzak '971" and collectively the "Guzak patents") are invalid for obviousness, and (2) that Lucent's VitalSuite product does not infringe claims 12 and 17 of the Guzak '319 patent. (Doc. No. 182.) On February 1, 2008, after full initial briefing by the parties, the Court held a hearing on these and other summary judgment motions. (*See* Doc. Nos. 329, 343.) The Court previously ruled on other motions considered at the hearing, but it deferred questions related to the Guzak patents to allow supplemental briefing on a question of claim construction. (*See* Doc. Nos. 332, 347.) In accordance with the Court's scheduling order, Microsoft filed a supplemental brief on February 15, 2008, and Lucent filed its supplemental brief on February 22, 2008. (Doc. Nos.355, 368.)

Background

I. The Patents

The Guzak patents, which derive from the same original application, are generally directed to computer user interfaces called "tree views" or "tree view controls." These interfaces display a hierarchical list of items, allow the user to expand or collapse nodes in the hierarchy, and allow the user to select items from the list. (*See, e.g.,* Guzak '319 Abstract.) Guzak '319 is entitled "System Provided Child Window Control for Displaying Items in a Hierarchical Fashion." Guzak '971, which issued on a continuation application approximately one year after Guzak '319, is entitled "Tree View Control."

Microsoft asserts four claims from these patents against Lucent's VitalSuite software product: claims 12 and 17 of Guzak '319, and claims 2 and 3 of Guzak ' 971. Claims 12 and 17 of Guzak '319 depend on claims 11 and 16, respectively. Claim 11 of Guzak '319 states:

In a computer system having an output device and an input device, a method comprising the steps of:

displaying a hierarchical tree of items having at least two levels of items on the output device as part of a window control;

in response to a user using the input device, selecting one of the items displayed in the hierarchical tree of items; and

expanding the hierarchical tree of items independently of the selecting so that an additional level of items is displayed as part of the hierarchical tree of items on the output device such that the expanding occurs in response to a user action that does not result in another selection of one of the items.

(Guzak '319 86:3-16.) Claim 12 further limits this as follows:

The method of claim 11 where the computer system includes application programs and wherein the child window control is a system resource for use by the application programs.

(Guzak '319 86:17-20.) Claim 16 states:

In a computer system having an output device and an input device, a method comprising the steps of:

displaying a hierarchical tree of items having at least two levels of items on the output device as part of a child window control;

in response to a user using the input device, selecting one of the items displayed in the hierarchical tree of items; and

collapsing the hierarchical tree of items independently of the selecting so that one of the levels of items of the hierarchical tree that was displayed in the displaying step is no longer displayed as part of the hierarchical tree on the output device in response to a user selection that does not result on another selection of one of the items

(Guzak '319 86:32-47.) Finally, with respect to Guzak '319, claim 17 adds the further limitation:

The method of claim 16 wherein the computer system includes application programs and wherein the child window control is a system resource for use by the application programs.

(Guzak '319 86:48-51.) Moving to Guzak '971, claims 2 and 3 each begin:

A computer-readable medium having computer-executable instructions which, when executed in a computer system having an output device and an input device, perform a method comprising: ...

After this text, claim 2 goes on to recite the same method steps as claim 11 of Guzak '319, while claim 3 goes on to recite the same method steps as claim 16 of Guzak '319. (Guzak '971 87:17-50.)

In summary, claims 11 and 16 of Guzak '319 each define a method, and the corresponding claims of Guzak '971 define a computer-readable medium for performing those methods. Claims 12 and 17 of Guzak '319 further limit these methods to situations where the child window control is a "system resource for use by the application programs."

The Court previously construed certain terms from these claims. A "computer system" is "a system containing one or more computers." (*See* Claim Construction Order, Doc. No. 156 at 63.) The Court construed both "window control" and "child window control" as "computer code, along with an accompanying graphical representation, that sends notification messages to a parent window when events, like user input, occur within the window control." (*Id*. at 64-65.) An "application program" is "a program that puts the resources and capabilities of a computer to use." (*Id*. at 65.) Finally, the Court construed a "system resource for use by the application programs" as "a system resource available for use by any application program in the computer system." (*Id*. at 65.) Currently, however, the Court is considering whether it should clarify the phrase "for use by any application program" as "for use by all application programs." (*See* Order Authorizing Supplemental Briefing Regarding the Guzak Patents, Doc. No. 332.)

Claim 1, though not asserted here, is also relevant to the Court's analysis of the claim language and prosecution history. It states:

In a computer system having a processor for running a first and a second application program and an output device, a method comprising the steps of:

providing a child window control as a system resource that may be used by the first and the second application program to display a list of items as a hierarchical tree on the output device;

using the child window control by the first application program to display at least a portion of a list of items as a hierarchical tree on the output device; and

using the child window control by the second application program to display at least a portion of another list of items as a hierarchical tree on the output device

(Guzak '319 85:2-14.)

II. The Asserted Prior Art and Prosecution History

As indicated by the report of Microsoft's expert, Dale E. Buscaino, the parties do not dispute that general concepts for presenting information in hierarchical trees were well known before the Guzak patents, and even before the advent of computers. (*See* Decl. John Gartman Supp. Microsoft's Opp'n Defs.' Mots. Summ. J. ("Gartman Decl.") Ex. VII-1 ("Buscaino Rpt.") para. 36.) Early computer systems used hierarchical trees in a static form without interactivity. (*See id*. para. 37.) As computer systems enabled increasing interactivity with larger amounts of information, interactive tree views became a viable way of aiding the user through display of hierarchical information. (*See id*. para.para. 38-39.) The patents' specifications also indicate that display of hierarchical lists was known, though not necessarily with all the facets of the claimed invention. (*See, e.g.,* (Guzak '319 1:16-26 (stating that "[i]n conventional systems, if application programs wish to display a hierarchical list of items, they must provide code and data structures for displaying the list" and characterizing the invention as a more efficient alternative to this approach); Guzak '971 1:15-25 (same).) The parties do dispute, however, whether particular aspects of the Guzak patents are obvious in light of the prior art, including the specific capabilities of the tree view control described and the provisioning of that control as a child window control and system resource.

In 1992, Microsoft published *Programming Windows 3.1* ("the Programming Guide"), including Chapter 6 on "Child Window Controls ." (McDavit Decl. Ex. 32.) The Programming Guide describes a purportedly more efficient method for creating child window controls by taking advantage of "predefined window classes (and window procedures)" such as "buttons, check boxes, edit boxes, list boxes, combo boxes, text strings, and scroll bars." (*Id*. at CCMS_ 027646.)

As described in the report of Lucent's expert Nathaniel Polish, XTree for Windows, released in February 1993, is a program designed to run on the Microsoft Windows 3.1 operating system. (Gartman Decl. Ex. VII-2 ("Polish Rpt.") at 20.) XTree was capable of generating hierarchical displays of a computer's directory structure. (*See id*. at 20-21 (including screen captures of XTree in operation).) In the directory structure, a user may select an item and expand and contract parts of the tree without changing which item is selected. (*See id*. at 21-23; *see also* McDavit Decl. Ex. 31 (XTree User's Manual).) FN1 The PTO considered the Programming Guide during prosecution history, but not XTree for Windows.

FN1. The parties' experts discuss other prior art, but since Lucent's motion is based on the Programming Guide and XTree for Windows, the Court does not discuss the other prior art here. (*See, e.g.,* Polish Rpt. at 12-20 (describing alleged prior art from Digital Equipment Corporation and Apple Corporation).)

During prosecution, the PTO also considered the Microsoft Windows 3.1 User's Guide ("Windows 3.1"), initially rejecting claims 12 and 17 (application claims 14 and 20) as anticipated by Windows 3.1 under 35 U.S.C. s. 102(b). (*See* Decl. Desa Burton Supp. Microsoft's Supplemental Briefing Regarding Guzak Patents ("Burton Decl.") Ex. 1 ("'319 File History") at CCMS__027553.) Responding to this rejection, the applicants admitted that the Windows 3.1 file manager allows a user to expand and contract a hierarchical tree, but noted that the user could not do this without changing the selected item, as provided by the claimed invention. (*Id*. at CCMS_027561.) The applicants also distinguished asserted claims 12 and 17, stating that: "The claim also requires that the child window control be a system resource for use by the application programs. The file manager does not use a child window control that is a system resource; rather, the file manager includes its own specialized code for displaying files in a hierarchical fashion." (*Id*. at CCMS_027561-62.)

The PTO also considered U.S. Patent Number 5,361,361 ("the Hickman patent"), entitled "Hierarchical

Integrated Help Utility for a Computer System." (Burton Decl. Ex. 2.) The Hickman patent describes "[a]n improved method and apparatus for hierarchically integrating help information across multiple applications ...." (Hickman Abstract.) In the preferred embodiment of Hickman, an integrated help utility organizes the help file directories of multiple applications and displays them in a hierarchical fashion in a "Bookcase" help window. (*See, e.g.,* Hickman Figs. 3, 5.) The examiner rejected claim 1, which is not asserted here, as anticipated by the Hickman patent under 35 U.S.C. s. 102(b). In response, the applicants argued that:

Hickman et al. does not provide a child window control *as a system resource*.... The Bookcase help window ... is not provided as a child window control that is a system resource for use by any application program. Rather, the Bookcase help window is a separate window produced by a help utility that is not a child window control and is independent of the application programs.

('319 File History at CCMS_0275562-63 (emphasis in original).)

III. Lucent's VitalSuite Product

VitalSuite is a collection of software tools for network and application management. This collection of tools includes VitalSuite NET, or "VitalNET;" VitalSuite Real-time Event Analysis; VitalSuite Advanced Reporting Tool, or "VitalART;" and VitalSuite APPS, or "VitalApps." (*See* Declaration Jonas McDavit Supp. Plfs.' Mot. Summ J. Ex. 28 (excerpt from VitalSuite user's guide).) Microsoft does not dispute the assertion by Lucent's expert that VitalSuite is installed onto, and runs from, servers. (Burton Decl. Ex. 3 at 11.)

According to Microsoft' expert, VitalSuite includes a hierarchical tree display that meets the requirements of the asserted claims, such as the ability to expand or collapse the tree without changing the current selection. (*See* Burton Decl. Ex. 8 ("Buscaino Am.2d Suppl. Rpt.") at Ex. F.) One of Lucent's witnesses has admitted that the various parts of VitalSuite share at least some components. (*See* Burton Decl. Ex. 4. ("Hu Depo. Tr.") at 15:20-16:8.) For example, VitalApps and VitalNet "share a common GUI portal." (*Id*. at 15:23-24.) Microsoft's expert notes that VitalSuite can be installed on a Windows 2003 Server. (Burton Decl. Ex. 9 ("Buscaino 3d Suppl. Rpt.") at 1.) He also asserts that the allegedly infringing tree control is implemented through HTML and JavaScript and therefore available to every application on such a sever:

The tree control for the domain tree in VitalSuite is implemented through HTML and JavaScript. As such, the tree control is available for use by all application programs running on such a sever. As one example, in a Windows-based environment such as the one described above [Windows 2003 Server], all applications designed to operate with Windows have HTML and JavaScript capabilities with the only limitation being design choice. Thus, the tree control provided by VitalSuite will be available for use by these applications.

(*Id*. at 1.)

Microsoft argues that Lucent's expert, Dr. Polish, conceded that "the tree control code from Vital Suite is available to windows-based applications." (*See* Microsoft's Supp. Briefing at 7.) Dr. Polish stated that he was unsure whether the various components of VitalSuite could function together if all installed on the same server. (Burton Decl. Ex. 11 ("Polish Depo. Tr.") 231:9-233:15.) He went on to state that, to the extent VitalSuite applications could function on the same server, they could share tree-related code "to the extent that those applications are using the web browser that's able to use that JavaScript code." (*Id*. 233:16-234:9.) Dr. Polish further clarified that the code "requires a browser or the equivalent functionality of a

browser in order to be useful." (*Id*. 237:14-16.) When asked whether another application could use the code in question outside the browser context, Dr. Polish indicated that "you would have to build a whole [ ]lot of context, including a JavaScript interpreter and an HTML interpreter, which would in fact mean you were building a browser." (*Id*. 240:21-24.) Dr. Polish also indicated that one Windows application can potentially launch another application. (*Id*. 251:3-252:7.) Lucent submitted a supplemental report from Dr. Polish that further clarifies and emphasizes his theory, as presented in the deposition, that VitalSuite is only usable inside a web browser, and any application using VitalSuite would have to be both (1) web-based and (2) hosted on the same computer. (*See* Decl. William C. Mercer ("Mercer Decl.") Supp. Lucent's Supplemental Mem. Ex. 2 ("Polish Rebuttal") at 5.)

Discussion

I. Summary Judgment Standard

Under Rule 56(c) of the Federal Rules of Civil Procedure, a court may grant summary judgment upon a claim "if the pleadings, the discovery and disclosure materials on file, and any affidavits show that there is no genuine issue as to any material fact and that the movant is entitled to judgment as a matter of law." A party moving for summary judgment bears the initial burden of establishing the absence of a genuine issue of material fact for trial. *See* Celotex Corp. v. Catrett, 477 U.S. 317, 323, 106 S.Ct. 2548, 91 L.Ed.2d 265 (1986). The moving party's burden "may be discharged by 'showing'-that is, pointing out to the district court-that there is an absence of evidence to support the nonmoving party's case." Celotex Corp., 477 U.S. at 325.

Once the moving party meets the requirements of Rule 56, the party opposing the motion must set forth specific facts showing that there is a genuine issue of material fact. *See* Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 248-56, 106 S.Ct. 2505, 91 L.Ed.2d 202 (1986). The opposing evidence must be sufficiently probative to permit a reasonable trier of fact to find in favor of the opposing party. *See* id. at 249-50. Thus, the non-moving party cannot oppose a properly supported summary judgment motion by "rest[ing] upon mere allegation or denials of his pleading." Id. at 256. If the non-moving party fails to make a sufficient showing of an element of its case, the moving party is entitled to judgment as a matter of law. *See* Celotex, 477 U.S. at 322-23.

On a motion for summary judgment the court views the evidence in the light most favorable to the non-moving party. United States v. Diebold, Inc., 369 U.S. 654, 655, 82 S.Ct. 993, 8 L.Ed.2d 176 (1962). However, "[w]hen opposing parties tell two different stories, one of which is blatantly contradicted by the record, so that no reasonable jury could believe it, a court should not adopt that version of the facts" for summary judgment purposes. *See* Scott v. Harris, 550 U.S. 372, ----, 127 S.Ct. 1769, 1776, 167 L.Ed.2d 686 (2007).

II. Claim Construction

"It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention ....' " Phillips v. AWH Corp., 415 F.3d 1303, 1312 (Fed.Cir.2005) (quoting Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc., 381 F.3d 1111, 1115 (Fed.Cir.2004)). At issue here is whether, in claims 12 and 17, the "system resource ..." limitation requires that the child window control be for use by all application programs in the computer system. The plain language of these claims supports this conclusion. Both claims describe a situation "where the computer system includes application programs and wherein the child window control is a system resource for use by the application programs." (Guzak '319 86:17-20, 48-51.) The second use of

"application programs" in this excerpt relates back to the first, in which "applications programs" describes whatever the computer system happens to include, without additional limitations. Even the bare term "system resource" indicates a resource related the whole system, not some subset of the system.

Other claims, whether asserted or not, are also "valuable sources of enlightenment as to the meaning of a claim term." Phillips, 415 F.3d at 1314. Microsoft would have the Court interpret "application programs" in accordance with its use in claim 1. There, the patent describes a method, in a system for running two application programs, where the child window control is "a system resource that may be used by the first and the second application programs." (Guzak '319 85:2-8.) Claim 22 also contains a similar limitation. (*See* Guzak '319 87:6-22.) These limitations actually run contrary to Microsoft's position. The patentees' decision to specify the number of application programs in some claims indicates that they intended the more open-ended description of application programs in claims 12 and 17 to be broader, encompassing any number of application programs. (*See* Guzak '319 86:17-20, 48-51.) The patentees proved themselves capable of limiting the number of application programs involved in particular claims, as in claim 1, but they did not do so for claims 12 or 17.

Courts also read the claims in view of the specification, which is not to the contrary here. *See* Phillips, 415 F.3d at 1315-16. The patent states that "[a] child window control is provided as a system resource. This child window control may be used by one or more application programs ...." (Guzak ' 319 1:30-32.) The fact that only one application program might actually use the system resource does not imply that a system resource is not still available for use by the entire system. The specification is therefore consistent with the revised construction.

The prosecution history is also intrinsic evidence that "can often inform the meaning of the claim language by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be." Phillips, 415 F.3d at 1317. The Court concludes that the proposed construction, requiring that a system resource in claims 12 and 17 must be "for use by all application programs," is also consistent with the prosecution history. In distinguishing Windows 3.1, the patentees merely reiterated the language of claims 12 and 17, arguing that the file manager was not a system resource for use by the application programs. (*See* '319 File History at CCMS_ 027561-62.) This approach is consistent with the Court's new construction, though it does not further clarify of the claim language. The applicants also distinguished the Bookcase help window of the Hickman patent on the ground that it was not "for use by any application program." (*See* '319 File History at CCMS-0275562-63.) On its face, the term "any" could mean "one or more," "all," or other possible meanings. The Court therefore concludes that the applicants' use of this phrase is also consistent with the Court's proposed construction. FN2

FN2. The Court's original construction focused on this phrase. Upon further examination, however, the Court's draws its revised construction as much or more from the language of the claims and the specification than from the statements during prosecution history.

In light of all the intrinsic evidence, it is sufficiently clear that the applicants meant for a "system resource," in claims 12 and 17, to be a resource usable by the entire system. Accordingly, the Court concludes that, in the asserted claims, the proper construction of a "system resource for use by the application programs" is "a system resource available for use by all application programs in the computer system."

III. VitalSuite's Alleged Infringement of Guzak '319

A. Legal Standard

"Infringement, whether literal or under the doctrine of equivalents, is a question of fact." Terlep v. Brinkmann Corp., 418 F.3d 1379, 1382 (Fed.Cir.2005). The Court first construes the claim and then compares the claim to the accused product or device. Id. at 1381. Every element of the claim must be present in the accused device or process, either literally or by a substantial equivalent. Id. at 1384. Under the doctrine of equivalents, a "substantial equivalent" is one that "performs substantially the same function in substantially the same way to achieve substantially the same result." *See, e.g.*, Wolverine World Wide, Inc. v. Nike, Inc., 38 F.3d 1192, 1196 (Fed.Cir.1994).

B. VitalSuite Installed Across Multiple Servers

For its infringement analysis, the Court considers two postulated configurations of VitalSuite: (1) installation of VitalSuite components across a group of severs on a network, and (2) installation of VitalSuite components on a single sever. With respect to the first scenario, the Court concludes that VitalSuite does not infringe claims 12 and 17 of Guzak '319. The Court construed "computer system" as "a system containing one or more computers." To the extent that the computer system is a group of servers operating across a network, Microsoft would have to show that the alleged system resource is available to all applications across all servers on the network. Mr. Buscaino's theory, however, is based on the ability of applications on a single server to access the HTML or JavaScript code of VitalSuite on that server. Mr. Buscaino's supplemental infringement analysis, using the alternate construction which the Court now adopts, does not explain how an application on one sever would use the HTML or JavaScript code on a different server. (*See* Buscaino 3d Suppl. Rpt.)

Accordingly, the Court concludes that there can be no infringement under the first theory. Microsoft fails to provide sufficient evidence that VitalSuite provides a system resource "for use by all application programs in the computer system." The Court also concludes that Microsoft does not present a viable theory under the doctrine of equivalents. Finally, the Court notes that the product documentation and other evidence describing VitalSuite indicates that installation across multiple servers is the typical mode of operation, and Microsoft does not dispute this fact.

C. VitalSuite Installed on a Single Server

Microsoft also attempts to maintain an infringement theory based on the installation of VitalSuite components on a single server. Even in this context, the Court concludes that Microsoft has not offered sufficient evidence of infringement to survive summary judgment, even viewed in a light most favorable to Microsoft. The experts agree that the tree code in question is implemented via HTML or JavaScript. Microsoft has not disputed Dr. Polish's statements that an application would need to use a web browser or implement similar functionality to make use of VitalSuite's code. Mr. Buscaino states that "all applications designed to operate with Windows have HTML and JavaScript capabilities with the only limitation being design choice." (*See* Buscaino 3d Suppl. Rpt. at 1.)

Viewed in a light most favorable to Microsoft, the experts' opinions indicate that the VitalSuite code can only be called by an application if the application is designed to make use of HTML and JavaScript code. The code could be available, however, to any appropriately designed applications. This situation does not fit within the Court's claim construction. The patent makes clear that the purpose of the claims was to provide

reusable code that would increase the efficiency of providing an interface across the system, not merely a subset of types of applications with certain design choices. Accordingly, the Court concludes that Microsoft fails to establish that the VitalSuite code is a system resource for "for use by all application programs in the computer system," whether literally or under the doctrine of equivalents, even where the "computer system" is a single server.

IV. Obviousness of Guzak '319 and '971

A. Legal Standard

"The ultimate judgment of obviousness is a legal determination," and summary judgment may be appropriate if "the content of the prior art, the scope of the patent claim, and the level of ordinary skill in the art are not in material dispute, and the obviousness of the claim is apparent in light of these factors ...." KSR Int'l Co. v. Teleflex Inc., 550 U.S. 398, 127 S.Ct. 1727, 1746, 167 L.Ed.2d 705 (2007) (citing Graham v. John Deere Co. of Kansas City, 383 U.S. 1, 17, 86 S.Ct. 684, 15 L.Ed.2d 545 (1966)). Courts also consider secondary factors, including "commercial success, long felt but unsolved needs, failure of others, etc....." which may dislodge a determination of obviousness. Id. at 1734 (quoting *Graham,* 353 U.S. at 17-18.) District courts weigh expert testimony to determine if there is an open question of fact, though a merely conclusory affidavit will not preclude summary judgment. *See* id. at 1745-46.

In *KSR,* the Supreme Court rejected a rigid application of the Federal Circuit's "teaching, suggestion, or motivation" test. *See id*. at 1734 (citing *Al-* Site Corp. v. VSI Int'l, Inc., 174 F.3d 1308, 1323-24 (Fed.Cir.1999), as an example of this test). Under this test, proof of obviousness required some teaching, suggestion, or motivation to combine prior art references "found in the prior art, the nature of the problem, or the knowledge of a person having ordinary skill in the art." *Id*. The Court determined that while "teaching, suggestion, or motivation" had "captured a helpful insight" into obviousness, it was incompatible with Supreme Court precedent when applied in a rigid and mandatory fashion. Id. at 1741. The Supreme Court observed that other more recent Federal Circuit decisions reflected a broader approach that may be consistent with its opinions. Id. at 1743 (citing DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co., 464 F.3d 1356, 1367 (Fed.Cir.2006); Alza Corp. v. Mylan Labs., Inc., 464 F.3d 1286, 1291 (Fed.Cir.2006)).

When determining obviousness, "neither the particular motivation nor the avowed purpose of the patentee controls." KSR Int'l Co., 127 S.Ct. at 1741-42. Instead, courts should determine whether the "objective reach of the claim" encompasses obvious subject matter. Id. at 1742. This may include "noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims." *Id*. "[T]he results of ordinary innovation are not the subject of exclusive rights under the patent laws." *Id*. at 1746. However, courts must avoid "falling prey to hindsight bias," "*ex post* reasoning," and "[r]igid preventative rules that deny factfinders recourse to common sense." *Id*. at 1742-43. Furthermore, "when the prior art teaches away from combining certain known elements, discovery of a successful means of combining them is more likely to be nonobvious." *Id*. at 1740.

"[A] patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art." KSR Int'l Co., 127 S.Ct. at 1740. A combination is likely nonobvious if the elements work together "in an unexpected and fruitful manner." Id. at 1740. In contrast, a patent is likely to be obvious if it merely yields a predictable result by substituting one element for another known in the field. *Id*.

An issued patent is presumed valid, so the burden of persuasion for invalidity defenses, including obviousness, is one of clear and convincing evidence *See, e.g.,* Takeda Chem. Indus., Ltd. v. Alphapharm Pty., Ltd., 492 F.3d 1350, 1355 (Fed.Cir.2007); Oakley, Inc. v. Sunglass Hut Int'l, 316 F.3d 1331, 1339 (Fed.Cir.2003).

B. The Guzak Patents

Lucent seeks summary judgment that the Guzak patents are obvious in light of XTree for Windows and the Windows Programming Guide. For the asserted claims of both patents, the Court concludes that Lucent has not established that there are no questions of material fact, particularly given its burden of clear and convincing evidence on this issue. Lucent argues that XTree for Windows disclosed every aspect of the claimed tree interface, including the ability to expand and collapse levels without changing the selection. Even assuming this is true, there is a question of fact regarding whether it would have been obvious, in light of the Windows Programming Guide, to provide the interface of XTree for Windows as a child window control. While the Programming Guide does provide guidance on efficient creation of child window controls for Windows, it does not directly propose the sort of child window control implemented by XTree for Windows. The Court therefore concludes that, viewing the evidence in a light most favorable to Microsoft, a reasonable juror could conclude that it would not have been obvious to provide this interface as a child window control. (*See* McDavit Decl. Ex. 32 (Programming Guide chapter regarding child window controls).) In summary, the Court denies Lucent's motion to the extent its seeks summary judgment of obviousness for the Guzak patents.

Conclusion

The Court hereby AMENDS its construction of the Guzak patents so that, in the asserted claims, "system resource for use by the application programs" means "a system resource available for use by all application programs in the computer system." The Court GRANTS IN PART Lucent's motion for summary judgment, having concluded that VitalSuite does not infringe claims 12 or 17 of the Guzak ' 319 patent. Finally, the Court DENIES IN PART Lucent's motion for summary judgment, to the extent Lucent seeks summary judgment that the asserted claims of the Guzak patents are obvious.

IT IS SO ORDERED.

S.D.Cal.,2008.
Lucent Technologies, Inc. v. Microsoft Corp.